FP6-IST-002020

COGNIRON

The Cognitive Robot Companion

Integrated Project

Information Society Technologies Priority

D2.1.1
Report on human tracking and identification

Due date of deliverable: 31/12/2004
Actual submission date: 26/1/2005

Start date of project: January 1st, 2004 **Duration:** 48 months

Organisation name of lead contractor for this deliverable:
University of Amsterdam

Revision: final
Dissemination Level: PU

Executive Summary

This report presents a summary of our results on tracking and identification of humans from a mobile robot in everyday environments. The report starts with a short analysis of the sensor sources that can be used for human tracking from a mobile robot. The report is continued with a summary our results on fast and robust people tracking using vision and other sensor sources. This summary is based on a number of published or submitted scientific reports. First, we present some results on detection and tracking of people using only vision. We analyze and propose an improved motion segmentation algorithm and a color feature tracking algorithm. The developed methods are analyzed and then combined to design a fast and robust vision based people tracking algorithm for a mobile robot. Second, combination with sensors other then vision was considered. A multi-modal tracking algorithm is presented where audio and laser sensors are combined with vision for robust tracking of humans. This algorithm was implemented on a real robot and tested in real environments. Finally, the problem of labelling the individuals that were tracked by the robot is addressed. A Bayesian framework was developed to associate the persons in its field of view with the people that were observed earlier. The framework was tested for vision-based people identification using simple color features. The published or submitted scientific reports that contain more details are attached to this document.

Role of (topic of deliverable) in Cogniron

Interaction with people is one of the essential capabilities of a cognitive robot assistant and therefore the interaction is an important topic in this project. For the interaction with people it is crucial that the robot is able to detect, track and label individuals it interacts with. This makes the topic of this deliverable important for the project.

Relation to the Key Experiments

The results on tracking and identification of humans presented here are essential for the Key Experiment 1: 'Home Tour Scenario'. The robot should be able to detect and track people that interact with the robot and lead the robot for a tour around the house. For the interaction it is also important to label individuals it interacts with. For the Key Experiment 2: 'Curious Robot', it is important to detect and track humans initially and for interacting with them. Detecting and tracking human is important as well for the Key Experiment 3, "Learning Skills and Tasks", for learning from observations.

1 Human tracking and identification from a mobile robot

First, a short discussion about the sensor sources that can be used for human tracking from a mobile robot is presented. Second, a short summary is given of the published or submitted reports that describe our results about human tracking and identification. The actual reports are attached to this document.

1.1 Sensor sources

One of the standard sensors used on mobile robots is a digital camera. A camera provides rich information about the environment. Due to the aspects that will be considered later in the project, like face identification and eye gaze tracking, it seems to be important to use a high resolution camera instead of normal resolution one.
For robust people tracking it would be useful to have some additional 3D information. Laser scanner is used on most mobile platforms for navigation and provides coarse 3D information in one plane. The output from the laser scanner can be used to detect legs of people walking near the robot [3]. Denser 3D information can be obtained using vision. Stereo camera would be a simple solution [2]. There are also other sensors that provide more reliable dense 3D data. For example 'time of flight' (TOF) sensors. The decision to use a TOF 3D camera instead of stereo vision can be motivated by the fact that the 3D camera is an active sensor. It transmits light and therefore is unaffected by different lightning conditions. It also means, that the depth information is detected, regardless if there is a structured surface or not. Stereo cameras need some structure to find corresponding points in both pictures. Without structure there is no depth information. However the stereo cameras are much cheaper and will be considered also.
The different types of 2D and 3D cameras are listed and compared in the attached internal report [1]. The digital camera DFK 41F02 and TOF camera Swiss Ranger SR-1 where chosen by one of the partners to be used on a robot platform. High quality data from these 2D and 3D sensors will be available for other partners for some experiments during the project. Experiments will be performed also using stereo cameras instead of the TOF camera [2].

1.2 Vision based people tracking

Vision is an important sensor source for a mobile robot. Efficient background/foreground segmentation is presented in [5]. This module is important since it can be used to detect the objects of interest (people) in a simple, fast and robust way when the camera is static. It also gives a fine segmentation of the objects from the scene which will be important for extracting the features that describe the object that are used for later identification of the objects. It is interesting to see that biological studies also support this claims: both humans and most of the animals do not have continuous but step-wise eye movements even when they are moving their head continuously. They use their eye movements to fixate their eyes to a certain point while moving their head in order to stabilize their view and to be able to detect the moving objects better.
When robot is moving it is still possible to segment the moving object from the background. A simple motion segmentation algorithm we used is described in [4]. We also developed an efficient tracking algorithm using color features [6]. An overview of the advantages and disadvantages of the developed algorithms is given in Table 1. This table is used to design a robust people tracking system that combines these algorithms described in [4].

Table 1: Fast and robust vision detection/tracking modules

	Background subtraction	Color tracking	Motion segmentation
Detecting objects- static camera	yes	no	yes
Detecting objects- moving camera	no	no	yes
Segmentation	blob (still camera)	coarse	coarse
Detecting/tracking- moving camera	no	yes	yes
Detecting/tracking- object static (short time)	yes (still camera)	yes	no
Detecting/tracking- object static (long time)	no	yes	no
Detecting/tracking- varying light conditions	no	no	yes

1.3 Multi-modal people tracking

Vision is a rich information source but other sensors can be used to greatly improve the robustness of the people tracking and identification algorithms in natural environments. A method is presented for multi-modal person tracking which uses a pan-tilt camera for face recognition, two microphones for sound source localization, and a laser range finder for leg detection. The method was implemented on a real robot and tested in natural environments. The analysis of this system is given in [3].

1.4 People identification (labelling)

In order to track the encountered people, a robot must associate persons detected in its field of view with the people that were observed earlier. The task is typically broken down into two subproblems: 1) local tracking, where the person remains in the field of view (discussed in the previous two sections) and 2) global tracking (labelling), which is the association between local tracks. The global tracking aims to find a correspondence between a local trajectory of some person with other (earlier) local trajectories of the same person. The essential difference between the two tasks follows from the assumption of persons smooth motion between frames, which provides important cues for frame-to-frame tracking.
We present in [4] our global labelling (tracking) method. A generative model is presented in which each person is identified with an unique label and each local trajectory is considered as a noisy observation from the hidden label. A Bayesian approach is used to determine the posterior density on the labels given the observations. For online performance we developed an approximation algorithm which falls into the class of deterministic assumed-density filtering approximations.

2 Future Work

The developed algorithms should be further refined. Combining vision with other sensor sources seems to be very important for designing a robust system that works in natural environments. The vision-based tracking from [4] should be further integrated into the multi-modal person tracking framework [3]. Initial experiments were already performed. Other sensor sources could also be considered (see section: 'Sensor sources'). Only some initial experiments were performed with the 'time of flight' 2.5D range sensor.

This deliverable considers only whole body tracking of humans. Further work within RA2 includes also tracking of human body parts and estimating their pose which is important for interaction with humans. Combination different sensors will be even more important there.

References

[1] K.Pfeiffer. Overview of 2d cameras and 3d TOF sensors. *IPA internal report*, 2004.

[2] S.Vacek. Sensors at UniKarl. *UKA internal report*, 2004.

[3] B. Wrede, A. Haasch, N. Hofemann, S. Hohenner, S. Hüwel, M. Kleinehagenbrock, S. Lang, S. Li, I. Toptsis, G. A. Fink, J. Fritsch, and G. Sagerer. Research issues for designing robot companions: BIRON as a case study. In P. Drews, editor, *Proc. IEEE Conf. on Mechatronics & Robotics*, volume 4, pages 1491–1496, Aachen, Germany, September 2004. Eysoldt-Verlag, Aachen.

[4] W.Zajdel, Z.Zivkovic and B.Krose. Keeping track of humans: Have I seen this person before? In *Proceedings of ICRA 2005, Barcelona, Spain*, 2005.

[5] Z. Zivkovic. Improved adaptive Gausian mixture model for background subtraction. *In Proceedings Int'l Conference Pattern Recognition*, 2004.

[6] Z. Zivkovic and B. Krose. An EM-like algorithm for color-histogram-based object tracking. *In Proceedings Conference on Computer Vision and Pattern Recognition*, 2004.

Annexes

Attached scientific papers (see references)...

2D cameras

Brand	Vendor	Resolution	Interface	Framerate	Lightamplification
ZC-XY30	Computar	752 x 582	Ethernet	25	bis 32x
ZC-Y30PH4	Computar	752 x 582	BNC	analog	bis 32x
ZC-Y11PH4	Computar	752 x 582	BNC	analog	keine
DFW-V500	Sony	640 x 480	Firewire	30	k.A.
DFW-VL500	Sony	640 x 480	Firewire	30	k.A.
DFK 21F04	The Imaging Source	640 x 480	Firewire	30	k.A.
DFK 31F03	The Imaging Source	1024 x 768	Firewire	15	k.A.
DFK 41F02	The Imaging Source	1280 x 960	Firewire	7,5	k.A.
DFW-X700	Sony	1024 x 768	Firewire	15	k.A.
DFW-SX900	Sony	1280 x 960	Firewire	7,5	k.A.
AXIS 2100	Axis	640 x 480	Ethernet	10	k.A.
AXIS 2120	Axis	704 x 576	Ethernet	25	k.A.

Sensitivity [Lux]	Connector	Size [mm] (B x H x T)	Specials	Prize [€]
Std. 3; RL 0,03	CS-Gewinde	71 x 68,5 x 175	Webserver;	1499
Std. 2,2; RL 0,03	CS-Gewinde	74 x 65 x 136,5		1075
Std. 1,2	CS-Gewinde	60 x 46 x 95		475
Std. 6 bei F1.2	C-Mount	60 x 61 x 116,7		940
Std. 14 bei F1.8	vorhanden	60 x 61 x 118,5	12x Motor zoom	1064
Std. 4 bei 1/30s	C/CS-Mount	50 x 50 x 50	Software and SDK delivered with camera	590
Std. 13 bei 1/50s	C/CS-Mount	65 x 65 x 65	Software and SDK delivered with camera	890
Std. 19 bei 1/50s	C/CS-Mount	65 x 65 x 65	Software and SDK delivered with camera	1490
Std. 20	C-Mount	50 x 50 x 110		1796
Std. 20	C-Mount	50 x 50 x 110		2596
k.A.	35mm vorhanden	41 x 102 x 147	Webserver;	399
k.A.	3,5-8mm	125 x 48 x 155	Webserver;	1130

Comparison color cameras

	Resolution	Interface	Framerate	Size [mm]	Prize [€]
Desired value:	1280 x 960	Firewire	7,5	60 x 60 x 100	1500
ZC-XY30	👎	👉	👍	👎	👍
ZC-Y30PH4	👎	👎	👉	👎	👍
ZC-Y11PH4	👎	👎	👉	👍	👍
DFW-V500	👎	👍	👍	👎	👍
DFW-VL500	👎	👍	👍	👎	👍
DFK 21F04	👎	👍	👍	👍	👍
DFK 31F03	👎	👍	👍	👍	👍
DFK 41F02	👍	👍	👍	👍	👍
DFW-X700	👎	👍	👍	👍	👎
DFW-SX900	👍	👍	👍	👍	👎
AXIS 2100	👎	👉	👍	👎	👍
AXIS 2120	👎	👉	👍	👎	👍

3D TOF cameras

Brand	Vendor	Resolution			Depth resolution max. [mm]	Interface	Framerate max. [fps]
PMD	PMDTec GmbH	16	x	16	10	RS232	2
Swiss Ranger SR-1	CSEM	124	x	160	5	USB 2.0	30
Zmini	3DV Systems	752	x	582	5	needs PC for control	30
DMC100	3DV Systems	768	x	494	5	Firewire	60

Range max. [m]	Optics	Size [mm] (B x H x T)						Specials	Prize
7,5	± 10° viewing angle	90	x	90	x	160			12.500,00 €
7,5	16mm fix	135	x	45	x	32		Also detects intensity	9.570,60 €
3,5	C-Mount	195	x	205	x	280		To big; Also detects RGB	28.383,35 €
3,5	45°	126	x	83	x	160		needs lot of computational power (Dualsystem with 2.8GHz); Resolution can be devided by 2 and 4 - Frame rate can be reduced - everything configurable by software; Range and opening angle can be customized; detects intensity also	11.210,80 €

Comparison 3D-TOF cameras

	Resolution			Depth resolution max.[mm]	Frame rate max. [fps]	Range max. [m]	Size [mm] (B x H x T)					Prize [€]
Desired value:	160	x	120	10	25	3	80	x	80	x	150	10000
PMD	👎			👎	👎	👍	👎					👎
Swiss Ranger SR-1	👍			👍	👍	👍	👍					👍
Zmini	👍			👍	👍	👍	👎					👎
DMC100	👍			👍	👍	👍	👎					👎

UKA internal report

Sensors at UniKarl

UKA uses a colour stereo camera from *videre design* (http://www.videredesign.com). The camera head is a *mega-d* camera consisting of two CMOS chips with a maximum resolution of 1280x960 pixels. For the tracking on a mobile robot a resolution of 320x240 pixels is used. The reduction of the resolution is done by binning (averaging over a block of pixels) and decimation (leaving out every second pixel and every second line) of the whole image. The camera does not possess an automatic gain control nor white-balancing. Instead, the values for gain and exposure as for blue and red channels can be adjusted manually via the provided software interface.

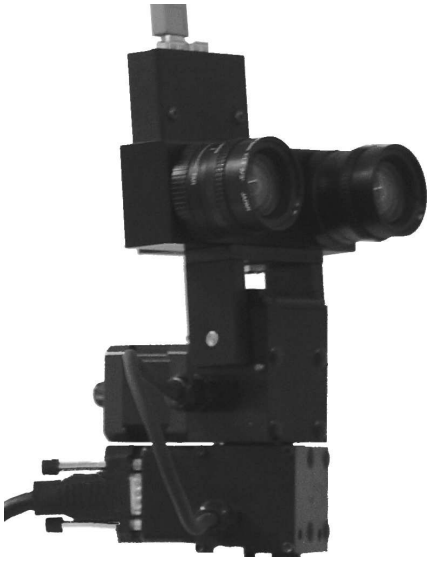


Figure 1: *mega-d* camera

As lenses, standard C-mount lenses are used. During the studies two different lenses from rainbow (s. http://www.rainbowcctv.com/) were used with focal lengths of 4.8mm and 7.5mm, respectively. The angular field of view is 85.0° x 69.0° (4.8mm) and 60.8° x 47.5° (7.5mm) respectively.

The picture on the right (fig. 1) shows the *mega-d* camera mounted on a pan-tilt-unit (model PTU-D46-17) from *directed perception* (s. http://www.dperception.com). The following figure (fig. 2) shows an image captured at 320x240 with adjusted red and blue colour channel.

Figure 2: *image at 320x240*

The camera is shipped together with software for transferring images from the camera over firewire (IEEE 1394) in the computer and for calculating depth images. For depth image

calculation the images are first rectified (that is, removal of lens distortion and alignment of images), secondly, significant features are enhanced with a Laplacian-of-Gaussian-filter and in the last step, the disparities are calculated by evaluating the cross-correlation between the left and the right image. The following figure 3 shows a rectified image-pair and the resulting disparity-map. The brighter a pixel the nearer the point is to the camera.

The last image pair (fig. 4) shows two different views of the 3d-reconstruction of the scene with the calculated depth information.

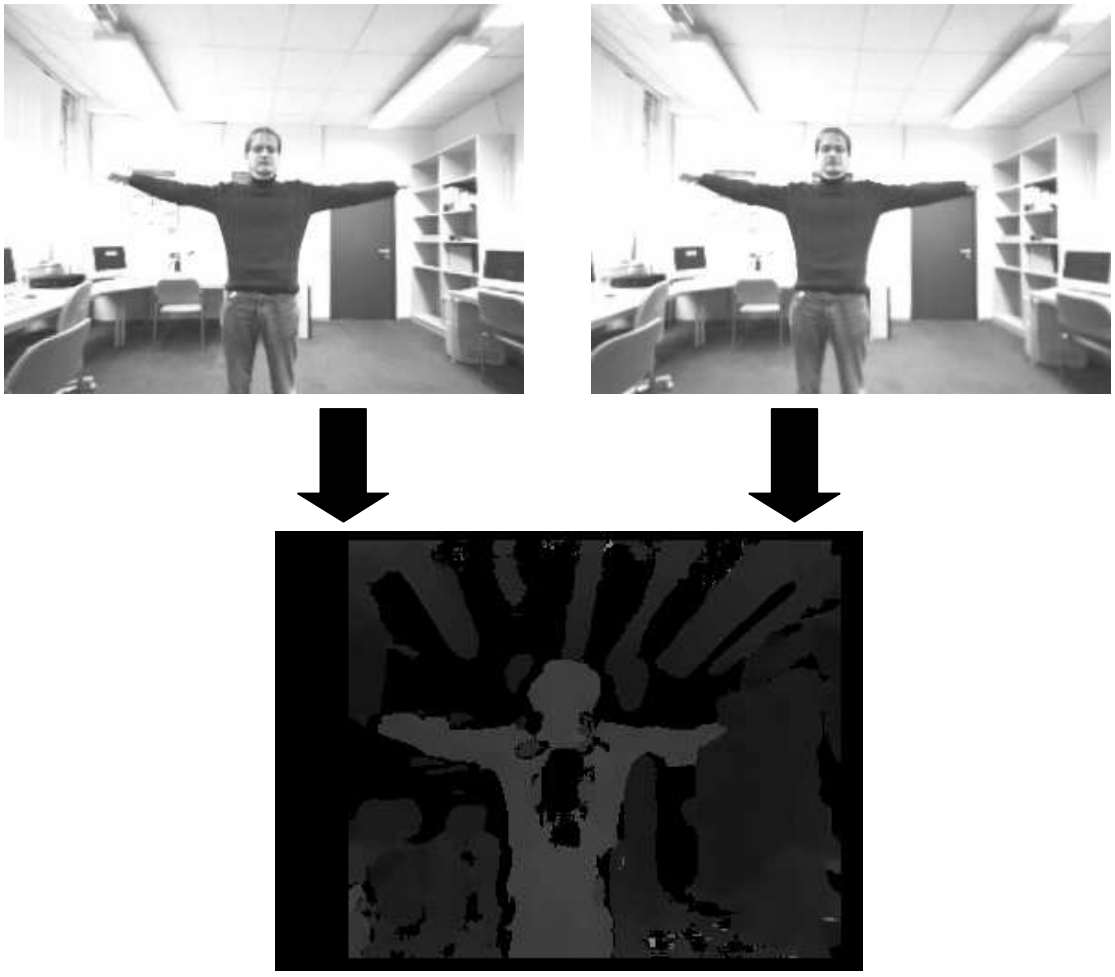


Figure 3: *stereo images and resulting disparity*

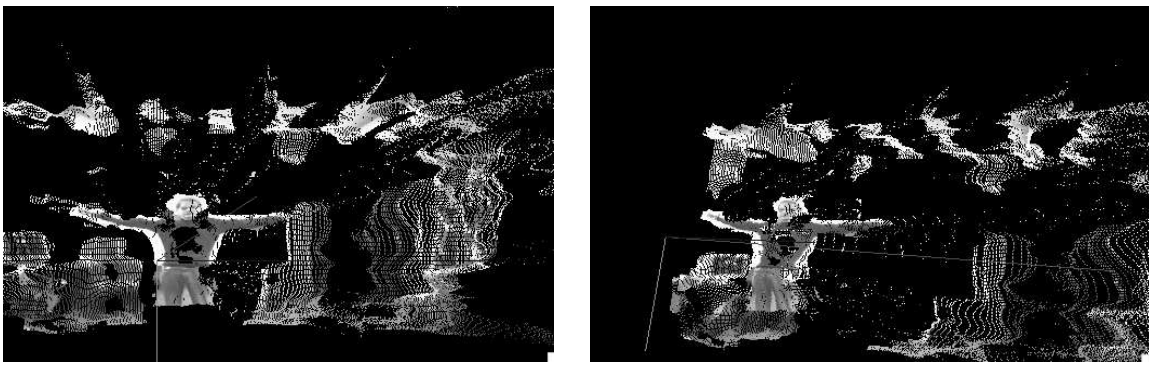


Figure 4: *reconstructed scene from two different view points*

Research Issues for Designing Robot Companions: BIRON as a Case Study

B. Wrede, A. Haasch, N. Hofemann, S. Hohenner, S. Hüwel,
M. Kleinehagenbrock, S. Lang, S. Li, I. Toptsis, G. A. Fink, J. Fritsch, and G. Sagerer*
*Bielefeld University, Faculty of Technology, 33594 Bielefeld, Germany
Email: bwrede@TechFak.Uni-Bielefeld.DE

***Abstract*— Current research in robotics is driven by the goal to achieve a high user acceptance of service robots for private households. This implies that robots have to increase their social aptness in order to become a robot companion that is able to interact in a natural way and to carry out tasks for the user.**

In this paper we present the Bielefeld Robot Companion (BIRON) as an example for the development of robot companions. BIRON is a mobile robot that is equipped with an attention system based on a multi-modal sensor system. It can carry out natural speech based dialogs and performs basic functions such as following and learning objects. We argue that the development of robot companions has to be tightly coupled with evaluation phases. We present user studies with BIRON which indicate that the functionality of a robot does not receive as much attention as the natural language interface. This indicates that the communicative behavior of a robot companion is a critical component of the system and needs to be improved before the actual functionalities of the robot can be evaluated and re-designed.

I. Introduction

One of the main current issues in developing interactive autonomous robots is the design of social robots. This focus is motivated by the insight that robots have to exhibit a basic social behavior apart from their functional capabilities in order to be accepted in the environment of a private household. Dauthenhahn and Billard offer a definition of the term *social robots* with respect to the capabilities they exhibit in the interaction with their social environment [5]:

social robots are embodied agents that are part of a heterogeneous group: a society of robots or humans. They are able to recognize each other and engage in social interactions, they possess histories (perceive and interpret the world in terms of their own experience), and they explicitly communicate with and learn from each other.

In order to achieve these goals it is proposed in [6] that a robot has to be able to show the following features and capabilities: Embodiment, emotion, dialog, personality, human-oriented perception, user model, social learning, and intentionality.

Current robotic systems' capabilities are far from showing a human-like level in all these dimensions. However, different aspects have been realized with different degrees of complexity mainly with respect to the features embodiment, human-oriented perception, and dialog.

When comparing the different service robots with respect to these features it becomes apparent that most of them share a similar level of embodiment: the systems are generally based on mobile platforms (e.g. Care-O-bot II [11], CERO [14], HERMES [2], Jijo-2 [1], Lino [16], ROBITA [20]) but only very few have actuators like arms and hands (e.g. Car-O-bot, HERMES) that enable them to fetch and carry objects, which would be one of the fundamental functionalities for a service robot at home. Sensors on such systems generally encompass visual and acoustic (speech) modalities (e.g. Care-O-bot II, HERMES, Jijo-2, Lino, ROBITA, SIG [21]). Thus, despite great differences in their physical appearance current service robots exhibit a rather standardized level of embodiment.

As for human-oriented perception, most systems are able to demonstrate attention-like behavior by visually tracking persons and focusing on a speaking person. Some systems are also able to identify different persons. It is generally observed that this is a crucial basic behavior for robots to gain and keep a person's attention and motivation for interaction.

Less homogenous – and more difficult to compare – are the dialog competences of such robots. It is generally agreed upon that a natural language interface is necessary for easy and intuitive instruction of the robot. However, current dialog systems are often restricted to prototypical command sentences and simple underlying finite state automata. Other modalities than speech, e.g. gestures, are generally ignored.

Emotional perception and production, the development of a personality, building a model of the communication partner, as well as social learning and exhibiting intentionality are features that have partly been demonstrated in so called *sociable robots* (e.g. Kismet [3] or Leonardo [4]) but not on fully autonomous robots that are supposed to fulfill service tasks. However, even such sociable robots do generally not possess sophisticated verbal communication capabilities.

In order to move towards the ambitious goal of a robot companion, which should exhibit both social aptness and service functionalities, it is necessary to perform the development in a closely coupled design-evaluation cycle. In effect, long term user studies such as, for example, performed with CERO are necessary in order to understand the long term influence of contextual variables such as ergonomic features or the

[1]This work has been supported by the European Union within the 'Cognitive Robot Companion' (COGNIRON) project (FP6-IST-002020) and by the German Research Foundation within the Collaborative Research Center 'Situated Artificial Communicators' as well as the Graduate Programs 'Task Oriented Communication' and 'Strategies and Optimization of Behavior'.

reactions of bypassing people. With our robot BIRON we want to address this intersection of social capabilities and functional behavior by enabling the system to carry out a more sophisticated dialog for handling instructions and learning new parts of its environment. One scenario that we envision within the COGNIRON project[1] is a home-tour where a user is supposed to show BIRON around his or her home. This scenario requires BIRON to carry out a natural dialog in order to understand commands e.g. for following and to learn new objects and rooms.

We addressed the issue of evaluation by performing first preliminary user studies in order to evaluate single system components and to better understand in which direction we have to guide the further development of our robot. As we will show, a robot has to reach a certain level of verbal competence before it will be accepted as a social communication partner and before its functional capabilities will be perceived as interesting and useful.

In this paper we will first present the overall system architecture (Section II) and hardware (Section III) before describing the modules in more detail in Sections IV to VI. The current interaction capabilities are shortly described in Section VII. We present results from our user studies Section VIII.

II. System Overview and Architecture

Since interaction with the user is the basic functionality of a robot companion, the integration of interaction components into the architecture is a crucial factor. We propose to use a special control component, the so-called *execution supervisor*, which is located centrally in the robot's architecture [15]. The data flow between all modules is event-based and every message is coded in XML. The modules interact through a specialized communication framework [25]. The robot control system (see Fig. 1) is based on a three-layer architecture [9] which consists of three components: a reactive feedback control mechanism, a reactive plan execution mechanism, and a mechanism for performing deliberative computations.

The execution supervisor, the most important architecture component, represents the reactive plan execution mechanism. It controls the operations of the modules responsible for deliberative computations rather than vice versa. This is contrary to most hybrid architectures where a deliberator continuously generates plans and the reactive plan execution mechanism just has to assure that a plan is executed until a new plan is received. To continuously control the overall system the execution supervisor performs only computations that take a short time relative to the rate of environmental change perceived by the reactive control mechanism.

While the execution supervisor is located in the intermediate layer of the architecture, the dialog manager is part of the deliberative layer. It is responsible for carrying out dialogs to receive instructions given by a human interaction partner. The

[1]COGNIRON is an integrated Project of a European consortium that is supported by the European Union. For more details of this project see http://www.cogniron.org.

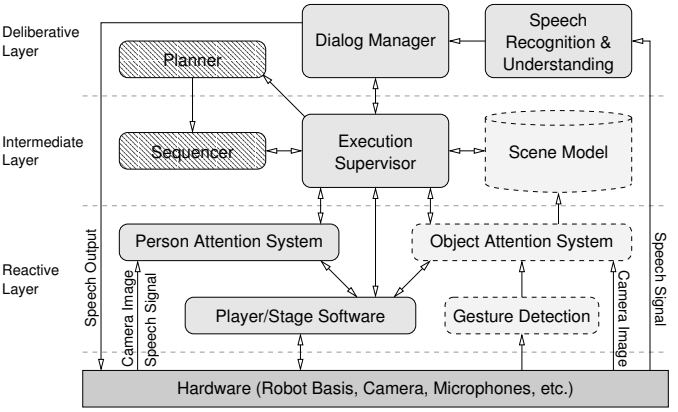


Fig. 1. Overview of the BIRON architecture (implemented modules are drawn with solid lines, modules under development with dashed lines).

dialog manager is capable of managing interaction problems and resolving ambiguities by consulting the user (see Section VI). It receives input from speech processing which is also located on the topmost layer (see Section V) and sends valid instructions to the execution supervisor.

The person attention system represents the reactive feedback control mechanism and is therefore located on the reactive layer (see Section IV). However, the person attention system does not directly control the robot's hardware. This is done by the *Player/Stage* software [10]. *Player* provides a clean and simple interface to the robot's sensors and actuators. Even though we currently use this software to control the hardware directly, the controller can easily be replaced by a more complex component which may be based on, e.g., behaviors.

In addition to the person attention system we are currently developing an object attention system for the reactive layer. The execution supervisor can shift control of the robot from the person attention system to the object attention system in order to focus objects referred to by the user. The object attention will be supported by a gesture detection module which recognizes deictic gestures [13]. Combining spoken instructions and a deictic gesture allows the object attention system to control the robot and the camera in order to acquire visual information of a referenced object. This information will be sent to the scene model in the intermediate layer.

The scene model will store information about objects introduced to the robot for later interactions. This information includes attributes like position, size, and visual information of objects provided by the object attention module. Additional information given by the user is stored in the scene model as well, e.g., a phrase like *"This is my coffee cup"* indicates owner and use of a learned object.

The deliberative layer can be complemented by a component which integrates planning capabilites. This planner is responsible for generating plans for navigation tasks, but can be extended to provide additional planning capabilities which could be necessary for autonomous actions without the human. As the execution supervisor can only handle single commands,

a sequencer on the intermediate layer is responsible for decomposing plans provided by the planner. However, in this paper we will focus on the interaction capabilities of the robot.

III. Hardware

Our system architecture is implemented on our mobile robot BIRON (see Fig. 2). Its hardware platform is a Pioneer PeopleBot from ActivMedia with an on-board PC (Pentium III, 850 MHz) for controlling the motors and the on-board sensors and for sound processing. An additional PC (Pentium III, 500 MHz) inside the robot is used for image processing and for data association.

The two PCs running Linux are linked by an 100 Mbit Ethernet LAN and the controller PC is equipped with wireless LAN to enable remote control of the robot. As additional interactive device a 12" touch screen display is provided on the front side.

A pan-tilt color camera (Sony EVI-D31) is mounted on top of the robot at a height of 141 cm for acquiring images of the upper body part of humans interacting with the robot. Two AKG far-field microphones which are usually used for hands free telephony are located at the front of the upper platform at a height of 106 cm, right below the touch screen display. The distance between the microphones is 28.1 cm. A SICK laser range finder is mounted at the front at a height of approximately 30 cm.

Fig. 2. BIRON.

IV. The Person Attention System

A robot companion should enable users to engage in an interaction as easily as possible. For this reason the robot has to continuously keep track of all persons in its vicinity and must be able to recognize when a person starts talking to it. Therefore, both acoustic and visual data provided by the on-board sensors have to be taken into account: At first the robot needs to know which person is speaking, then it has to recognize whether the speaker is addressing the robot, i.e., looking at it. On BIRON the necessary data is acquired from a multi-modal person tracking framework which is based on *multi-modal anchoring* [8].

A. Multi-Modal Person Tracking

Multi-modal anchoring allows to simultaneously track multiple persons. The framework efficiently integrates data coming from different types of sensors and copes with different spatio-temporal properties of the individual modalities. Person tracking on BIRON is realized using three types of sensors. First, the laser range finder is used to detect humans' legs. Pairs of legs result in a characteristic pattern in range readings and can be easily detected [8]. Second, the camera is used to recognize faces and torsos. Currently, the face detection works

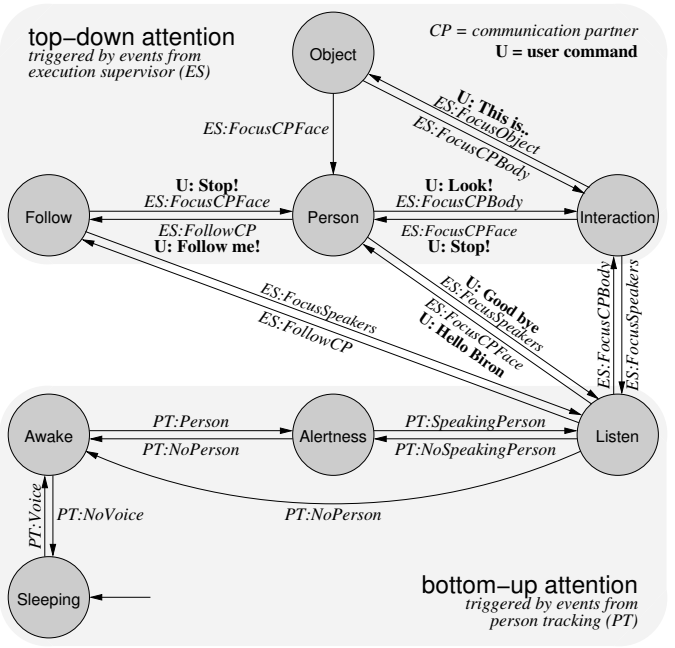


Fig. 3. Finite state machine realizing the different behaviors of the person attention mechanism. Commands from the user, that are processed by the dialog component, are displayed in bold face.

for faces in frontal view only [17]. The clothing of the upper body part of a person is observed by tracking the color of the person's torso [7]. Third, the stereo microphones are applied to locate sound sources in front of the robot. By incorporating information from the other cues robust speaker localization is possible [17]. Altogether, the combination of depth, visual, and auditory cues allows the robot to robustly track persons in its vicinity.

However, since BIRON has only limited sensing capabilities – just like a human has only limited cognitive resources – we implemented an attention mechanism for more complex situations with many people moving around BIRON.

B. Attention Mechanism

The attention mechanism has to fulfill two tasks: On the one hand it has to select the person of interest from the set of observed persons. On the other hand it has to control the alignment of the sensors in order to obtain relevant information from the persons in the robot's vicinity.

The attention mechanism is realized by a finite state machine (see Fig. 3). It consists of several states of attention, which differ in the way the robot behaves, i.e., how the pan-tilt unit of the camera or the robot itself is controlled. The states can be divided into two groups representing *bottom-up attention* while searching for a communication partner and *top-down attention* during interaction.

When bottom-up attention is active, no particular person is selected as the robot's communication partner. The selection of the person of interest as well as transitions between different states of attention solely depend on information provided by the person tracking component. For selecting a person of

interest, the observed persons are divided into three categories with increasing degree of relevance. The first category consists of persons that are not speaking. The second category comprises all persons that are speaking, but at the same time are either not looking at the robot or the corresponding decision is not possible, since the person is not in the field of view of the camera. Persons assigned to the third category are of most interest to the robot. These persons are speaking and at the same time are looking at the robot. In this case the robot assumes to be addressed and considers the corresponding person to be a potential communication partner.

Top-down attention is activated as soon as the robot starts to interact with a particular person. During interaction the robot's focus of attention remains on this person even if it is not speaking. Here, in contrast to bottom-up attention, transitions between different states of attention are solely triggered by the execution supervisor which reacts to user commands processed by the dialog component. For detailed information concerning the control of the hardware see [12].

V. Speech Processing

As speech is the most important modality for a multimodal dialog, speech processing has to be done thoroughly. On BIRON there are two major challenges: Speech recognition has to be performed on distant speech data recorded by the two on-board microphones and speech understanding has to deal with spontaneous speech.

While the recognition of distant speech with our two microphones is achieved by beam-forming [18], the activation of speech recognition is controlled by the attention mechanism presented in the previous section. Only if a tracked person is speaking and looking at the robot at the same time, speech recognition and understanding takes place. Since the position of the speaker relative to the robot is known from the person tracking component, the time delay can be estimated and taken into account for the beam-forming process.

The speech understanding component processes recognized speech and has to deal with spontaneous speech phenomena. For example, large pauses and incomplete utterances can occur in such task oriented and embodied communication. However, missing information in an utterance can often be acquired from the scene. For example the utterance *"Look at this"* and a pointing gesture to the table can be combined to form the meaning *"Look at the table"*. Moreover, fast extraction of semantic information is important for achieving adequate response times.

We obtain fast and robust speech processing by combining the speech understanding component with the speech recognition system. For this purpose, we integrate a robust LR(1)-parser into the speech recognizer as proposed in [24]. Besides, we use a semantic-based grammar which is used to extract instructions and corresponding information from the speech input. A semantic interpreter forms the results of the parser into frame-based XML-structures and transfers them to the dialog manager. Hints in the utterances about gestures are also incorporated. For our purpose, we consider co-verbal gestures only.

For the object attention system it is intended to use this information in order to detect a specified object. Thus, this approach supports the object attention system and helps to resolve potential ambiguities.

VI. Dialog

The model of the dialog manager is based on a set of *finite state machines* (FSM), where each FSM represents a specific dialog [23]. The FSMs are extended with the ability of recursive activation of other FSMs and the execution of an action in each state. Actions that can be taken in certain states are specified in the *policy* of the dialog manager. These actions include the generation of speech output and sending events like orders and requests to the execution supervisor. The dialog *strategy* is based on the so-called *slot-filling* method [22]. The task of the dialog manager is to fill enough slots to meet the current dialog goal, which is defined as a goal state in the corresponding FSM. The slots are filled with information coming from the user and other components of the robot system. After executing an action, which is determined by a lookup in the dialog policy, the dialog manager waits for new input from the execution supervisor or the speech understanding system.

As users interacting with a robot companion often switch between different contexts, the slot-filling technique alone is not sufficient for adequate dialog management. Therefore, the processing of a certain dialog can be interrupted by another one, which makes alternating instruction processing possible. Dialogs are specified using a declarative definition language and encoded in XML in a modular way. This increases the portability of the dialog manager and allows an easier configuration and extension of the defined dialogs.

VII. Interaction Capabilities

In the following we describe the interaction capabilities BIRON offers to the user in our current implementation. Initially, the robot observes its environment. If persons are present in the robot's vicinity, it focuses on the most interesting one. A user can start an interaction by greeting the robot with, e.g., *"Hello BIRON"* (see Fig. 3). Then, the robot keeps this user in its focus and can not be distracted by other persons talking. Next, the user can ask the robot to follow him to another place in order to introduce it to new objects. While the robot follows a person it tries to maintain a constant distance to the user and informs the person if she moves too fast. When the robot reaches a desired position the user can instruct it to stop. Then, the user can ask the robot to learn new objects. In this case the camera is lowered to also get the hands of the user in the field of view. When the user points to a position and gives spoken information like *"This is my favorite cup"*, the object attention system is activated in order to center the referred object. However, since the gesture recognition and the object attention modules are not yet integrated in our system, this behavior is simulated by always moving the camera to a

Fig. 4. Several scenes from users interacting with BIRON during our first user studies.

predefined position when reaching the attentional state *Object*. If the user says *"Good-bye"* to the robot or simply leaves while the robot is not following the user, the robot assumes that the current interaction is completed and looks around for new potential communication partners.

VIII. Evaluation

We carried out first user studies with BIRON by assessing qualitative statements from users about the capabilities of BIRON. We asked 21 subjects to interact with BIRON. Figure 4 shows some interaction scenes from these experiments. Interaction times (i.e. the time where only one user interacted with BIRON) averaged between 3 and 5 minutes. As an introduction the users were given an overview of BIRON's interaction capabilities which displayed a schema of potential commands similar to the graph shown in Figure 3. Afterwards they had to fill out a questionnaire where we asked, among others, for the most and the least preferred features that they had experienced during their interactions with BIRON. More detailed results of this evaluation are reported [19].

It turned out that the most interesting features for users were the natural language interface and the person attention behavior (see Fig. 5). The more task-oriented functions – the following behavior and the object learning ability – received less positive feedback. This indicates that the functional capabilities of BIRON did not receive as much attention as one would expect and seem to be obscured by other features of the system.

On the other hand, although all users did already have some experience with speech recognition systems (ASR), the most frequently named dissatisfaction concerned the errors of the ASR system (see Fig. 6). Wishes for a more flexible dialog and a more stable system were the only other significant dimensions of answers to this open question, although less frequently named.

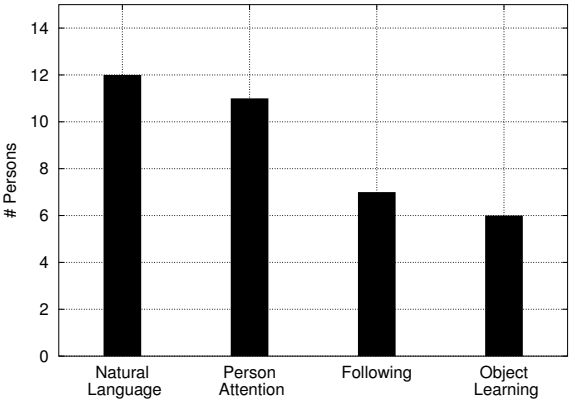


Fig. 5. User answers to the question "What did you like most?"

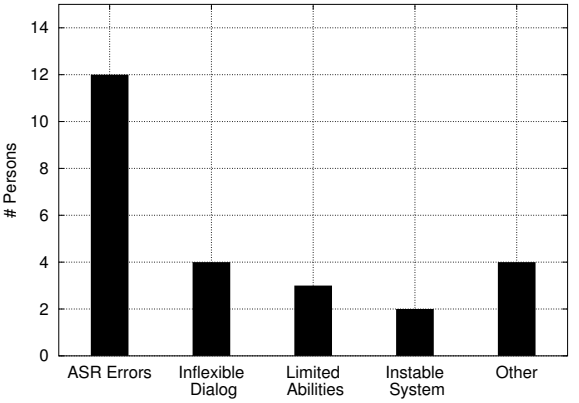


Fig. 6. User answers to the question "What did you like least?"

These results emphasize the importance of a natural language interface which allows for natural interactions. However, they also demonstrate that users are extremely sensitive to problems that occur within the communication. Thus, the natural language capability of a robot is a crucial part for human-robot interaction. If the communication does not proceed in a smooth way, the user will not be motivated to access all the potential functionalities of the robot.

In addition to these results we also assessed the usefulness of the feedback of different internal processing results and states. It turned out that users generally found feedback very helpful. However, users tend to have highly individual preferences as to the means of feedback they prefer. While some users liked to see the results of the ASR system, others found these too technical and disturbing from the actual task. On the other hand, the feedback of the internal attentional state of the system was generally perceived as very helpful. This shows that while feedback on the internal system status is helpful it has to be conveyed in an acceptable way to the user. A powerful means that humans use in their communication are nonverbal signals such as gestures or mimic. It seems to be promising to implement more of such nonverbal communication on a robotic companion as demonstrated on sociable

robots such as Kismet or Leonardo ([3], [4]).

IX. Conclusion

In order for a robot to be accepted as a social communication partner it should exhibit a range of features and functionalities. The main features that current state-of-the-art robots exhibit concern embodiment, human-oriented perception and dialog.

In this paper we argued that the levels of embodiment and human-oriented perception, that current state-of-the-art robots share, have reached a standard which is – with the exception of missing actuators – quite acceptable for human users. We demonstrated this with first user studies on BIRON which showed that the attentional behavior of BIRON receives significant positive feedback while the functional features (person following, object learning) did not receive as much attention by the same subjects. We suppose that this is due to the limitations of the natural language interface which, while being the preferred communication channel for human users, is currently the most critical system component. Here, user wishes direct our research towards a more robust speech recognition system and a more flexible dialog. We are currently planning to use a head-mounted microphone for getting cleaner speech for the speech recognition system in addition to the stereo microphones that we use for the speaker localization.

These results indicate that a robot companion has to show acceptable communication skills in order to be acceptable both at a social and a functional level. They also demonstrate that it is necessary to tightly couple user studies with design and development phases. In order to build robots that are acceptable as social communication partners it is necessary to identify critical aspects of the system. Within the design-development-evaluation cycle of BIRON the current findings direct our research towards developing new means for a more robust, embodied communication framework.

References

[1] H. Asoh, Y. Motomura, F. Asano, I. Hara, S. Hayamizu, K. Itou, T. Kurita, T. Matsui, N. Vlassis, R. Bunschoten, and B. Kröse. Jijo-2: An office robot that communicates and learns. *IEEE Intelligent Systems*, 16(5):46–55, 2001.

[2] R. Bischoff and V. Graefe. Demonstrating the humanoid robot *HERMES* at an exhibition: A long-term dependability test. In *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems; Workshop on Robots at Exhibitions*, Lausanne, Switzerland, 2002.

[3] C. Breazeal. *Designing Sociable Robots*. Bradford Books, 2002.

[4] C. Breazeal, D. Buchsbaum, J. Gray, D. Gatenby, and B. Blumberg. Learning from and about others: Towards using imitation to bootstrap the social understanding of others by robots. *Artificial Life*, 2004. to appear.

[5] K. Dautenhahn and A. Billard. Bringing up robots or – the psychology of socially intelligent robots: From theory to implementation. In *Proc. of the Autonomous Agents*, 1999.

[6] T. Fong, I. Nourbakhsh, and K. Dautenhahn. A survey of socially interactive robots. *Robotics and Autonomous systems*, 42:143–166, 2003.

[7] J. Fritsch, M. Kleinehagenbrock, S. Lang, G. A. Fink, and G. Sagerer. Audiovisual person tracking with a mobile robot. In *Proc. Int. Conf. on Intelligent Autonomous Systems*, pages 898–906. IOS Press, 2004.

[8] J. Fritsch, M. Kleinehagenbrock, S. Lang, T. Plötz, G. A. Fink, and G. Sagerer. Multi-modal anchoring for human-robot-interaction. *Robotics and Autonomous Systems, Special issue on Anchoring Symbols to Sensor Data in Single and Multiple Robot Systems*, 43(2–3):133–147, 2003.

[9] E. Gat. On three-layer architectures. In D. Kortenkamp, R. P. Bonasso, and R. Murphy, editors, *Artificial Intelligence and Mobile Robots: Case Studies of Successful Robot Systems*, chapter 8, pages 195–210. MIT Press, Cambridge, MA, 1998.

[10] B. P. Gerkey, R. T. Vaughan, and A. Howard. The player/stage project: Tools for multi-robot and distributed sensor systems. In *Proc. Int. Conf. on Advanced Robotics*, pages 317–323, 2003.

[11] B. Graf, M. Hans, and R. D. Schraft. Care-O-bot II—Development of a next generation robotic home assistant. *Autonomous Robots*, 16(2):193–205, 2004.

[12] A. Haasch, S. Hohenner, S. Hüwel, M. Kleinehagenbrock, S. Lang, I. Toptsis, G. A. Fink, J. Fritsch, B. Wrede, and G. Sagerer. BIRON – The Bielefeld Robot Companion. In *Proc. Int. Workshop on Advances in Service Robotics*, pages 27–32, 2004.

[13] N. Hofemann, J. Fritsch, and G. Sagerer. Recognition of deictic gestures with context. In *Proc. DAGM'04*. Springer-Verlag, 2004. to appear.

[14] H. Hüttenrauch and K. Severinson Eklundh. Fetch-and-carry with CERO: Observations from a long-term user study with a service robot. In *Proc. IEEE Int. Workshop on Robot-Human Interactive Communication (ROMAN)*, pages 158–163. IEEE Press, 2002.

[15] M. Kleinehagenbrock, J. Fritsch, and G. Sagerer. Supporting advanced interaction capabilities on a mobile robot with a flexible control system. In *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Sendai, Japan, September/October 2004. to appear.

[16] B. J. A. Kröse, J. M. Porta, A. J. N. van Breemen, K. Crucq, M. Nuttin, and E. Demeester. Lino, the user-interface robot. In *European Symposium on Ambient Intelligence (EUSAI)*, pages 264–274, 2003.

[17] S. Lang, M. Kleinehagenbrock, S. Hohenner, J. Fritsch, G. A. Fink, and G. Sagerer. Providing the basis for human-robot-interaction: A multi-modal attention system for a mobile robot. In *Proc. Int. Conf. on Multimodal Interfaces*, pages 28–35. ACM, 2003.

[18] S. J. Leese. Microphone arrays. In G. M. Davis, editor, *Noise Reduction in Speech Applications*, pages 179–197. CRC Press, Boca Raton, London, New York, Washington D.C., 2002.

[19] S. Li, M. Kleinehagenbrock, J. Fritsch, B. Wrede, and G. Sagerer. "BIRON, let me show you something": Evaluating the interaction with a robot companion. In *Proc. IEEE Int. Conf. on Systems, Man, and Cybernetics, Special Session on Human-Robot Interaction*, The Hague, The Netherlands, October 2004. IEEE. to appear.

[20] Y. Matsusaka, T. Tojo, and T. Kobayashi. Conversation robot participating in group conversation. *IEICE Trans. on Information and System*, E86-D(1):26–36, 2003.

[21] H. G. Okuno, K. Nakadai, and H. Kitano. Social interaction of humanoid robot based on audio-visual tracking. In *Proc. Int. Conf. on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems*, Cairns, Australia, 2002. Lecture Notes in Artificial Intelligence, Springer.

[22] B. Souvignier, A.Kellner, B. Rueber, H. Schramm, and F. Seide. The thoughtful elephant: Strategies for spoken dialog systems. In *IEEE Trans. on Speech and Audio Processing*, volume 8, pages 51–62, 2000.

[23] I. Toptsis, S. Li, B. Wrede, and G. A. Fink. A multi-modal dialog system for a mobile robot. In *Proc. Int. Conf. on Spoken Language Processing*, 2004. to appear.

[24] S. Wachsmuth, G. A. Fink, and G. Sagerer. Integration of parsing and incremental speech recognition. In *Proc. European Conf. on Signal Processing*, volume 1, pages 371–375, Rhodes, 1998.

[25] S. Wrede, J. Fritsch, C. Bauckhage, and G. Sagerer. An XML based framework for cognitive vision architectures. In *Proc. Int. Conf. on Pattern Recognition*, Cambridge, UK, 2004. to appear.

Keeping track of humans: have I seen this person before?

Wojciech Zajdel, Zoran Zivkovic and Ben Kröse
Intelligent Autonomous Systems Group
University of Amsterdam
Kruislaan 403, 1098SJ Amsterdam
*wzajdel,zoran,krose@*science.uva.nl

***Abstract*— In this paper we describe a system which enables a mobile robot equipped with a color vision system to track humans in indoor environments. We developed a method for tracking humans when they are within the field of view of the camera, based on motion and color cues. However, the robot also has to keep track of humans which leave the field of view and re-enter later. We developed a dynamic Bayesian network for such a global tracking task. Experimental results on real data confirm the viability of the developed method.**

***Index Terms*— Human-robot interaction, people tracking, vision-based user interfaces.**

I. Introduction

Current robotic mobile platforms are moving out of the factory floor into environments inhabited by human. Examples are museum robots or exhibition robots [2], [18], care-for-elderly robots [6], office robots [3] and home and entertainment robots [19]. In their role as guide, servant or companion these systems have to interact with the humans they encounter. The robot has to be aware of their presence, intentions and context, and has to be responsive to their needs, habits, gestures and emotions. As a part of this tremendous difficult task we present in this paper a system which is able to keep track of multiple people and is able to identify them.

Tracking multiple people has been traditionally studied for surveillance with static cameras, where the research focuses on detection of moving objects, the fusion of data from multiple overlapping cameras and on the association (tracking) between non-overlapping cameras [12], [20]. In robotic applications typically laser range finders are used [15] or combinations of laser and vision, where the vision is used for identification [5], [9]. Detecting moving persons with a moving camera is also studied in the field of intelligent vehicles [10]

In this paper we focus on the problem of tracking and (re)identification people from a (possibly moving) mobile robot equipped with a vision system.

II. Overview

In order to track the encountered people, a robot must associate persons detected in its field of view with the people that were observed earlier. The task is typically (e.g. [12]) broken down into two subproblems: 1) local tracking, where the person remains in the field of view and 2) global tracking, which is the association between local tracks. Local tracking takes place at a frame-to-frame level and aims to collect all observations of an individual as long as it stays in the robot's field of view. Global tracking aims to find a correspondence between a local trajectory of some person with other (earlier) local trajectories of the same person.

The essential difference between the two tasks follows from the assumption of person's smooth motion between frames, which provides important cues for frame-to-frame tracking. In this paper we treat local tracking as a video preprocessing step, and focus on identification from local trajectories, as outlined in Fig. 1.

In section III we present our approach for detection and local tracking from a robot platform. In section IV we present our global tracking method. A generative model is presented in which each person is identified with an unique label and each local trajectory is considered as a noisy observation from the hidden label. A Bayesian approach is used to determine the posterior density on the labels given the observations. For online performance we developed an approximation algorithm which falls into the class of deterministic assumed-density filtering approximations [4]. In section V we present experimental results on real data.

III. Object detection and local tracking

Our video processing module needs to detect people visible within the robot's field of view, has to represent the detected persons as pixel regions (blobs) and has to track each blob as long as the corresponding person remains visible. We need to known which pixels belong to a person in order to give an accurate description on its appearance, which later becomes the key cue for identification. The task is particularly difficult since the robot, and hence the camera, may be in motion.

We use three processes to detect and track moving objects in a static background: a) an extension of the 'background subtraction' technique, b) an optic-flow based method and c) a color-histogram based tracker. Robot motion information is used to switch between the optic-flow based process and the background subtraction process. Figure 1 shows the scheme of the module.

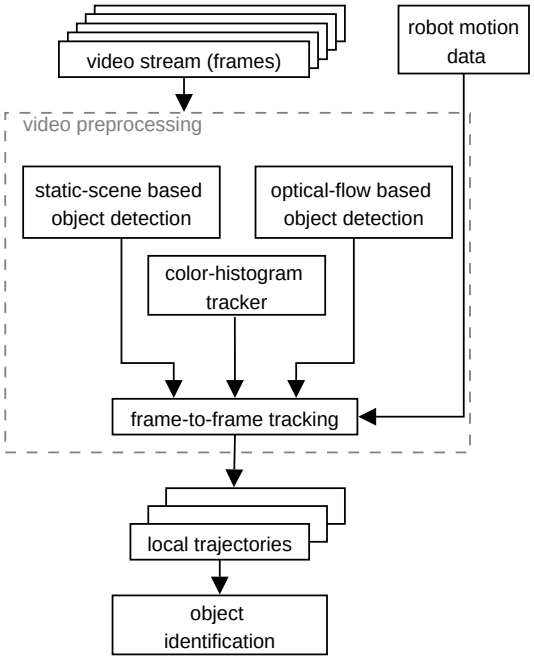
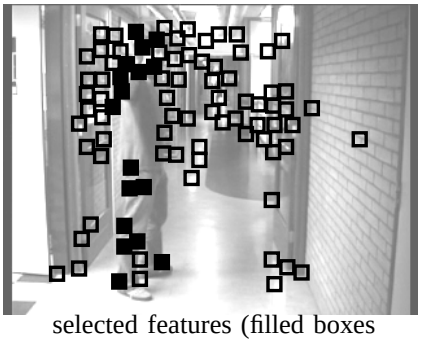


Fig. 1. Basic components of the identification system.

original image



selected features (filled boxes represent the features that do not fit the rigid motion model of the scene)

Fig. 2. Robust object tracking with a moving camera.

A. Object detection with a static camera

One of the most commonly used approaches for the detection of moving objects with a static camera is presented in [17] and further elaborated in [21]. We used the latter method, extended with a shadow-pixel removal from [14]. In a postprocessing step we filter the image to extract blobs. We used here just first two moments of the blobs to detect the interesting ones. Since the procedure is on-line and adaptive we have to take care when the robot state changes from moving to static: background estimation should be started carefully. The objects that were tracked during the moving phase might stay still and occlude a part of the background. This part of the background is learned only when the objects that were tracked during the moving phase leave the image.

B. Object detection with moving camera

In the case of a moving platform, we use the full 3D rotation and translation motion model. Our approach is similar to the recent robust solution from [16], but simpler and appropriate for real-time applications. We track 100 corner-like points using the Kanade-Lucas-Tomasi tracker. We use a robust method to estimate the fundamental matrix and detect the points that deviate from the model. The detected points correspond to the moving objects, which can also be non-rigid. An example is presented in Fig 2. To estimate the position of the object we use again the second order moments to approximate the 2D shape of the object by an ellipse. We use the moment estimates from the previous frame and keep them fixed. We use the mean-shift algorithm [13] to find the ellipse that will contain the most of the detected points.

C. Color-histogram tracking

Using color-histogram as the model of the tracked object is a very robust representation of the object appearance. An efficient method for color-histogram-based object tracking is presented in [7] and extended by [22]. The shape of the tracked non-rigid object is approximated by its second order moments - an ellipse. The appearance of the object is described by its color histogram. The algorithm finds the candidate region from the new image that maximizes the similarity function between the color histogram of the object and the color histogram of the candidate ellipsoidal region. The similarity measure we use is the Bhattacharrya coefficient (see [7]). We use the estimated position of the objects from the previous frame (as estimated by the background subtraction or optic flow method) to initialize the search but we also try some random initializations to find global maximum. This algorithm is robust to movement of the camera or when the object remains static for a long time. However, the algorithm does not detect objects but just finds the most similar region in the image.

IV. Global tracking (re-identification)

Global tracking aims to re-identify a person when it leaves the filed of view and re-renters later. This task requires association of local trajectories. The problem

cannot be solved with smooth-motion based trackers (e.g. Kalman filter), due to motion discontinuity between local trajectories.

We consider a local trajectory of a person as a *single* observation $y_k = \{o_k, d_k\}$, where k is the observation index (in time order), o_k describes r-dimensional color features computed while a person was visible; and $d_k = \{t_k^{\mathrm{e}}, s_k^{\mathrm{e}}, t_k^{\mathrm{q}}, s_k^{\mathrm{q}}\}$ are spatio-temporal features: the robot's location (s_k^{e}) and time (t_k^{e}) when the person entered the filed of view, and location (s_k^{q}) and time (t_k^{q}) – when quited.

The underlying idea is to identify every person with an unique label and define a probabilistic dependency (a generative model) between the labels and observations. This dependency takes the form of a mixture model, where each mixture component is parameterized by the state of a different person. Since the number of people is unknown, we postulate a new mixture component for every observation. Our model resembles "Dirichlet process mixture models" [1], which have been recently applied to tracking with multiple static cameras [20].

A. Generative Model

Although the underlying color properties of a person do not change, the color features differ whenever the person is observed due to the varying pose or illumination. We assume that the features are samples from a Normal pdf specific to the person. For every observed o_k we introduce a latent variable $x_k = \{\mathbf{m}_k, \mathbf{V}_k\}$ that represents parameters of the Normal density (kernel) from which o_k is sampled. The $r \times 1$ vector $\mathbf{m}$ describes the person's specific, expected features. The $r \times r$ covariance matrix $\mathbf{V}$ tells how sensitive the person's appearance is to the changing observation conditions. For instance, the appearance a person dressed uniformly in black is relatively independent of illumination or pose, so his/her covariance has small eigenvalues. The appearance of a person dressed in white or non-uniform colors is easily affected by pose or illumination, so we model such a person with a 'broad' kernel.

We treat the parameters $x = \{\mathbf{m}, \mathbf{V}\}$ as a latent state of an object. The state is considered a random variable with a prior distribution $\pi(x)$. In Bayesian statistics [11], a convenient joint model for the mean $\mathbf{m}$ and covariance $\mathbf{V}$ of a Gaussian kernel is a product of a Normal ($\mathcal{N}$) density w.r.t $\mathbf{m}$, and an Inverse Wishart ($\mathcal{IW}$) density w.r.t $\mathbf{V}$. The $\mathcal{IW}$ model is a multivariate generalization of the Inverse Gamma distribution. We denote the joint model as:

$$\pi(x) = \phi(x|\theta_0) = \mathcal{N}(\mathbf{m}; \mathbf{a}_0, \kappa_0 \mathbf{V}) \mathcal{IW}(\mathbf{V}; \eta_0, \mathbf{C}_0) \quad (1)$$

where $\theta_0 = \{\mathbf{a}_0, \kappa_0, \eta_0, \mathbf{C}_0\}$ are hyperparameters defining the prior density [11]. One of the greatest advantages of this approach is that the noise variance need not be learned with a maximum likelihood criterion (e.g. with EM algorithm). The noise variance is person-dependent, and as a part of the state, it will be estimated on-line by a filtered density.

The spatio-temporal features d of a local pass are observed noise-free. A sequence $\{d_1^{(n)}, d_2^{(n)}, \ldots\}$ assigned to nth person (denoted by the superscript) defines a path in the building. We model the path as a random, first-order Markov process. The path is started by sampling from an initial distribution P_{δ_0}, and extended by sampling from a transition distribution $P_\delta(d_{i+1}^{(n)} | d_i^{(n)})$. The distributions P_δ and P_{δ_0} follow from the topology of the environment and robot's prior knowledge about typical paths. One can use a simple model that just prevents observing a person more than once at the same time, or an elaborate (but first-order) model of paths, that is learned beforehand [5].

a) Association variables: Our model is organized into slices, where each slice corresponds to a single observation y_k. For every y_k there is a corresponding variable s_k that denotes the label of the person to which y_k is assigned. Within the first k data, $y_{1:k} \equiv \{y_1, \ldots, y_k\}$, there may be at most k different people, so s_k has k different states; $s_k \in \{1, \ldots, k\}$. The label s_k is accompanied by auxiliary variables: a counter c_k, and pointers $z_k^{(1)}, \ldots, z_k^{(k)}$. The counter, $c_k \in \{1, \ldots, k\}$, indicates the number of different persons present in the data $y_{1:k}$. The nth pointer variable, $z_k^{(n)} \in \{0, \ldots, k-1\}$, denotes the index of the *last* observation of the nth person *before* slice k. Value $z_k^{(n)} = 0$ indicates that the person n has not yet been observed. At the kth slice, there can be up to k persons, so we need $z_k^{(n)}$ for $n = 1, \ldots, k$. Note, that the auxiliary variables provide immediate 'lookup' reference to the information that is already encoded by $s_{1:k}$.

b) One-step generation: The counter is initialized $c_0 = 0$ and our model generates observations one-by-one. To generate y_k, $k \geq 1$, we select a label s_k. People enter field of view irregularly, so we choose uniformly between observing one of the known c_{k-1} persons or a new one;

$$s_k \sim \text{Uniform}(1, \ldots, c_{k-1}, c_{k-1} + 1). \quad (2)$$

The uniform distribution is one choice; other distributions are also applicable. Given the label we deterministically update the counter and pointers

$$c_k = c_{k-1} + [s_k > c_{k-1}], \quad (3)$$

$$z_k^{(k)} = 0, \quad (4)$$

$$z_k^{(n)} = z_{k-1}^{(n)} [s_{k-1} \neq n] + (k-1)[s_{k-1} = n], \quad (5)$$

where $n = 1, \ldots, k-1$. The symbol $[f]$ is a binary indicator; $[f] \equiv 1$ iff the binary proposition f is true, and $[f] \equiv 0$ otherwise. If the label indicates a new person, $s_k = c_{k-1} + 1$, then the counter has to be increased, as in (3). The pointers summarize associations before slice k, so we update them using previous label s_{k-1}, as in (4)–(5). A person labeled as k cannot be observed before slice k, so the pointer to his/her last observation $z_k^{(k)}$ is set to zero. The pointer to the last observation of the nth, $n < k$, person either does not change or we set it to the index of

the preceding observation, $k-1$, only if the label of this observation was $s_{k-1} = n$.

The second step is generating the latent state x_k of the person indicated by s_k. If this person has been already observed, then the index of his/her last observation $z_k^{(s_k)}$ is non-zero. By our assumption the state does not change, so we copy x from the previous instance. If the current person is observed for the first time, $z_k^{(s_k)} = 0$, then we sample the state from the prior $\pi(x)$;

$$x_k = x_{z_k^{(s_k)}}[z_k^{(s_k)} > 0] + x^{\text{new}}[z_k^{(s_k)} = 0], \quad (6)$$

$$x^{\text{new}} \sim \pi(x). \quad (7)$$

The final step is rendering the observation $y_k = \{o_k, d_k\}$ given the latent state, (i.e. the parameters of a Gaussian kernel) $x_k = \{\mathbf{m}_k, \mathbf{V}_k\}$ and the pointer to the past spatio-temporal features of the current person, $z_k^{(s_k)} = i$;

$$o_k \sim \mathcal{N}(\mathbf{m}_k, \mathbf{V}_k), \quad (8)$$

$$d_k \sim P_\delta(d_k|d_i)[i > 0] + P_{\delta_0}(d_k)[i = 0]. \quad (9)$$

B. Graphical model

Figure 3 shows the Dynamic Bayesian Network representing our model, where we have used a variable $h_k \equiv \{s_k, c_k, z_k^{(1)}, \ldots, z_k^{(k)}\}$ to denote the discrete association variables at the kth slice.

From Fig. 3 we realize that the association variables evolve as a Markov process with transitions $P(h_k|h_{k-1})$ following from (2)–(5). The auxiliary variables evolve deterministically when conditioned on the labels.

The state evolutions $x_{1:k}$ resemble a Dirichlet process (DP) mixture model [1]. Such models are applied for inference of mixture distributions with unknown number of components. A DP mixture model defines the prior for parameters of a component density as a mixture of a global prior and Delta densities centered around parameters of other components. To see the relationship with our model, consider state prediction $p(x_k|x_{1:k-1}, h_k)$ following from equations (6)–(7): when we integrate over $z_k^{(s_k)}$, then the state x_k will be distributed according to a density which is a mixture of the prior $\pi(x_k)$ and $k-1$ Delta distributions $\delta(x_k - x_j)$, where $j = 1, \ldots, k-1$. However, the model is not identical with DP because the spatio-temporal features affect the predictive distribution for states.

C. Online tracking by probabilistic filtering

When an observation y_k of some person arrives, we solve association by probabilistic filtering, i.e., computing the filtered distribution of the latent variables conditioned on all available data $y_{1:k}$. From this distribution we find the most likely label s_k to identify the current person. Figure 3 reveals that latent states are not Markovian: the influence of past $y_{1:k-1}$ is mediated through all variables $x_{1:k-1}$, therefore the filtered density takes the form $p(x_{1:k}, h_k|y_{1:k})$.

Unfortunately, this density cannot be computed exactly, what is a typical problem with hybrid (state-space) models

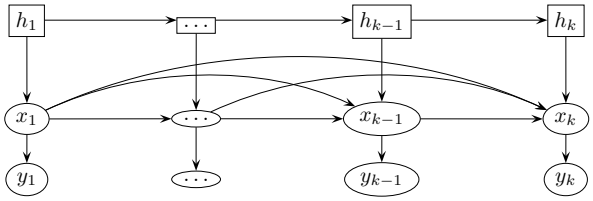


Fig. 3. A graphical model, represented as Dynamic Bayesian Network. The latent variables are $x_{1:k}$, $h_{1:k}$, the observed $y_{1:k}$. For clarity, the dependencies induced by spatio-temporal features are not shown.

(e.g. Switching Kalman Filter). A convenient on-line approximation method is assumed-density filtering [4] (ADF), where the filtered density is approximated with a factorial family. We choose a family

$$p(x_{1:k}, h_k|y_{1:k}) \approx q(s_k, c_k) \prod_{i=1}^{k} q_k(x_i) q(z_k^{(i)}) \quad (10)$$

that factorizes the discrete variables from the continuous. Approximating the joint distribution on h_k with a product of simpler models sidesteps maintaining a large table with probabilities for every combination of their state. The state is represented with a 'Normal-Inverse Wishart' family; $q_k(x_i) = \phi(x_i|\theta_{i,k})$, where $\theta_{i,k}$ are hyperparameters specific to the ith kernel after k filtering steps (c.f. (1)). (This family is conjugate to the Normal density [11].)

One-step filtering: When y_k arrives we have to compute the assumed approximation to the filtering density. First, we find a predictive density

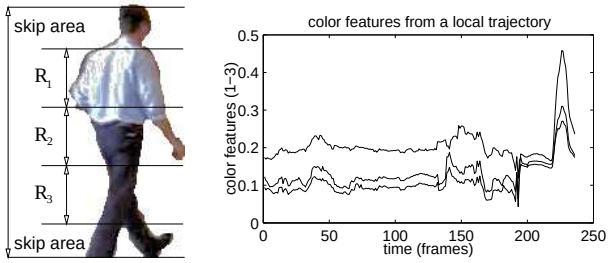
$$p_{\text{r}}(x_{1:k}, h_k) = \sum_{h_{k-1}} p(h_k|h_{k-1}) p(x_k|x_{1:k-1}, h_k) \times p(x_{1:k-1}, h_{k-1}|y_{1:k-1}), \quad (11)$$

where the last term comes from the previous filtering step. The filtered density is found by updating

$$p(x_{1:k}, h_k|y_{1:k}) \propto p(y_k|x_k, h_k) p_{\text{r}}(x_{1:k}, h_k) \quad (12)$$

where $p(y_k|x_k, h_k)$ follows from (8)–(9). The dependency of y_k on spatio-temporal features $d_{1:k-1}$ is not written explicitly, however it is always assumed. The term (12) does not belong to the assumed family. ADF projects it to such member of the family that offers the closest approximation in the Kullback-Leibler (KL) sense. The nearest in the KL-sense factorial distribution is the product of marginals [4], so we recover the representation (10) by computing the marginals of (12).

The detailed marginals are provided in [20]. We note that the marginalization is efficient for two reasons. First, our model is sparse. Second the auxiliary variables evolve deterministically when conditioned on labels.



a) b)

Fig. 4. a) Appearance features of a person. The features are defined as three color vectors (R,G,B), where each color vector was the average pixel color in a corresponding horizontal region R_i, $i = 1, 2, 3$ in the person's image. The height of each region is 25% of the blob's height. The "skip areas" are meant to remove parts that have little discriminative power. Their height was set to 12.5% of the total height. b) The first three components of the feature vector computed from subsequent blobs of a local trajectory.

V. Experiments

We have tested our method using video clips recorded by a simple mobile platform equipped with a color camera and odometers. The clips contain observations of people who were followed by a robot in three places with different illumination conditions in an office-like environment.

Background substraction: We allowed for online learning of background model when the platform was static. Our implementation [21] sets the learning rate so that an object is considered background only when it remains static for more than 220 frames (15s at 15 fps).

Color features: Every local trajectory y_k includes multiple frames when a person was detected in the field of view. We have taken the appearance features o_k to be the averages over features computed form multiple frames. In every frame, the blob representing a person is split into three fixed regions as in Fig. 4. For each region we computed the 3D average color, obtaining in total a 9D feature. The regions are our heuristic for describing people that provide a low-dimensional summary of color content and its geometrical layout (unlike color histograms). In order to compute color features that are independent from illuminating light, we have transformed the original RGB pixels into a color-channel normalized space [8].

Prior state distribution: Our method requires a prior distribution $\pi(x_k)$ on states i.e, means and covariances of Gaussian kernels. This distribution $\pi(x)$ is a 'Normal-Inverse Wishart' density. We set its parameters $\theta_0 = \{\mathbf{a}_0, \kappa_0, \eta_0, \mathbf{C}_0\}$ as follows: the expected features $\mathbf{a}_0 = \mathbf{0}_{9\times 1}$ (9 dimensional zero vector); the scale $\kappa_0 = 100$; the degrees of freedom $\eta_0 = 9$ (dimensionality of observations), the matrix $\mathbf{C}_0 = 10^{-3}\mathbf{I}_{9\times 9}$, where $\mathbf{I}_{9\times 9}$ is a 9 dimensional identity matrix. Parameter $\mathbf{C}_0$ with small eigenvalues indicates that we are expecting relatively 'sharp' kernels. Since the means $\mathbf{m}$ are not known, we set the scale κ_0 to a large value.

Spatio-temporal features: In the experiments we have used a simple Markov chain model for spatio-temporal features d_k of a local trajectory o_k. The prior density P_{δ_0} was uniform. The transition model P_δ was set to prevent starting a new local trajectory of a person before his/her previous trajectory had finished: $P_\delta(d_i|d_j) = 0$ iff $t_i^{\mathrm{e}} \leq t_j^{\mathrm{q}}$, and $P_\delta(d_i|d_j) = 1$ otherwise.

Results: The method is evaluated by measuring the number of mis-identified observations (identification error) and the error in the number of distinct people recognized from the data. In the first video clip all persons were correctly detected, tracked and identified. Figure 5 shows several frames from the clip with identities of detected people given by numerical labels. In this clip persons were observed in two places with similar illumination conditions. During the second video clip, our platform traveled a longer distance, between two places where the illumination conditions changed significantly. In this case one out of five encountered persons was wrongly identified as a new individual (rather than a previously observed one). Sample frames from this clip are shown in Fig. 6. The complete set of labeled clips is available at the following web address: `http:\\www.science.uva.nl\~wzajdel\Icra`.

Our experiments simulate home-like environments, where the robot has to cope with a small number of distinct persons. The tests show, that although the robots postulates a new person with every new local trajectory, the model is able to estimate the number of distinct people when the illumination changes slowly.

VI. Conclusions and further work

Keeping track of identities of multiple persons observed from a mobile robotic platform is a complex task that requires detecting people with a mobile camera, local tracking within filed of view and identification when people re-enter the field of view. We have presented a system that combines various visual cues and robot's odometric data to achieve detection and local tracking that is robust to occlusions, slow illumination changes and non-static (but rigid) background scenes. We have also presented a Bayesian algorithm for visual identification of people that left the field of view and reappeared later. The algorithm relies on spatio-temporal and visual cues to distinguish between people. In the presented experiments we have shown that in an office-like scenario already weak motion constraints lead to accurate associations under slowly changing illumination. The experiments also reveal that the major challenge for vision-based identification are varying illumination conditions. The future extensions of the method have to address this problem, by e.g. using texture-based features to describe persons appearance.

Acknowledgment

The work of ZZ and BK described in this paper was conducted within the EU Integrated Project COGNIRON ("The

Fig. 5. Selected frames (353, 618, 2272, 2847) from the first test sequence. Each frame shows the estimated bounding box and estimated label of every detected person. Objects with the same label correspond to the same person. We see that all detected persons are correctly associated.

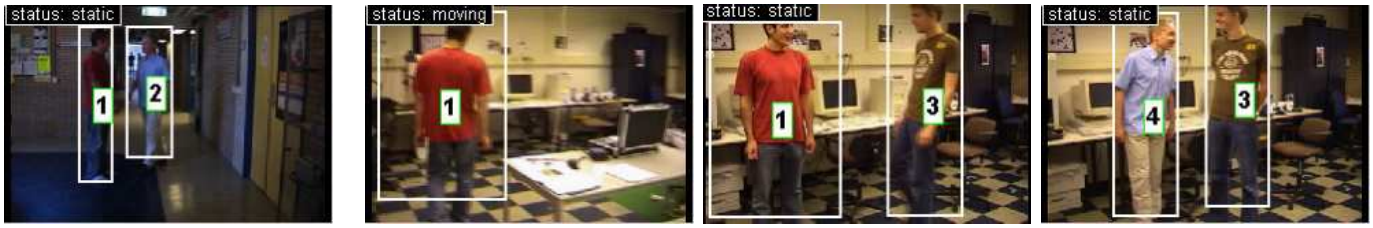


Fig. 6. Selected frames (1017, 2083, 2679, 3158) from the second test sequence. In this sequence illumination conditions changed significantly between initial location (the leftmost frame) and final location (other frames). Boxes labeled as "2" and "4" represent in fact the same person. The other persons are correctly identified.

Cognitive Companion") and was funded by the European Commission Division FP6-IST Future and Emerging Technologies under Contract FP6-002020. WZ was supported by the STW foundation under project ANN.5312.

REFERENCES

[1] C. E. Antoniak. Mixtures of Dirichlet processes with applications to Bayesian nonparametric problems. *Annals of Statistics*, 2:1152–1174, 1974.

[2] K. Arras, R. Philippsen, N. Tomatis, M. de Battista, M. Schilt, and R. Siegwart. A navigation framework for multiple mobile robots and its application at the expo.02 exhibition. In *Proc. of the IEEE Int. Conference on Robotics and Automation (ICRA)*, 2003.

[3] H. Asoh, N. Vlassis, Y. Motomura, F. Asano, I. Hara, S. Hayamizu, K. Itou, T. Kurita, T. Matsui, R. Bunschoten, and Ben Kröse. Jijo-2: An office robot that communicates and learns. *IEEE Intelligent Systems*, 5(16):46–55, 2001.

[4] Xavier Boyen and Daphne Koller. Tractable inference for complex stochastic processes. In *Proc. of Conf. on Uncertainty in Artificial Intelligence (UAI)*, pages 33–42. Morgan Kaufman, 1998.

[5] G. Cielniak, M. Bennewitz, and W. Burgard. Where is ...? Learning and utilizing motion patterns of persons with mobile robots. In *Proc. of Int. Joint Conference on Artificial Intelligence (IJCAI)*, 2003.

[6] A. J. Davison, M. Montemerlo, J. Pineau, N. Roy, S. Thrun, and V. Verma. Experiences with a mobile robotic guide for the elderly. In *Proc. of the AAAI National Conf. on Artificial Intelligence*, 2002.

[7] D.Comaniciu, V.Ramesh, and P.Meer. Real-time tracking of non-rigid objects using mean shift. *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2:142–149, 2000.

[8] M.S. Drew, J. Wei, and Z.N. Li. Illumination-invariant color object recognition via compressed chromaticity histograms of color-channel-normalized images. In *Proc. of Int. Conf. on Computer Vision (ICCV)*, pages 533–540, 1998.

[9] J. Fritsch, M. Kleinehagenbrock, S. Lang, T. Botz, G. A. Fink, , and G. Sagerer. Multi-modal anchoring for human-robot-interaction. *Robotics and Autonomous Systems, Special issue on Anchoring Symbols to Sensor Data in Single and Multiple Robot Systems*, 43(2–3):133–147, 2003.

[10] D. M. Gavrila. Pedestrian detection from a moving vehicle. In *Proc. of European Conf. on Computer Vision (ECCV)*, pages 37–49, 2000.

[11] Andrew Gelman, John B. Carlin, Hal S. Stern, and Donald D. Rubin. *Bayesian Data Analysis*. Chapman & Hall, 1995.

[12] Timothy Huang and Stuart Russell. Object identification: A Bayesian analysis with application to traffic surveillance. *Artificial Intelligence*, 103(1–2):1–17, 1998.

[13] K.Fukunaga and L.D.Hostetler. The estimation of the gradient of a density function, with applications in pattern recognition. *IEEE Trans. Information Theory*, 21:32–40, 1975.

[14] A. Prati, I. Mikic, M. Trivedi, and R. Cucchiara. Detecting moving shadows: Formulation, algorithms and evaluation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(7):918–924, 2003.

[15] D. Schulz, W. Burgard, D. Fox, and A.B. Cremers. People tracking with a mobile robot using sample-based joint probabilistic data association filters. *Int. Journal of Robotics Research*, 22(2), 2003.

[16] J. Sivic, F. Schaffalitzky, and A. Zisserman. Object level grouping for video shots. *In Proceedings European Conference Computer Vision (ECCV)*, 2004.

[17] C. Stauffer and W.E.L. Grimson. Adaptive background mixture models for real-time tracking. *In Proceedings CVPR*, pages 246–252, 1999.

[18] S. Thrun, M. Bennewitz, W. Burgard, A.B. Cremers, F. Dellaert, D. Fox, D. Hahnel, C.R. Rosenberg, N. Roy, J. Schulte, and D. Schulz. MINERVA: A second generation mobile tour-guide robot. In *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*, 1999.

[19] A.J.N van Breemen, K. Crucq, B.J.A Kröse, M. Nuttin, J.M.Porta, and E. Demeester. A user-interface robot for ambient intelligent environments. In *Proc. of the 1st Int. Workshop on Advances in Service Robotics, (ASER)*, pages 132–139. Fraunhofer IRB Verlag, 2003.

[20] W. Zajdel, A.T. Cemgil, and B. Kröse. Online multicamera tracking with a switching state-space model. In *Proc. of Int. Conf. on Pattern Recognition (ICPR)*, 2004.

[21] Z. Zivkovic. Improved adaptive gausian mixture model for background subtraction. In *Proc. of Int. Conf. on Pattern Recognition (ICPR)*, 2004.

[22] Z. Zivkovic and B. Krose. An em-like algorithm for color-histogram-based object tracking. *In Proceedings Computer Vision and Pattern Recognition (CVPR)*, 2004.

In Proc. IEEE Conference Computer Vision Pattern Recognition, 2004.

An EM-like algorithm for color-histogram-based object tracking

Zoran Zivkovic Ben Kröse

Intelligent and Autonomous Systems Group
University of Amsterdam
The Netherlands
email:{zivkovic,krose}@science.uva.nl

Abstract

The iterative procedure called 'mean-shift' is a simple robust method for finding the position of a local mode (local maximum) of a kernel-based estimate of a density function. A new robust algorithm is given here that presents a natural extension of the 'mean-shift' procedure. The new algorithm simultaneously estimates the position of the local mode and the covariance matrix that describes the approximate shape of the local mode. We apply the new method to develop a new 5-degrees of freedom (DOF) color histogram based non-rigid object tracking algorithm.

1. Introduction

Visual data is often complex and there are usually many data points that are not well explained by the applied models. In order to deal with the outliers robust estimation techniques are very important for solving vision problems [8]. Vision problems are often very specific and the methods from robust statistics [7] need to be modified in such a way that they are made appropriate for vision problems. A robust method that is often used for solving vision problems [3, 2, 1] is the 'mean-shift' procedure [9]. Data samples are used to get a kernel based approximation of the probability density function [17]. The mean-shift algorithm is a procedure to search for a local mode of the empirical density function. The position of the local mode is known to be very tolerant to outliers.

Efficient color-histogram-based tracking presented in [3] is based on the mean-shift procedure. Color histogram is a very robust representation of the object appearance [16]. In [3] the shape of the tracked non-rigid object is represented by an ellipse. A similarity function is defined between the color histogram of the object and the color histogram of a candidate ellipsoidal region from a new image from an image sequence. The mean-shift procedure is used to find the region in the new image that is the most similar to the object. See section 5 and [3] for more details. The problem of adapting the ellipse that approximates the shape of the object when the shape and the size of the tracked object change remained unsolved. Some local shape descriptors were used in [5]. In [3] after each tracking step the ellipse is adapted by checking a +10% larger and a -10% smaller ellipse and choosing the best one. In [14] an extensive search is performed within a range of scales of the ellipse.

In this paper we present an extension of the mean-shift algorithm. Instead of only estimating the position of a local mode the new algorithm simultaneously estimates the covariance matrix that describes the shape of the local mode. This is illustrated in figure 1. Further, we show how the algorithm can be applied to color-histogram-based object tracking in a similar way as in [3]. We propose a 5-DOF color-histogram-based tracking method that estimates the position of the tracked object but also simultaneously estimates the ellipse that approximates the shape of the object. The new algorithm solves the mentioned problem of adapting the ellipse in an efficient way.

The paper is organized as follows. In section 2 we introduce mean-shift as a robust estimation technique. In section 3 we present how the mean-shift can be viewed as an EM-like algorithm. In section 4 we extend the EM-like algorithm to estimate also the local scale. In section 5 we apply the new algorithm to color histogram based tracking. Some experiments are given in section 6 and in section 7 we report some conclusions.

2. Extreme outlier model

We will denote a data set of N independent samples by $\mathcal{X} = \{\vec{x}_1, ..., \vec{x}_N\}$. Let us assume that the probability density function $p(\vec{x})$, for example a Gaussian $p(\vec{x}) = \mathcal{N}(\vec{x}; \vec{\theta}, V)$, is a good generative model for our data. Maximum Likelihood (ML) estimates for the mean vector $\vec{\theta}$ and the covariance matrix V are the values that maximize the likelihood function $\prod_{i=1}^{N} p(\vec{x}_i)$. Often in practice we are confronted with a data set which is polluted by some outliers. Uniform distribution $1/A$ (where A is the area of the domain of $\vec{x}$)

can be used to model the outliers. If e presents the probability that a data sample is an outlier, we can write a common generative model, that takes into account the outliers, as:

$$p'(\vec{x}_i) = e/A + (1-e)p(\vec{x}_i). \quad (1)$$

The likelihood of the data is now $\prod_{i=1}^{N} p'(\vec{x}_i)$ and Taylor expansion of the likelihood in $(1-e)$ is given by:

$$(e/A)^N + (e/A)^{N-1}(1-e)\sum_{i=1}^{N} p(\vec{x}_i) + O((1-e)^2). \quad (2)$$

In an extreme case where there are a lot of outliers, e is close to 1 and only the first two terms matter [12]. The first term is constant and the ML estimates are obtained by maximizing $\sum_{i=1}^{N} p(\vec{x}_i)$. For Gaussian p, the objective function to be maximized can be written as:

$$f(\vec{\theta}, V) = \sum_{i=1}^{N} \mathcal{N}(\vec{x}_i; \vec{\theta}, V). \quad (3)$$

Given a fixed V and if we add $1/N$ in front, (3) resembles an empirical density estimate using Gaussian kernels, where V can be regarded as the bandwidth factor [17]. The mean-shift can be used to get a robust estimate of $\vec{\theta}$ - the mode of this empirical density function. In section 4 of this paper we show how to get also robust estimates for V using this extreme outlier model. Vision problems often involve analyzing only a local part of an image and disregarding the data from the rest of the image regardless of how large the image is. The extreme outlier model is obviously appropriate for such problems.

The robust statistics procedure called 'iteratively reweighted least squares' (IRLS) [6] is very similar to the mean-shift procedure. In fact we can see the mean-shift as a version of IRLS for the extreme outlier model. In a similar way, the new procedure we present here is a special version of the robust scale estimators [11]. We mentioned that V in (3) can be regarded as the bandwidth factor in kernel-based density estimation. However, the objective in bandwidth estimation [17] is quite different.

3. Mean-shift as an EM-like algorithm

If each data point has also a weight factor ω_i, a more general version of (3) is given by:

$$f(\vec{\theta}, V) = \sum_{i=1}^{N} \omega_i \mathcal{N}(\vec{x}_i; \vec{\theta}, V). \quad (4)$$

We would like to find the parameters $\vec{\theta}$ and V for which the maximum value of (4) is achieved. This can be done iteratively using EM-like iterations [4, 13]. From the Jensen's inequality we get:

$$\log f(\vec{\theta},V) \geq G(\vec{\theta},V,q_1,...,q_N) = \sum_{i=1}^{N} \log\left(\frac{\omega_i \mathcal{N}(\vec{x}_i;\vec{\theta},V)}{q_i}\right)^{q_i} \quad (5)$$

where q_i-s are arbitrary constants that meet the following requirements:

$$\sum_{i=1}^{N} q_i = 1 \text{ and } q_i \geq 0. \quad (6)$$

Let us assume that the current estimate values of the parameters are denoted by $\vec{\theta}^{(k)}$ and $V^{(k)}$. The E and M steps described below are repeated then until convergence:

1. E step: find q_i-s to maximize G while keeping $\vec{\theta}^{(k)}$ and $V^{(k)}$ fixed. It is easy to show that the maximum (equality sign in (5)) is achieved for:

$$q_i = \frac{\omega_i \mathcal{N}(\vec{x}_i; \vec{\theta}^{(k)}, V^{(k)})}{\sum_{i=1}^{N} \omega_i \mathcal{N}(\vec{x}_i; \vec{\theta}^{(k)}, V^{(k)})}. \quad (7)$$

2. M step: maximize G from (5) with respect to $\vec{\theta}$ and V while keeping q_i-s constant. The q_i-s are now fixed we need to minimize only a part of G that depends on the parameters:

$$g(\vec{\theta}, V) = \sum_{i=1}^{N} q_i \log \mathcal{N}(\vec{x}_i; \vec{\theta}, V). \quad (8)$$

From $\frac{\partial}{\partial \vec{\theta}} g(\vec{\theta}, V) = 0$ we get:

$$\vec{\theta}^{(k+1)} = \sum_{i=1}^{N} q_i \vec{x}_i = \frac{\sum_{i=1}^{N} \vec{x}_i \omega_i \mathcal{N}(\vec{x}_i; \vec{\theta}^{(k)}, V^{(k)})}{\sum_{i=1}^{N} \omega_i \mathcal{N}(\vec{x}_i; \vec{\theta}^{(k)}, V^{(k)})} \quad (9)$$

Note that this update equation for the position estimate is equivalent to the 'mean shift' update equation for the Gaussian kernels. For other kernel types this might be different. This new EM-like view of the problem will lead to update equations for V as described next.

4. Scale selection

If $p^*(\vec{x})$ is the true distribution of the data, the expected value of (3) is:

$$E\left[f(\vec{\theta},V)\right] = \int_{\vec{x}} p^*(\vec{x}) \mathcal{N}(\vec{x}; \vec{\theta}, V). \quad (10)$$

This can be seen as a smoothed version of the original p^* and the maximum with respect to V does not have some desirable properties [15]. For example, if p^* is locally a Gaussian $\mathcal{N}(\vec{x}; \vec{\theta}^*, V^*)$, the expected value (10) is a smoothed Gaussian $\mathcal{N}(\vec{x}; \vec{\theta}^*, V^* + V)$. The expected maximum is for $\vec{\theta} = \vec{\theta}^*$, but unfortunately for the trivial value $V = 0$ since the value at the local mode is decreasing with larger V. We

normalize the result by multiplying density estimate (4) by $|V|^{\gamma/2}$ and we get what we can call a 'γ-normalized' function:

$$f_\gamma(\vec{\theta}, V) = |V|^{\gamma/2} f(\vec{\theta}, V). \quad (11)$$

Under the same assumption that the local mode is approximately a Gaussian, the value at the mode will now be proportional to $|V|^{\gamma/2}/|V^*{+}V|^{1/2}$. The maximum with respect to V is at:

$$\frac{\partial}{\partial V} \frac{|V|^{\gamma/2}}{|V^*{+}V|^{1/2}} = 0 \quad (12)$$

Since $\frac{\partial}{\partial V}|V| = |V| \left[2V^{-1} - \mathrm{diag}(V^{-1})\right]$ we get:

$$\gamma|V|^\gamma \left[2V^{-1} - \mathrm{diag}(V^{-1})\right] |V^*{+}V| - |V|^\gamma |V^*{+}V| \left[2(V^*{+}V)^{-1} - \mathrm{diag}((V^*{+}V)^{-1})\right] = 0. \quad (13)$$

From here we get $\gamma V^{-1} = (V^*{+}V)^{-1}$ and $V = \frac{\gamma}{1-\gamma} V^*$. Obviously only for $\gamma \epsilon (0, 1)$ we get a positive value. For $\gamma = 1/2$ it follows that expected maximum is for $V = V^*$. The solution using the γ-normalized function is not biased and this is a desirable property of an estimation algorithm.

The extreme outlier model in the limit case can be explained also as a model where only one observation is not an outlier [12]. Then it is understandable that V can not be estimated reliably using this model. The γ-normalization can be seen as introducing a certain informative prior for V to regularize the solution and get non-biased estimates. Another interesting connection is with some image filtering algorithms. For example, in [10] γ-normalized image convolution was studied for selecting the scale of the filtering operator. If we have a 2D case and we replace p^* in (10) with an image, the connection with the image convolution is evident.

The EM-like iterative algorithm from the previous section can be applied to the γ-normalized function. The only difference is in the M-step. Instead of (8) we have now:

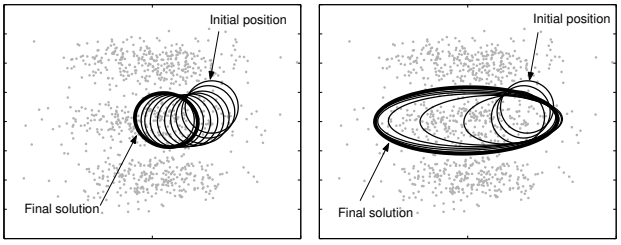
$$g(\vec{\theta}, V) = \sum_{i=1}^{N} q_i \log |V|^{\gamma/2} \mathcal{N}(\vec{x}; \vec{\theta}, V). \quad (14)$$

The position update equation (9) stays the same. From $\frac{\partial}{\partial V} g(\vec{\theta}, V) = 0$ it is easy to show that the update equation for V in the M-step is given by:

$$\vec{V}^{k+1} = \beta \sum_{i=1}^{N} q_i (\vec{x}_i - \vec{\theta}^{(k)})(\vec{x}_i - \vec{\theta}^{(k)})^T, \quad (15)$$

where $\beta = 1/(1-\gamma)$.

In figure 1 an example is shown to illustrate the performance of the new algorithm. The simulated data consists of 600 samples generated using a mixture of three Gaussian distributions. The three modes are clearly visible in figure 1. The iterations (the 2-sigma contours of the estimated Gaussian) of the mean-shift procedure are plotted in figure 1a. In figure 1b we show the iterations of the new EM-like algorithm with $\gamma = 1/2$ ($\beta = 2$). We can observe how the new algorithm simultaneously estimates both the position of the local mode and the covariance matrix that describes the shape of the mode. Note that $\beta = 2$ is appropriate if the underlying distribution is Gaussian. If some other distribution is approximated by a Gaussian some other value for β might be needed in order to avoid biased solution. Similar parameter and similar discussion is also given in the standard robust statistics methods [11, 7]. The difference is that the results we present here are for the extreme outlier model.



a) mean-shift iterations b) EM-shift iterations

Figure 1: Performance of the two algorithms on simulated 2D data.

5. Color histogram tracking

We assume that the shape of a non-rigid object is approximated by an ellipsoidal region in an image. Initially the object is selected manually or detected using some other algorithm, background subtraction for example. Let $\vec{x}_i$ denote a pixel location and $\vec{\theta}_0$ the initial location of the center of the object in the image. The second order moment can be used to approximate the shape of the object:

$$V_0 = \sum_{\text{all the pixels that belong to the object}} (\vec{x}_i - \vec{\theta}_0)(\vec{x}_i - \vec{\theta}_0)^T. \quad (16)$$

Further, the color histogram is used to model the object appearance. Let the histogram have M bins and let the function $b(\vec{x}_i) : R^2 \rightarrow 1, ..., M$ be the function that assigns a color value of the pixel at location $\vec{x}_i$ to its bin. The color histogram model of the object consists then of the M values of the M bins of the histogram $\vec{o} = [o_1, ..., o_M]^T$. The value of the m-th bin is calculated by:

$$o_m = \sum_{i=1}^{N_{V_0}} \mathcal{N}(\vec{x}_i; \vec{\theta}_0, V_0) \delta \left[b(\vec{x}_i) - m\right], \quad (17)$$

where δ is the Kronecker delta function. We use the Gaussian kernel $\mathcal{N}$ to rely more on the pixels in the middle of the object and to assign smaller weights to the less reliable pixels at the borders of the objects. We use only the N_{V_0} pixels from a finite neighborhood of the kernel and the pixels further than 2.5-sigma are disregarded.

5.1. Similarity measure

Let us assume that we have a new image from an image sequence and the object we are tracking is present in the image. The goal of a tracking algorithm is to find the object in the new image. Let an ellipsoidal region in the new image be defined by its position $\vec{\theta}$ and its shape described by the covariance matrix V. The color-histogram that describes the appearance of the region is $\vec{r}(\vec{\theta}, V)$ and the value of the m-th bin is calculated by:

$$r_m(\vec{\theta}, V) = \sum_{i=1}^{N_V} \mathcal{N}(\vec{x}_i; \vec{\theta}, V)\delta\left[b(\vec{x}_i) - m\right]. \qquad (18)$$

The similarity of the region to the object is defined by the similarity of their histograms. As in as in [3] we use Bhattacharyya coefficient as a measure of similarity between two histograms:

$$\rho\left[\vec{r}(\vec{\theta}, V), \vec{o}\right] = \sum_{m=1}^{M} \sqrt{r_m(\vec{\theta}, V)}\sqrt{o_m}. \qquad (19)$$

The first order Taylor approximation around the current estimate $\vec{r}(\vec{\theta^{(k)}}, V^{(k)})$ is given by:

$$\rho\left[\vec{r}(\vec{\theta}, V), \vec{o}\right] \approx c_1 + c_2 \sum_{i=1}^{N_V} \omega_i \mathcal{N}(\vec{x}_i; \vec{\theta}, V), \qquad (20)$$

where c_1 and c_2 are some constant factors and

$$\omega_i = \sum_{m=1}^{M} \sqrt{\frac{o_m}{r_m(\vec{\theta}^{(k)}, V^{(k)})}}\delta\left[b(\vec{x}_i) - m\right]. \qquad (21)$$

Since the last term in (20) has the same form as (4) we can use the new EM-like algorithm to search for the local maximum of the similarity function (20). The weights are recalculated before each iteration using (21) and then the update is done using (7),(9) and (15). Some practical issues are presented next.

5.2. Practical algorithm

For the sake of clarity we present here the whole algorithm:

Input: the object model $\vec{o}$, its initial ($k = 0$) location $\vec{\theta}^{(k)}$ and shape defined by $V^(k)$.

1. Compute the values of the color histogram of the current region defined by $\vec{\theta}^{(k)}$ and $V^{(k)}$ from the current frame using (18).
2. Calculate weights using (21).
3. Calculate q_i-s using (7).
4. Calculate new position estimate $\vec{\theta}^{(k+1)}$ using (9).
5. Calculate new variance estimate $V^{(k+1)}$ using (15).
6. If no new pixels are included using the new elliptical region defined by the new estimates $\vec{\theta}^{(k+1)}$ and $V^{(k+1)}$ stop, otherwise set $k \leftarrow k + 1$ and go to 1.

The procedure is repeated for each frame. In the simplest version the position and shape of the ellipsoidal region from the previous frame are used as the initial values for the new frame.

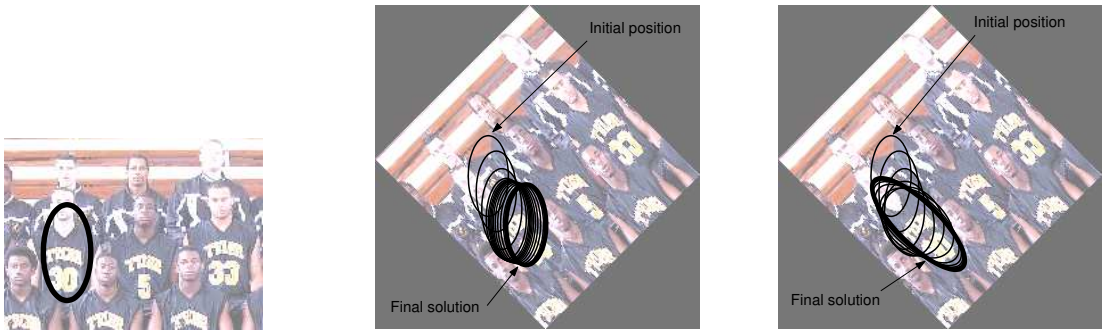
The function (20) that is regarded as the underlying density function is not a Gaussian. We also used the approximation that the weights ω_i are constant during one iteration. The maximum of (20) is well defined with respect to V for $\beta = 1$. However since we disregard the samples further than 2.5-sigma and it is easy to show that we should use $\beta \approx 1.1$. The correct value for the β depends on the noise that is present in the image sequence. Small errors in choice of β leads to slightly biased solution but since the ellipse is just an approximation of the shape this is acceptable.

Finally, because of the approximation that the weights ω_i are constant during one iteration the convergence proof does not hold. An additional line search should be performed to make sure that we increase the value of (19) as it was mentioned in [3]. However the approximation is usually good in the small neighborhood and this is not needed. This was also noted for the mean-shift algorithm presented in [3].

6. Experiments

The new 5-DOF color-histogram-based tracking was applied to a number of sequences and some results are reported in this section. The position and shape of the tracked objects is represented by the dashed ellipse.

First in figure 2 we illustrate the performance of the algorithm. A player is selected as indicated by the elliptical in figure 2a. For better presentation we increased the brightness of the images we present here. The original images was darker. The image is scaled 1.5 times in the vertical direction and then rotated for 45 degrees as presented in figure 2b. We use the initial shape of the region and we manually select a position in the new rotated and scaled image. The iterations and the final of the mean-shift procedure are presented in figure 2b. In figure 2c we present the iterations and the final solution of our algorithm. Both the new shape



a) the selected region b) 'mean-shift' search c) EM-shift search

Figure 2: The mean shift and the new EM-like algorithm

and the position are accurately estimated. The new elliptical region contains the same content as the content of the initial region.

The 'hall' sequence (figure 3)is a long video from a surveillance camera. Very hard lightning conditions are present. We used only H and S from HSV color space to be more robust to the light effects. The objects was represented using a 8×8 histogram in the HS space. Since the objects were walking people we did not expect the orientation of the ellipse that approximates the shape of the objects to be other than vertical. Therefore we constrained V to be diagonal. Two frames from that represent a typical situation from the video are presented in figure 3a. The object moves towards the camera and the size of the object changes considerably. Standard mean-shift tracking from [3] fails to adapt to these size changes. This is similar to the sequence that was used in [14]. Our algorithm has no problems with adapting as the much slower extensive search method from [14].

The 'PETS1' is a sequence from the standard data set from www.visualsurveillance.org. The covariance matrix is now not constrained to be diagonal since the vehicles are also changing orientation. We used RGB space and $8 \times 8 \times 8$ histogram. Two frames from the sequence are shown in 3b.

The 'hand' sequence is used to demonstrate the full 5-DOF color-histogram-based tracking. To be robust to light conditions we used again 8×8 histogram in the HS space. The hand is tracked. The sequence has 250 frames and the position and the shape of the hand are changing rapidly. In figure 3c we can see that the new algorithm can track the hand and also adapt to the shape of the object. Hand tracking was used for example in [18]. However the algorithm they used is not very robust and can be used only for single colored objects.

Finally, in figure 4 we present the number of iterations of the algorithm for the 'hand' sequence. The average number of iterations per frame was approximately 6. This is slightly more then 4 that was reported for the mean-shift based iterations in [3]. The computational complexity of one iteration of the new algorithm is slightly higher than the computational complexity of the mean-shift. On average our algorithm is around 2 times slower but still fast enough for real-time performance. In our current implementation the algorithm works comfortably in real-time on a 1GHz PC.

7. Conclusions

We presented a new 5-DOF color-histogram-based non-rigid object tracking. We demonstrated that he new algorithm can robustly track the objects in different situations. The algorithm can also adapt to changes in shape and scale of the object. The algorithm works in real-time and the computational cost is only slightly higher than for the previously proposed algorithms that had problems with shape and scale changes. The new color-histogram-based object tracking procedure is based on a natural extension of the mean-shift algorithm that can be useful also for many other vision problems. This is a topic of our further research.

Acknowledgments

The work described in this paper was conducted within the EU Integrated Project COGNIRON ("The Cognitive Companion") and was funded by the European Commission Division FP6-IST Future and Emerging Technologies under Contract FP6-002020.

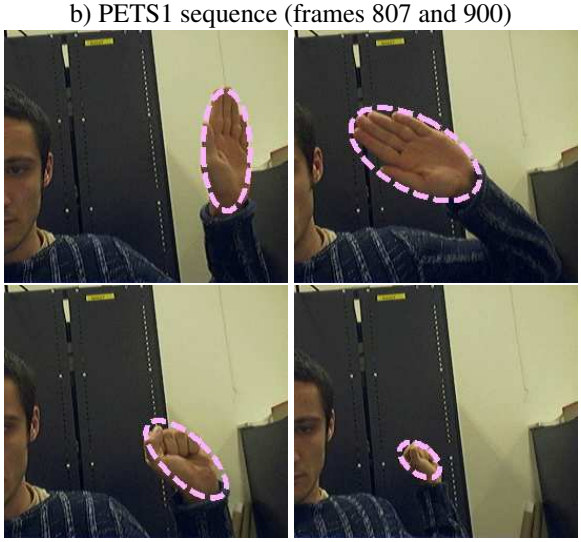
References

[1] H. Chen and P. Meer. Robust computer vision through kernel density estimation. *A. Heyden et al. (Eds.):ECCV 2002, LNCS2350*, pages 236–250, 2002.

[2] D.Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Trans. Pattern Anal. Machine Intell.*, 24(5), 2002.

a) hall sequence (frames 320 and 360)

b) PETS1 sequence (frames 807 and 900)



c) hand sequence (frames 0,100,200 and 250)

Figure 3: Some tracking results

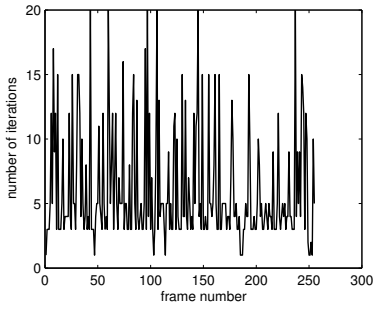


Figure 4: Number of iterations per frame for the hand sequence

[3] D.Comaniciu, V.Ramesh, and P.Meer. Real-time tracking of non-rigid objects using mean shift. *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2:142–149, 2000.

[4] A.P. Dempster, N. Laird, and D.B.Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society, Series B (Methodological)*, 1(39):1–38, 1977.

[5] G.R.Bradski. Computer vision face tracking as a component of a perceptual user interface. *Proc.IEEE Workshop on Applications of Computer vision*, pages 214–219, 1998.

[6] P. W. Holland and R. E. Welsch. Robust regression using iteratively reweighted least-squares. *Commun. Statist.-Theor. Meth.*, (6):813–827, 1977.

[7] P. Huber. *Robust Statistics*. Wiley, 1981.

[8] Special issue. Robust statistical techniques in image understanding. *Computer Vision and Image Understanding*, 2000.

[9] K.Fukunaga and L.D.Hostetler. The estimation of the gradient of a density function, with applications in pattern recognition. *IEEE Trans. Information Theory*, 21:32–40, 2002.

[10] T. Lindeberg. Feature detection with automatic scale selection. *International Journal of Computer Vision*, 30(2):77–116, 1998.

[11] R. Maronna. Robust M-estimators of multivariate location and scatter. *The Annals of Statistics*, 4(1):51–67, 1976.

[12] T. Minka. The 'summation hack' as an outlier model. *Tutorial note*, 2003.

[13] R. M. Neal and G. E. Hinton. A new view of the EM algorithm that justifies incremental, sparse and other variants. *In, M. I. Jordan editor, Learning in Graphical Models*, pages 355–368, 1998.

[14] R.Collins. Mean-shift blob tracking through scale space. *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2003.

[15] B.W. Silverman. *Density Estimation for Statistics and Data Analysis*. Monographs on Statistics and Applied Probability, Chapman and Hall, 1986.

[16] M. Swain and D. Ballard. Color indexing. *Intl. J. of Computer Vision*, 7(1):11–32, 1991.

[17] M. P. Wand and M. C. Jones. *Kernel Smoothing*. Chapman and Hall, 1995.

[18] C. Wren, A. Azarbayejani, T. Darrell, and A. Pentland. Pfinder: Real-time tracking of the human body. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):780–785, 1997.

In Proc. ICPR, 2004

Improved Adaptive Gaussian Mixture Model for Background Subtraction

Zoran Zivkovic

Intelligent and Autonomous Systems Group
University of Amsterdam, The Netherlands
email: zivkovic@science.uva.nl

Abstract

Background subtraction is a common computer vision task. We analyze the usual pixel-level approach. We develop an efficient adaptive algorithm using Gaussian mixture probability density. Recursive equations are used to constantly update the parameters and but also to simultaneously select the appropriate number of components for each pixel.

1. Introduction

A static camera observing a scene is a common case of a surveillance system. Detecting intruding objects is an essential step in analyzing the scene. An usually applicable assumption is that the images of the scene without the intruding objects exhibit some regular behavior that can be well described by a statistical model. If we have a statistical model of the scene, an intruding object can be detected by spotting the parts of the image that don't fit the model. This process is usually known as "background subtraction".

A common bottom-up approach is applied and the scene model has a probability density function for each pixel separately. A pixel from a new image is considered to be a background pixel if its new value is well described by its density function. For a static scene the simplest model could be just an image of the scene without the intruding objects. Next step would be for example to estimate appropriate values for the variances of the pixel intensity levels from the image since the variances can vary from pixel to pixel. This single Gaussian model was used in [1]. However, pixel values often have complex distributions and more elaborate models are needed. Gaussian mixture model (GMM) was proposed for background subtraction in [2]. One of the most commonly used approaches for updating GMM is presented in [3] and further elaborated in [10]. These GMM-s use a fixed number of components. We present here an improved algorithm based od the recent results from [12]. Not only the parameters but also the number of components of the mixture is constantly adapted for each pixel. By choosing the number of components for each pixel in an on-line procedure, the algorithm can automatically fully adapt to the scene.

The paper is organized as follows. In next section we list some related work. In section 3 the GMM approach from [3] is reviewed. In sections 4 we present how the number of components can be selected on-line and to improve the algorithm. In section 5 we present some experiments.

2. Related work

The value of a pixel at time t in RGB or some other colorspace is denoted by $\vec{x}^{(t)}$. Pixel-based background subtraction involves decision if the pixel belongs to background (BG) or some foreground object (FG). Bayesian decision R is made by:

$$R = \frac{p(BG|\vec{x}^{(t)})}{p(FG|\vec{x}^{(t)})} = \frac{p(\vec{x}^{(t)}|BG)p(BG)}{p(\vec{x}^{(t)}|FG)p(FG)} \qquad (1)$$

The results from the background subtraction are usually propagated to some higher level modules, for example the detected objects are often tracked. While tracking an object we could obtain some knowledge about the appearance of the tracked object and this knowledge could be used to improve background subtraction. This is discussed for example in [7] and [8]. In a general case we don't know anything about the foreground objects that can be seen nor when and how often they will be present. Therefore we set $p(FG) = p(BG)$ and assume uniform distribution for the foreground object appearance $p(\vec{x}^{(t)}|FG) = c_{FG}$. We decide then that the pixel belongs to the background if:

$$p(\vec{x}^{(t)}|BG) > c_{thr}(= Rc_{FG}), \qquad (2)$$

where c_{thr} is a threshold value. We will refer to $p(\vec{x}|BG)$ as the background model. The background model is estimated from a training set denoted as $\mathcal{X}$. The estimated model is denoted by $\hat{p}(\vec{x}|\mathcal{X}, BG)$ and depends on the training set as

denoted explicitly. We assume that the samples are independent and the main problem is how to efficiently estimate the density function and to adapt it to possible changes. Kernel based density estimates were used in [4] and we present here an improvement of the GMM from [3]. There are models in the literature that consider the time aspect of an image sequence and the decision depends also on the previous pixel values from the sequence. For example in [5, 11] the pixel value distribution over time is modelled as an autoregressive process. In [6] Hidden Markov Models are used. However, these methods are usually much slower and adaptation to changes of the scene is difficult.

Another related subject is the shadow detection. The intruding object can cast shadows on the background. Usually, we are interested only in the object and the pixels corresponding to the shadow should be detected [9]. In this paper we analyze the only basic pixel-based background subtraction. For various applications some of the mentioned additional aspects and maybe some postprocessing steps might be important and could lead to improvements but this is out of the scope of this paper.

3. Gaussian mixture model

In practice, the illumination in the scene could change gradually (daytime or weather conditions in an outdoor scene) or suddenly (switching light in an indoor scene). A new object could be brought into the scene or a present object removed from it. In order to adapt to changes we can update the training set by adding new samples and discarding the old ones. We choose a reasonable time period T and at time t we have $\mathcal{X}_T = \{x^{(t)}, ..., x^{(t-T)}\}$. For each new sample we update the training data set $\mathcal{X}_T$ and reestimate $\hat{p}(\vec{x}|\mathcal{X}_T, BG)$. However, among the samples from the recent history there could be some values that belong to the foreground objects and we should denote this estimate as $p(\vec{x}^{(t)}|\mathcal{X}_T, BG+FG)$. We use GMM with M components:

$$\hat{p}(\vec{x}|\mathcal{X}_T, BG+FG) = \sum_{m=1}^{M} \hat{\pi}_m \mathcal{N}(\vec{x}; \widehat{\vec{\mu}}_m, \widehat{\sigma}_m^2 I) \qquad (3)$$

where $\widehat{\vec{\mu}}_1, ..., \widehat{\vec{\mu}}_M$ are the estimates of the means and $\widehat{\sigma}_1, ..., \widehat{\sigma}_M$ are the estimates of the variances that describe the Gaussian components. The covariance matrices are assumed to be diagonal and the identity matrix I has proper dimensions. The mixing weights denoted by $\hat{\pi}_m$ are non-negative and add up to one. Given a new data sample $\vec{x}^{(t)}$ at time t the recursive update equations are [12]:

$$\hat{\pi}_m \leftarrow \hat{\pi}_m + \alpha(o_m^{(t)} - \hat{\pi}_m) \qquad (4)$$

$$\widehat{\vec{\mu}}_m \leftarrow \widehat{\vec{\mu}}_m + o_m^{(t)}(\alpha/\hat{\pi}_m)\vec{\delta}_m \qquad (5)$$

$$\widehat{\sigma}_m^2 \leftarrow \widehat{\sigma}_m^2 + o_m^{(t)}(\alpha/\hat{\pi}_m)(\vec{\delta}_m^T\vec{\delta}_m - \widehat{\sigma}_m^2), \qquad (6)$$

where $\vec{\delta}_m = \vec{x}^{(t)} - \widehat{\vec{\mu}}_m$. Instead of the time interval T that was mentioned above, here constant α describes an exponentially decaying envelope that is used to limit the influence of the old data. We keep the same notation having in mind that approximately $\alpha = 1/T$. For a new sample the ownership $o_m^{(t)}$ is set to 1 for the 'close' component with largest $\hat{\pi}_m$ and the others are set to zero. We define that a sample is 'close' to a component if the Mahalanobis distance from the component is for example less than three standard deviations. The squared distance from the m-th component is calculated as: $D_m^2(\vec{x}^{(t)}) = \vec{\delta}_m^T\vec{\delta}_m/\hat{\sigma}_m^2$. If there are no 'close' components a new component is generated with $\hat{\pi}_{M+1} = \alpha$, $\widehat{\vec{\mu}}_{M+1} = \vec{x}^{(t)}$ and $\hat{\sigma}_{M+1} = \sigma_0$ where σ_0 is some appropriate initial variance. If the maximum number of components is reached we discard the component with smallest $\hat{\pi}_m$.

The presented algorithm presents an on-line clustering algorithm. Usually, the intruding foreground objects will be represented by some additional clusters with small weights $\hat{\pi}_m$. Therefore, we can approximate the background model by the first B largest clusters:

$$p(\vec{x}|\mathcal{X}_T, BG) \sim \sum_{m=1}^{B} \hat{\pi}_m \mathcal{N}(\vec{x}; \widehat{\vec{\mu}}_m, \sigma_m^2 I) \qquad (7)$$

If the components are sorted to have descending weights $\hat{\pi}_m$ we have:

$$B = \arg\min_b \left(\sum_{m=1}^{b} \hat{\pi}_m > (1 - c_f) \right) \qquad (8)$$

where c_f is a measure of the maximum portion of the data that can belong to foreground objects without influencing the background model. For example, if a new object comes into a scene and remains static for some time it will probably generate an additional stabile cluster. Since the old background is occluded the weight π_{B+1} of the new cluster will be constantly increasing. If the object remains static long enough, its weight becomes larger than c_f and it can be considered to be part of the background. If we look at (4) we can conclude that the object should be static for approximately $\log(1 - c_f)/\log(1 - \alpha)$ frames. For example for $c_f = 0.1$ and $\alpha = 0.001$ we get 105 frames.

4. Selecting the number of components

The weight π_m describes how much of the data belongs to the m-th component of the GMM. It can be regarded as the probability that a sample comes from the m-th component and in this way the π_m-s define an underlying multinomial distribution. Let us assume that we have t data samples and each of them belongs to one of the components of the GMM. Let us also assume that the number of samples that belong to the m-th component is $n_m = \sum_{i=1}^{t} o_m^{(i)}$ where

$o_m^{(i)}$-s are defined in the previous section. The assumed multinomial distribution for n_m-s gives likelihood function $\mathcal{L} = \prod_{m=1}^{M} \pi_m^{n_m}$. The mixing weights are constrained to sum up to one. We take this into account by introducing the Lagrange multiplier λ. The Maximum Likelihood (ML) estimate follows from: $\frac{\partial}{\partial \hat{\pi}_m} \left(\log \mathcal{L} + \lambda(\sum_{m=1}^{M} \hat{\pi}_m - 1) \right) = 0$. After getting rid of λ we get:

$$\hat{\pi}_m^{(t)} = \frac{n_m}{t} = \frac{1}{t} \sum_{i=1}^{t} o_m^{(i)}. \tag{9}$$

The estimate from t samples we denoted as $\hat{\pi}_m^{(t)}$ and it can be rewritten in recursive form as a function of the estimate $\hat{\pi}_m^{(t-1)}$ for $t-1$ samples and the ownership $o_m^{(t)}$ of the last sample:

$$\hat{\pi}_m^{(t)} = \hat{\pi}_m^{(t-1)} + 1/t(o_m^{(t)} - \hat{\pi}_m^{(t-1)}). \tag{10}$$

If we now fix the influence of the new samples by fixing $1/t$ to $\alpha = 1/T$ we get the update equation (4). This fixed influence of the new samples means that we rely more on the new samples and the contribution from the old samples is downweighted in an exponentially decaying manner as mentioned before.

Prior knowledge for multinomial distribution can be introduced by using its conjugate prior, the Dirichlet prior $\mathcal{P} = \prod_{m=1}^{M} \pi_m^{c_m}$. The coefficients c_m have a meaningful interpretation. For the multinomial distribution, the c_m presents the prior evidence (in the maximum a posteriori (MAP) sense) for the class m - the number of samples that belong to that class a priori. As in [12] we use negative coefficients $c_m = -c$. Negative prior evidence means that we will accept that the class m exists only if there is enough evidence from the data for the existence of this class. This type of prior is also related to Minimum Message Length criterion that is used for selecting proper models for given data [12]. The MAP solution that includes the mentioned prior follows from $\frac{\partial}{\partial \hat{\pi}_m} \left(\log \mathcal{L} + \log \mathcal{P} + \lambda(\sum_{m=1}^{M} \hat{\pi}_m - 1) \right) = 0$, where $\mathcal{P} = \sum_{m=1}^{M} \pi_m^{-c}$. We get:

$$\hat{\pi}_m^{(t)} = \frac{1}{K}(\sum_{i=1}^{t} o_m^{(i)} - c), \tag{11}$$

where $K = \sum_{m=1}^{M} (\sum_{i=1}^{t} o_m^{(t)} - c) = t - Mc$. We rewrite (11) as:

$$\hat{\pi}_m^{(t)} = \frac{\hat{\Pi}_m - c/t}{1 - Mc/t}, \tag{12}$$

where $\hat{\Pi}_m = \frac{1}{t} \sum_{i=1}^{t} o_m^{(t)}$ is the ML estimate from (9) and the bias from the prior is introduced through c/t. The bias decreases for larger data sets (larger t). However, if a small bias is acceptable we can keep it constant by fixing c/t to $c_T = c/T$ with some large T. This means that the bias will always be the same as if it would have been for a data set with T samples. It is easy to show that the recursive version of (11) with fixed $c/t = c_T$ is given by:

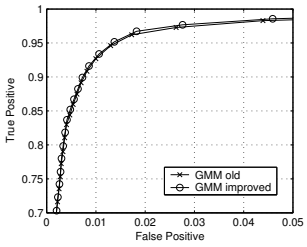


Figure 1. ROC curve for the laboratory sequence

$$\hat{\pi}_m^{(t)} = \hat{\pi}_m^{(t-1)} + 1/t(\frac{o_m^{(t)}}{1 - Mc_T} - \hat{\pi}_m^{(t-1)}) - 1/t\frac{c_T}{1 - Mc_T}. \tag{13}$$

Since we expect usually only a few components M and c_T is small we assume $1 - Mc_T \approx 1$. As mentioned we set $1/t$ to α and get the final modified adaptive update equation

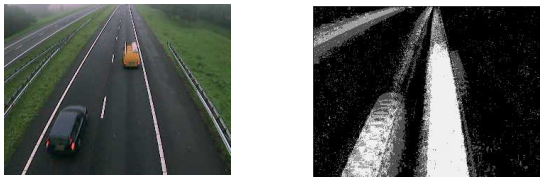
$$\hat{\pi}_m \leftarrow \hat{\pi}_m + \alpha(o_m^{(t)} - \hat{\pi}_m) - \alpha c_T. \tag{14}$$

This equation is used instead of (4). After each update we need to normalize π_m-s so that they add up to one. We start with GMM with one component centered on the first sample and new components are added as mentioned in the previous section. The Dirichlet prior with negative weights will suppress the components that are not supported by the data and we discard the component m when its weight π_m becomes negative. This also ensures that the mixing weights stay non-negative. For a chosen $\alpha = 1/T$ we could require that at least $c = 0.01 * T$ samples support a component and we get $c_T = 0.01$.

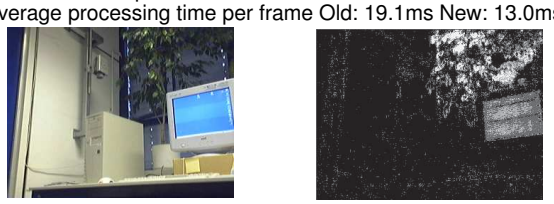
Note that direct recursive version of (11) given by: $\hat{\pi}_m^{(t)} = \hat{\pi}_m^{(t-1)} + (t - Mc)^{-1}(o_m^{(t)}(\vec{x}^{(t)}) - \hat{\pi}_m^{(t-1)})$ is not very useful. We could start with a larger value for t to avoid negative update for small t but then we cancel out the influence of the prior. This motivates the important choice we made to fix the influence of the prior.

5. Experiments

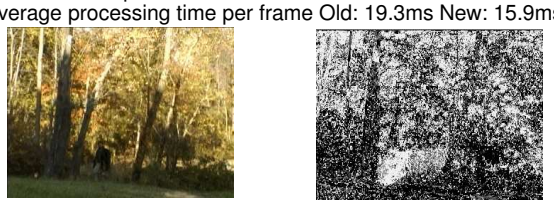
To analyze the performance of the algorithm we used three dynamic scenes. The sequences were manually segmented to generate the ground truth. We compare the improved algorithm with the original algorithm [3] with fixed number of components $M = 4$. For both algorithms and for



'traffic' sequence
selected number of modes M
average processing time per frame Old: 19.1ms New: 13.0ms



'lab' sequence
selected number of modes M
average processing time per frame Old: 19.3ms New: 15.9ms



'trees' sequence
selected number of modes M
average processing time per frame Old: 19.7ms New: 19.3ms

Figure 2. Full adaptation and processing times

different threshold values (c_{thr} from (2)), we measured the true positives - percentage of the pixels that belong to the intruding objects that are correctly assigned to the foreground and the false positives - percentage of the background pixels that are incorrectly classified as the foreground. In figure 1 we present the receiver operating characteristic (ROC) curve for the 'lab' sequence. We observe slight improvement in segmentation results. The same can be noticed for the other two sequences (ROC curves not presented here). Big improvement can be observed in reduced processing time, figure 2. The reported processing time is for 320×240 images and measured on a 2GHz PC. In figure 2 we also illustrate how the new algorithm adapts to the scenes. The gray values in the images on the right side indicate the number of components per pixel. Black stands for one Gaussian per pixel and a pixel is white if maximum of 4 components is used. For example, sequence 'lab' has a monitor with rolling interference bars in the scene. The plant from the scene was swaying because of the wind. We see that the dynamic areas are modelled using more components. Consequently, the processing time also depends on the complexity of the scene. For the highly dynamic 'tree' sequence [4] the processing time is almost the same as for the original algorithm [3]. Intruding objects introduce generation of new components that are removed after some time (see 'traffic' sequence). This also influences the processing speed.

6. Conclusions

We presented an improved GMM background subtraction scheme. The new algorithm can automatically select the needed number of components per pixel and in this way fully adapt to the observed scene. The processing time is reduced but also the segmentation is slightly improved.

Acknowledgments

The work described in this paper was conducted within the EU Integrated Project COGNIRON ("The Cognitive Companion") and was funded by the European Commission Division FP6-IST Future and Emerging Technologies under Contract FP6-002020.

References

[1] C. R. Wren, A. Azarbayejani, T. Darrell, and A. Pentland, "Pfinder: Real-time tracking of the human body," *IEEE Trans. on PAMI*, vol. 19, no. 7, pp. 780–785, 1997.

[2] N. Friedman and S. Russell, "Image segmentation in video sequences: A probabilistic approach," *In Proceedings Thirteenth Conf. on Uncertainty in Artificial Intelligence*, 1997.

[3] C. Stauffer and W. Grimson, "Adaptive background mixture models for real-time tracking," *In Proceedings CVPR*, pp. 246–252, 1999.

[4] A. Elgammal, D. Harwood, and L. S. Davis, "Non-parametric background model for background subtraction," *In Proceedings 6th ECCV*, 2000.

[5] K. Toyama, J. Krumm, B. Brumitt, and B. Meyers, "Wallflower: Principles and practice of background maintenance," *In Proceedings of ICCV*, 1999.

[6] J. Kato, S. Joga, J. Rittscher, and A. Blake, "An HMM-based segmentation method for traffic monitoring movies," *IEEE Trans. on PAMI*, vol. 24, no. 9, pp. 1291–1296, 2002.

[7] M. Harville, "A framework for high-level feedback to adaptive, per-pixel, mixture-of-gaussian background models," *In Proceedings of ECCV*, 2002.

[8] P. J. Withagen, K. Schutte, and F. Groen, "Likelihood-based object tracking using color histograms and EM," *In Proceedings ICIP, USA*, pp. 589–592, 2002.

[9] A. Prati, I. Mikic, M. Trivedi, and R. Cucchiara, "Detecting moving shadows: Formulation, algorithms and evaluation," *IEEE Trans. on PAMI*. vol. 25, no. 7, pp. 918–924, 2003.

[10] E.Hayman and J. Eklundh, "Statistical Background Subtraction for a Mobile Observer", *In Proceedings ICCV*, 2003.

[11] A.Monnet, A.Mittal, N.Paragios and V. Ramesh, "Background Modeling and Subtraction of Dynamic Scenes", *In Proceedings ICCV'03*, pp.1305–1312, 2003.

[12] Z.Zivkovic and F.van der Heijden, "Recursive Unsupervised Learning of Finite Mixture Models", *IEEE Trans. on PAMI, vol.26., no.5, 2004*.