Probabilistic Human-Robot Information Fusion

Tobias Kaupp

A thesis submitted in fulfillment
of the requirements for the degree of
Doctor of Philosophy

Australian Centre for Field Robotics
School of Aerospace, Mechanical and Mechatronic Engineering
The University of Sydney

March 2008

Declaration

I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person nor material which to a substantial extent has been accepted for the award of any other degree or diploma of the University or other institute of higher learning, except where due acknowledgement has been made in the text.

Tobias Kaupp

March 31, 2008

Abstract

Tobias Kaupp
The University of Sydney

Doctor of Philosophy
March 2008

Probabilistic Human-Robot Information Fusion

This thesis is concerned with combining the perceptual abilities of mobile robots and human operators to execute tasks cooperatively. It is generally agreed that a synergy of human and robotic skills offers an opportunity to enhance the capabilities of today's robotic systems, while also increasing their robustness and reliability. Systems which incorporate both human and robotic information sources have the potential to build complex world models, essential for both automated and human decision making.

In this work, humans and robots are regarded as equal team members who interact and communicate on a peer-to-peer basis. Human-robot communication is addressed using probabilistic representations common in robotics. While communication can in general be bidirectional, this work focuses primarily on human-to-robot information flow. More specifically, the approach advocated in this thesis is to let robots fuse their sensor observations with observations obtained from human operators. While robotic perception is well-suited for lower level world descriptions such as geometric properties, humans are able to contribute perceptual information on higher abstraction levels. Human input is translated into the machine representation via Human Sensor Models. A common mathematical framework for humans and robots reinforces the notion of true peer-to-peer interaction.

Human-robot information fusion is demonstrated in two application domains: (1) scalable information gathering, and (2) cooperative decision making. Scalable information gathering is experimentally demonstrated on a system comprised of a ground vehicle, an unmanned air vehicle, and two human operators in a natural environment. Information from humans and robots was fused in a fully decentralised manner to build a shared environment representation on multiple abstraction levels. Results are presented in the form of information exchange patterns, qualitatively demonstrating the benefits of human-robot information fusion.

The second application domain adds decision making to the human-robot task. Rational decisions are made based on the robots' current beliefs which are generated by fusing human and robotic observations. Since humans are considered a valuable resource in this context, operators are only queried for input when the expected benefit of an observation exceeds the cost of obtaining it. The system can be seen as adjusting its autonomy at run-time based on the uncertainty in the robots' beliefs. A navigation task is used to demonstrate the adjustable autonomy system experimentally. Results from two experiments are reported: a quantitative evaluation of human-robot team effectiveness, and a user study to compare the system to classical teleoperation. Results show the superiority of the system with respect to performance, operator workload, and usability.

Acknowledgements

I'd like to start by thanking Alex M. for advising me throughout this work. He is the one who inspired me in the first place to do a PhD by demonstrating the benefits of research and the lifestyle which goes with it. The principles of "work hard, play hard" and "how hard can it be" have guided me nicely through these 4 years. Which brings me to thanking Hugh for enabling me to pursue the research I wanted to do by providing me with a CAS scholarship, and not to forget, support my travels to conferences and summer schools.

I also owe a lot to many other people at ACFR. Above all, to Alex B. who's patiently reviewed most of my written work and taught me a great deal about robotics and how to write good code. Thanks to Ben, Fabio, and Bart who I worked closely with during the ANSERII project for co-authoring my papers. Thanks to Stef, Alex B., Ian, Fabio and Alex Bachmann for reading parts of the thesis.

Studying human-robot interaction requires human subjects and some say they're even harder to handle than robots. In any case, thanks to all the participants of my user studies. Also thanks to Garth who helped me analysing some of the data.

ACFR has always been more than just a work environment. Thanks to everybody who's been participating in maintaining the social side of ACFR: George, the master brewer, who trained me as his apprentice; Alex M., Alex B., and Michael (Moses) for the many BYO dinners in Newtown; the soccer crew (even though I ruptured my ACL); everybody who participated in the Abercrombie St pub crawls and numerous pub lunches. Also thanks to the "backpacker students" (Co, Britta, etc.) for stirring up the social dynamics and bringing new life into the ACFR routine. I'd also like to thank the ACFR girl-friends club for providing us with the latest gossip and for entertaining Eva, so I could spend more time in the lab. Thanks to Tomo who let me and Eva stay at his place during his sabbatical. Thanks to all my non-ACFR contacts (who have diminished over the years) for dragging me out of the lab.

I'd also like to thank my family and friends in Germany. I've managed to visit you quite a few times, and it's always been a very intense and unforgettable experience. Thanks for not forgetting about me and making me welcome every time. There are too many to name everybody personally, so I won't, I'm sure you know who you are.

Last but not least, I'd like to say thankyou to Eva who had to bear my moods and impatience when the research did not work out the way I wanted. Thanks for coming to Australia, it would have been much harder without you.

To my future wife

Contents

List of Figures

List of Tables

Nomenclature

Notation

$(\cdot)_k$	$(\cdot)$ at discrete time k
$(\cdot)_{0:k}$	$(\cdot)$ up to time k including k
$(\cdot)_{a:b}$	$(\cdot)$ from time a up to time b
$(\cdot)^n$	number n of $(\cdot)$
$(\hat{\cdot})$	estimate of $(\cdot)$
$parents(\cdot)$	parents of random variables in $(\cdot)$

General Symbols

k	an index into discrete time
K	total length of time sequence
A	a random variable
x	a scalar state (hidden random variable)
$\mathbf{x}$	a state vector (hidden random variable)
z	a scalar observation (observed random variable)
$\mathbf{z}$	an observation vector (observed random variable)
$\overline{\mathbf{z}}$	fixed observation (value)
$\mathbf{d}$	a vector of actions
d^*	the best action
$\mathbf{i}$	a potential information source
C	cost for obtaining information
$\mathcal{N}$	Normal (Gaussian) distribution
Θ	all parameters
μ,ν	mean vector of a Gaussian
Σ,Ψ	covariance matrix of a Gaussian
Λ	regression matrix of a Gaussian
μ	mean of a 1-dim. Gaussian

σ	standard deviation
ω	regression coefficient of a 1-dim. Gaussian
p	significance level
$F[a, b]$	F-ratio with degrees of freedom a and b

Typefaces

MYCOMPONENT	Software component
`Myinterface`	Component interface
MyNodeName	BN node name
MyDiscreteState	Discrete BN node's state

Abbreviations

AA	Adjustable Autonomy
AI	Artificial Intelligence
ANOVA	Analysis of Variance
BN	Bayesian Network
CPD	Conditional Probability Distribution
CPT	Conditional Probability Table
DAG	Directed Acyclic Graph
DDF	Decentralised Data Fusion
DBN	Dynamic Bayesian Network
EM	Expectation Maximisation
EU	Expected Utility
EUO	Expected Utility of Optimal Action
GMM	Gaussian Mixture Model
GPS	Global Positioning System
GUI	Graphical User Interface
HCI	Human Computer Interaction
HMM	Hidden Markov Model
HRI	Human Robot Interaction
HSM	Human Sensor Model
ID	Influence Diagram
IMU	Inertial Measurement Unit
KF	Kalman filter
kNN	k-Nearest Neighbour
MAP	Maximum A Posteriori
MFA	Mixture of Factor Analysers
ML	Maximum Likelihood
PCA	Principal Component Analysis

SA	Situation Awareness
SAR	Search and Rescue
SLAM	Simultaneous Localisation and Mapping
UAV	Unmanned Aerial Vehicle
UML	Unified Modelling Language
VOI	Value of Information

Chapter 1

Introduction

1.1 Problem Description and Motivation

This thesis is concerned with combining the perceptual abilities of mobile robots and human operators to execute tasks cooperatively. It is generally agreed that a synergy of human and robotic skills offers an opportunity to enhance the capabilities of today's robotic systems, while also increasing their robustness and reliability [23][52][155]. Despite the fact that perception of the environment is a critical element in performing any task, *cooperative* human-robot perception has not been studied extensively. Systems which incorporate both human and robotic information sources have the potential to build complex world models, essential for both automated and human decision making.

Although robotic systems are often perceived as a replacement for human labour, there are in fact many opportunities for human-robot cooperation. Mobile robots rarely operate in isolation from humans for several reasons: technical, ethical, and by design. Technical reasons refer to the fact that full robot autonomy is rarely achievable. Today's mobile robots typically operate autonomously for limited periods of time in relatively structured and known environments. The situation is unlikely to change in the near future. In fact, full autonomy may be considered a misnomer because at the very least, operators are required to provide high-level, abstract goals [56]. Ethical reasons for human involvement may arise in situations requiring decisions which only humans are qualified to make. Example application areas include military, search-and-rescue, and health. Finally, people are often an integral part of the system by design: physical interactions between humans and robots occur in assistive robotics and edutainment for example. In addition to the human involvement in robotic systems as described above, an opportunity arises to make use of human resources to improve system performance.

Human-robot cooperation is likely to fulfil this objective because humans and robots cover a wide range of skills in perception, cognition, and manipulation. This thesis focusses exclusively on the *perceptual* abilities of humans and robots. Human and robotic perception is often complementary in terms of modality, uncertainty, and types of failures. Humans have rich perceptual abilities, especially at higher abstraction levels, *e.g.* human innate pattern recognition skills [59]. Yet, people's performance is known to suffer from great variability between individuals and over time. On the other hand, robotic perception is highly consistent and accurate in measuring lower level descriptions such as geometric properties. At the same time, robotic perception has limited ability to generalise from preprogrammed concepts, *e.g.* in visual object recognition.

Human-robot cooperation typically involves communication. Important research questions for human-robot communication are:

1. What type of information should be communicated?
2. When should communication occur?
3. Who should communicate with whom?
4. What medium should be used for communication?

The space of options available in answering these questions is large and depends on a number of factors. Six factors are identified here: proximity, authority relationship, number of humans and robots, human factors, communication bandwidth, and task priority. The influence of each factor on answering the questions above are discussed next.

Proximity Interactions between humans and robots can either be remote or proximate [68]. If humans and robots are collocated as they are in a service robot application, the communication requirements differ greatly from a remote teleoperation scenario. Likewise, the choice of the communication medium depends on proximity, *i.e.* for close interactions, speech, gestures, haptic interfaces, or social cues may be appropriate while a graphical or textual user interface may be better suited for remote interactions.

Authority Relationship If humans act on a higher level of authority than robots, operators initiate communication and send instructions and commands to robots. If the relationship is more peer-like, communication could involve dialog which either side may initiate [53]. Information exchange related to dialog includes questions, answers, queries, and clarifications.

Number of Humans and Robots If many humans and robots are involved, the question of who communicates with whom becomes important. The objective is to find suitable communication topologies capable of maintaining the scalability of the system [171]. In that context, it is also important to communicate efficiently, *i.e.* only communicate when necessary and maximise the information content per message.

Human Factors The form of communication is greatly influenced by human factors such as expertise [53]. On one end of the spectrum is the robot designer who communicates with robots using a programming language. Ordinary users, on the other hand, may prefer the use of natural language and gestures. Other human factors such as operator workload, stress, and fatigue influence how frequently information should be exchanged.

Communication Bandwidth Limited bandwidth can cause communication problems such as message delays and frequent drop-outs. If bandwidth is a bottleneck, information exchange needs to be efficient as mentioned above. Limited bandwidth may also impose constraints on the possible network topologies [115]. Likewise for the communication medium: broadcasting a video stream to an operator may not be feasible.

Task Priority The form of communication also depends on the nature of the task. If safety is a priority, frequent communication and detailed messages can be justified. On the other hand, if timeliness is more important, information exchange should be limited to the most important messages [95].

All six factors are taken into consideration when designing communication patterns for human-robot cooperation. In this thesis, the focus is on questions (1)–(3) in order to fulfil the objective of combining the perceptual strengths of humans and robots. Question (4) falls outside the scope of this work. Graphical User Interfaces (GUIs) were used throughout the experiments presented in this thesis.

1.2 Approach

This work adopts a *peer-to-peer* interaction style to facilitate bidirectional communication between humans and robots. The Human-Robot Interaction (HRI) community generally agrees that humans and robots need to interact as peers to leverage each other's strengths effectively [52][117][23][77][31][18]. In systems where robots act as peers, they are treated as partners rather than tools. In *supervisory control*, in contrast, operators act on a higher level of authority compared to robots and may override their decisions [159]. Furthermore, timely responses from operators are expected in supervisory control. When treating robots as peers, humans contribute to the task without becoming a bottleneck, *i.e.* operators are not required to remain in the loop.

The approach presented in this thesis is to make use of *probabilistic* robotics representations for bidirectional human-robot communication. Thrun concluded in 2000 that *"we see a tremendous opportunity to apply probabilistic algorithms to a range of important robotic problems, including* [...] *human-robot interaction"* [173]. To the best of the author's knowledge, this thesis provides the first attempt to systematically investigate human-robot interactions from a probabilistic point of view. This approach is classified as *robot-centred* because it takes existing probabilistic robotics algorithms as a basis for communication. Figure 1.1 shows where the focus of this thesis is placed within the multidisciplinary field of HRI.

The information stored in a probabilistic robotics representation can be used for bidirectional human-robot communication. The majority of work to date has focussed on robot-to-human information flow whereby the information collected by robots is conveyed to human operators, *e.g.* to enhance Situation Awareness (SA) [156][153][195][43]. On the contrary, the focus of this thesis is on human-to-robot information flow, addressing the problem of how human operators can effectively contribute to a task performed by a human-robot system.

In this thesis, methods are presented which are able to probabilistically fuse information collected by human operators and robotic sensors. In robotics, integrating the strength of

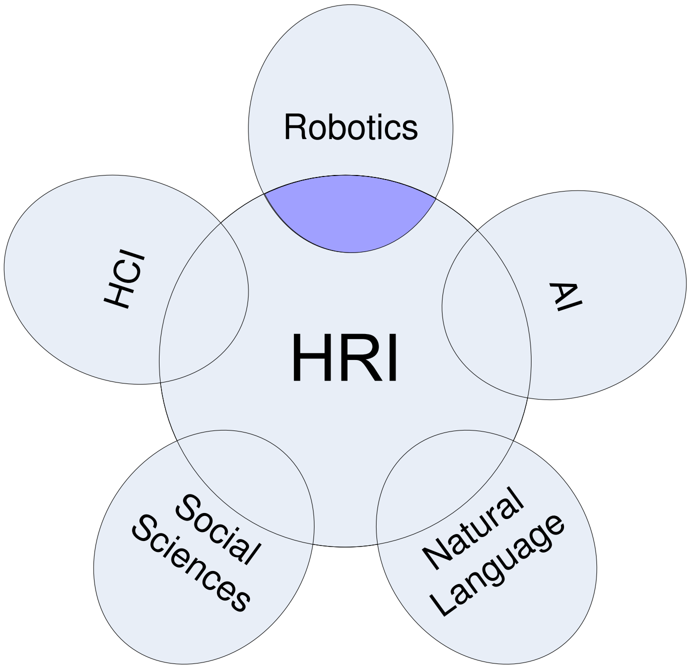


Figure 1.1: HRI is a multidisciplinary field with contributions from robotics, Human-Computer Interaction (HCI), social sciences (psychology, cognitive science, anthropology, human factors), natural language understanding, and Artificial Intelligence (AI). This thesis takes the viewpoint of robotics to facilitate human-robot interactions as indicated by the darker area in the figure.

different sensing modalities has traditionally been approached by performing *information fusion*. This thesis applies information fusion techniques to HRI by treating human operators as *information sources*. A common mathematical framework for humans and robots reinforces the notion of true peer-to-peer interaction.

1.3 Relationship Taxonomies

This section introduces relationship taxonomies applicable to human-robot interactions. The taxonomies are used throughout this document to set the contributions into the context of current HRI research.

The relationship between humans and robots can be classified from different points of view. First, an overview of the human-robot "system" is given. Second, novel taxonomies based

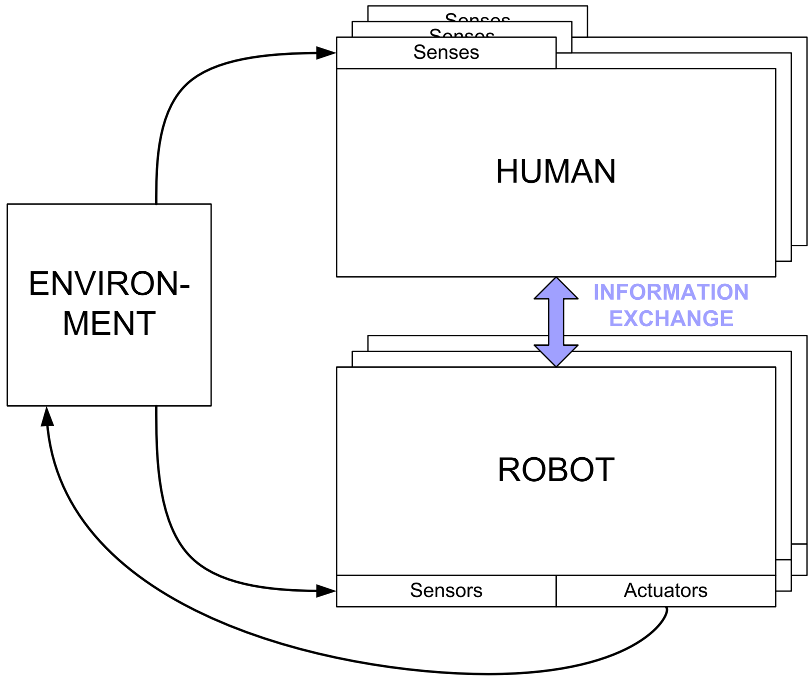


Figure 1.2: Human-robot system: multiple humans and robots interact with each other and the environment. The information exchange between humans and robots is the focus of this thesis.

on human-robot communication are proposed. Finally, taxonomies from the literature are reviewed.

1.3.1 The Human-Robot System

Humans and robots interact with each other and the environment in which they operate. Figure 1.2 visualises the system from a global point of view. The environment is observed by both humans and robots. Robots are able to alter the state of the environment through their actions using their actuators. It is assumed that humans are not able to change the environment directly through their actions. The focus of this work is on the information exchange between humans and robots as indicated by the bidirectional arrow.

Figure 1.3 visualises the internal structure of humans, robots and the environment. All parts concerned with *perception* are colour-coded blue, all parts concerned with *decision making* are colour-coded orange.

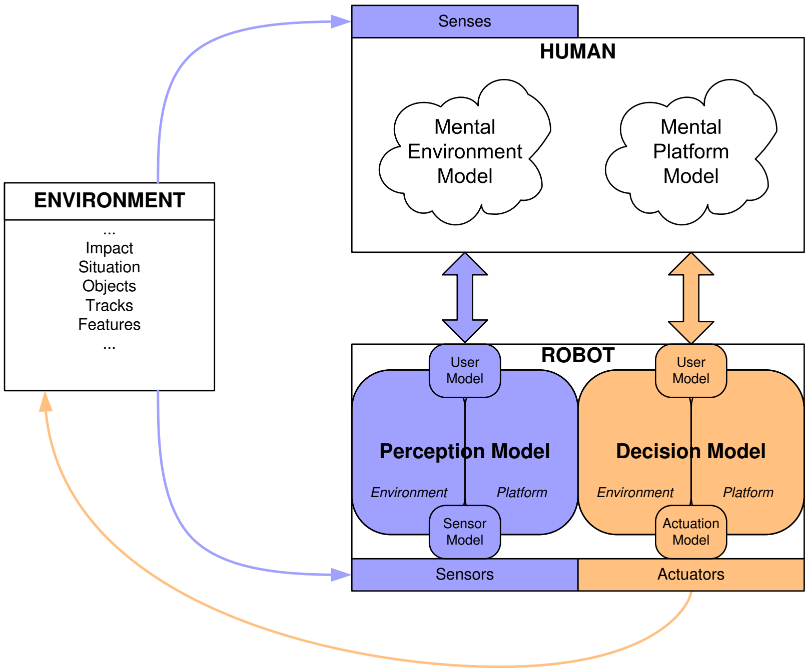


Figure 1.3: Internal structure of the environment, human and robots (compare to Figure 1.2). Clouds represent mental models whereas rounded rectangles represent computational models. Parts concerned with perception are blue, parts concerned with decision making are orange.

Environment The environment contains complex phenomena which can be observed at different abstraction levels. Here, a terminology common in the information fusion community is adopted [111]. For example, the environment can contain a number of mobile *objects* which interact to create a *situation* of certain *impact* to an interested party.

Humans Humans internally maintain *mental* models resulting from observations of the environment and the robotic platforms. Cognitive scientists define mental models as an internal scale-model representation of an external reality [60]. Mental models play an important role in improving the usability of computers [130], and to establish a common ground between humans and robots [97]. Mental models (visualised as clouds in the figure) are not discussed any further in this thesis.

Communication taxonomy	**Possible values**
Direction	human → robot, robot → human
Initiation	human, robot
Data type	perception, decision
Data origin	environment, platform

Table 1.1: Four human-robot communication taxonomies derived from Figure 1.3.

Robots Robots maintain two types of *computational* models[1] visualised as rounded rectangles: a *perception model* and a *decision model*. A distinction is made between environment and platform models: the former is concerned with representing the physical world, the latter with representing the robotic platforms within the world, *e.g.* their poses.

The *perception model* has two incoming data streams: information from physical sensors and human operators. Both streams are transformed using *sensor* and *user* models to be suitable for internal fusion. The perception model also has an outgoing stream to communicate perception information to the human operator.

The *decision model* uses the information stored in the perception model to make decisions. An *actuation model* represents the effects of the decisions on the environment and the platform itself. Humans can also interact with the decision model: either by observing the decisions that are made (outgoing stream), or by actively participating in decision making (incoming stream). As for the perception model, user input can be transformed by a user model.

1.3.2 Communication Taxonomies

This section proposes relationship taxonomies based on human-robot communication. Four communication taxonomies are identified, each derived from the internal structure of humans and robots as shown in Figure 1.3: *direction* of data flow, *initiation* of data flow, *data type*, and *data origin*. Table 1.1 summarises the four taxonomies which are explained next.

Direction specifies the main data flow and can either be human-to-robot or robot-to-human. Communication is either initiated by a human or a robot. Data types are related to either the perception or the decision model. Data origin refers to either environment or platform.

[1] Philosophical definitions of the term *computational model* include "*a mathematical modelling and idealisation of some hypothetical physical device, from a specific point of view of the world*" [16] and "*an implemented computer program used to generate data that will be compared to some real-world behaviour*" [166].

Direction	**Initiation**	
	human	robot
human → robot	human-push	robot-pull
robot → human	human-pull	robot-push

Table 1.2: Human-robot communication patterns defined by the direction and initiation of information flow.

Environment data describes properties of the environment such as object locations, object types, situations, and temperature. Platform data is concerned with the physical platforms. Examples are location in the world, drive commands, hardware states, and software component health status.

It is convenient to arrange the one-dimensional measures of direction and initiation into a two-dimensional matrix as shown in Table 1.2. Data flow direction and initiation together define one of four *communication patterns*: human-push, robot-pull, human-pull, and robot-push. The same principle is applied to data type and data origin as shown in Table 1.3. Four *messages* are assumed for communication: environment-perception, environment-decision, platform-perception, and platform-decision. All communication taxonomies and definitions are summarised in Figure 1.4.

Data origin	**Data type**	
	perception	decision
environment	environment-perception	environment-decision
platform	platform-perception	platform-decision

Table 1.3: Human-robot communication messages defined by the data type and data origin.

1.3.3 Other Taxonomies

The previous section presented four communication taxonomies to describe the relationship between humans and robots. Other relationship taxonomies commonly used in HRI include numeric, spatial, and authority [26][152]. Table 1.4 summarises the numeric relationships.

Table 1.5 shows the possible spatial relationships between humans and robots. Operators can work remotely in which case they might have a god's eye view of the robots' workspace (*e.g.* a map showing the environment and the robot) or look through the robot's eyes (*e.g.* camera image or laser scan). Operators can also work side-by-side to robots in which case they have access to the environment directly through their own senses.

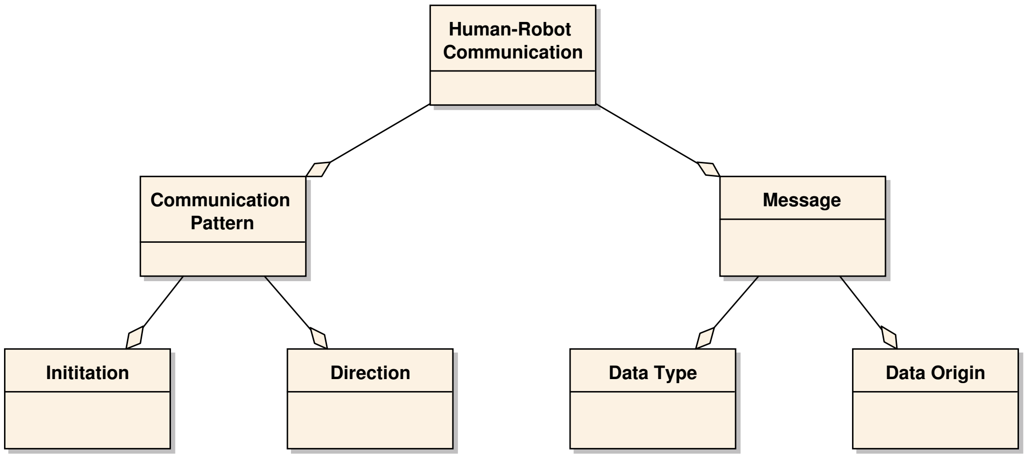


Figure 1.4: Aspects involved in human-robot communication shown as a UML class diagram. Diamonds indicate a "consists-of" relationship. A communication pattern consists of the initiation and the direction of information flow. A message consists of a data type and a data origin. The bottom row shows the four communication taxonomies proposed as part of this work.

Humans	**Robots**
One	One
One	Many
Many	One
Many	Many

Table 1.4: Numeric relationships of humans and robots.

Table 1.6 defines relationships in terms of the authority level: human roles include *supervisor*, *teleoperator*, and *peer*. The supervisor role can be characterised as monitoring and control of the overall situation. A supervisor commands what to do by specifying high-level mission goals or by switching operating modes and thus requires high-level perception of the situation.

A teleoperator interacts with the robot on a lower level by directly telling it how to achieve a task by classical teleoperation. In this role, the operator needs to access low-level information such as distance to obstacles.

Spatial relationship	**Human's point of view**
Remote	God's eye, sensor view
Beside	Bystander

Table 1.5: Spatial relationships of humans and robots.

Authority relationship	Operator tasks	Required context
Supervisor	Commands "what"	High-level perception
Teleoperator	Commands "how"	Low-level perception
Peer	Provides/requests assistance	High-level perception

Table 1.6: Authority relationships of humans and robots.

In the peer-to-peer interaction scheme, operators and robots act on the same level of authority. As for the supervisory mode, humans provide input on a higher level and thus require a higher level understanding of the situation. The difference is that robots are not "overridden" in this relationship but act as equal partners forming a *team* with human peers. Each team member contributes what it can do best and operators can also request assistance from robots.

1.4 Related Work

This section gives an overview of seven systems in which mobile robots and human operators communicate with each other. Only task-oriented human-robot systems are chosen, *i.e.* human-robot communication for social interactions are not considered. Work related to the communication medium, *i.e.* human-robot interfaces, is not reviewed either.

The relationship taxonomies presented in Section 1.3 are used to classify these systems as shown in Table 1.7. The table is to be read as follows: a filled circle indicates an important human-robot relationship for the contribution of that work. An open circle indicates that the relationship is mentioned or implemented but it is minor and does not contribute to the key arguments. A missing circle means the relationship is not covered in that work.

Fong's *Collaborative Control* is a form of teleoperation where humans are treated as a resource to robots [53][54]. Bidirectional communication in the form of human-robot dialog is used to exchange information of different types such as commands, queries and responses. Recently, this work has been extended to human-robot cooperation for space exploration missions [52][51]. As shown in Table 1.7, the focus of that work is on peer-to-peer interaction ("robot as a partner"). Robot-pull is the predominant communication pattern employed.

A system performing *Collaborative Teleoperation* is proposed by Goldberg *et al.* [64][65][163]. In this work, the term *collaboration* refers to multiple operators coming to a consensus over a shared resource which can be a remotely located robot [64], sensor [163], or *tele-actor* [65]. Consensus is achieved by combining input from multiple operators into a single output stream using vector averaging [64], dynamic voting [65], or task-specific approximation algorithms [163].

As a substitute for non-existing multi-robot systems, Jones *et al.* performed an ethnographic study of interactions between members of a SWAT (Police Special Weapons and Tactics) team [87]. The SWAT *commander* corresponds to the robot operator whereas the SWAT *team* corresponds to a multi-robot system. The findings were incorporated into their human-robot dialog system [88]. Only the operator can start off a dialog to which all robots in the system reply with their available information. Robots execute actions upon the user's selection of a particular robot and its task. As shown in Table 1.7, the focus of this work is on one-to-many interactions. Human-pull is the predominant communication pattern employed.

Bruemmer's work has focussed on building a *collaborative workspace* for humans and robots, *i.e.* a shared understanding of each others' tasks and the environment [25][23]. Different discrete autonomy levels have been developed which the operator can adjust online. The human role and authority varies with the autonomy level. The system has successfully been applied to a number of applications: map building using a team of air/ground vehicles and human operators [25], indoor search and exploration [23], and countermine operations [24]. As shown in Table 1.7, the focus of this work is on remote operations with the goal of creating a common representation for humans and robots acting as peers.

Another body of work which lets operators choose between a set of discrete autonomy levels is presented in [67]. Each autonomy mode has a certain *Neglect Tolerance*, *i.e.* a particular robot can be neglected for a certain time period without becoming less effective. These ideas were quantitatively evaluated by user studies [31]. As shown in Table 1.7, the focus of this work is on achieving many-to-many interactions. Human-push is the communication pattern employed for entering waypoints and high-level behaviours.

Sellner *et al.* designed a human-robot system for large-scale *assembly* of structures in space [155][154]. Robots with complementary perception and manipulation capabilities cooperate with a human operator who is able to take over control. As shown in Table 1.7, the focus of this work is on incorporating multiple robots into the system. As a consequence, the robot-pull communication pattern is considered adequate to reduce operator workload for one-to-many interactions.

Human interaction in the context of reactive multiagent control is addressed by Ali *et al.* [3][2]. Human operators can change the multi-robot system's global behaviour by either adding their own behaviours or change the behavioural parameters. The focus of this work is on supervisory control of multiple robots using a human-push communication pattern as shown in Table 1.7.

The seven systems presented above address one or more of the challenges of human-robot communication as identified by Sheridan ten years ago [160]. However, none of them takes the principled approach which is advocated in this thesis: use the mathematical framework of existing probabilistic robotics representations to generate human-robot communication schemes. A further difference to this work is that none of the systems directly exploits human perceptual abilities.

1.5 Contributions

The most general contribution of this thesis is a framework for human-robot communication based on probabilistic methods. More specifically, the contributions of this thesis include:

- The introduction of communication taxonomies to classify the relationship between humans and robots with respect to information exchange.
- The derivation of human-robot communication schemes based on probabilistic robotics representations, in particular for human-robot information fusion.
- The presentation and experimental validation of methods for probabilistically modelling humans as information sources.
- The integration of multiple human operators into a fully decentralised multi-robot system without jeopardising the scalability of the system.
- The formulation of a shared environment representation on multiple abstraction levels suitable for scalable human-robot information gathering.

		Collab. Control [53][54] [52][51]	**Collab. Teleop.** [64][65] [163]	**SWAT** [88][87]	**Collab. Worksp.** [25][23] [24]	**Neglect Tolerance** [67][31]	**Assembly** [155][154]	**Behav. Control** [3][2]	**Ch. 3**	**Ch. 4**	**Ch. 5**
Communication											
message	env.-perc.	○		○	○	○	○	○	●	●	●
	env.-dec.		○				●		○		
	plat.-perc.	○		○	○	○	○	○	●	○	●
	plat.-dec.	●	○	●	●	●	○	●	○	○	●
comms. pattern	human-push		○		○	●	○	○	○	●	○
	human-pull	○		●					○		
	robot-push	○		○	○	○	○		○	○	○
	robot-pull	●	○		○		●		○		●
Other											
numeric	one-one	○			○	○		○			○
(human-robot)	one-many	○		●		○	●	●			
	many-one		●		○						
	many-many					●	○			●	○
spatial	remote	○	●	○	●	○	○	○		○	○
	beside	○								○	
authority	supervisor			○	○	○		●			
	teleoperator		●		○	○	○		○	○	○
	peer	●			●		○		●	●	○

Table 1.7: Human-robot relationships in related work (centre column) and covered in this thesis (right column): filled circle means “important”, open circle means “minor”, no circle means “not applicable”. Abbreviations are: env. = environment, plat. = platform, perc. = perception, dec. = decision.

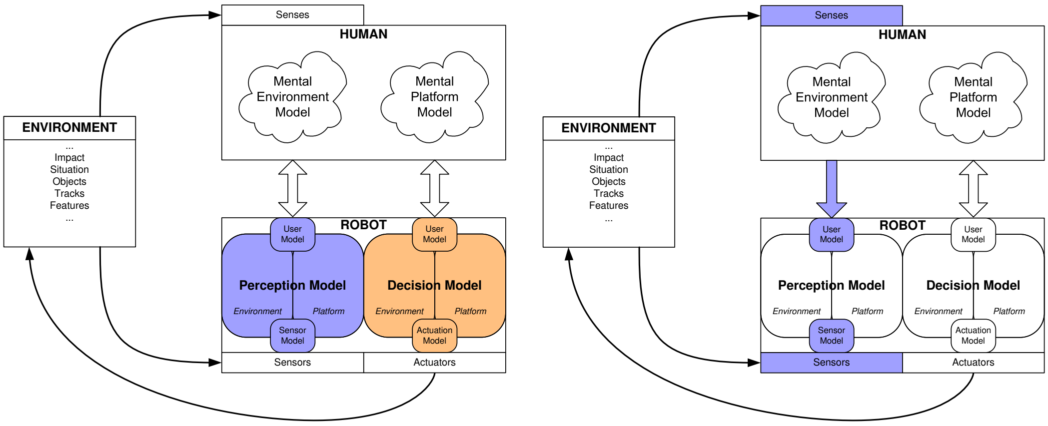
- The experimental demonstration of fully decentralised human-robot information fusion in an outdoor environment.
- The formulation of a decision-theoretic framework for cooperative human-robot decision making with humans acting as a resource to robots.
- The presentation of an adjustable autonomy system for which adjustments are triggered based on the uncertainty in the robot's belief.
- The proposal of a methodology for measuring the effectiveness of a human-robot team. This measure can be used to find an appropriate autonomy level for an adjustable autonomy system.
- The implementation of cooperative human-robot decision making using a mobile robot navigation task.
- The experimental evaluation of cooperative human-robot navigation in simulation and by conducting an extensive user study.

1.6 Thesis Structure and Publications

This section provides an overview of the thesis. Chapter 2 presents background whereas Chapters 3–5 are novel contributions. Each of the contribution chapters are organised identically: theory, experiments, related work. As a consequence, there is no need for chapters dedicated to related work and experimental results. Figure 1.5 uses the system view from Figure 1.3 to clarify the structure of the thesis. Each subfigure is repeated at the beginning of each chapter to remind the reader of how the chapter's content fits into the overall context. The figure is utilised to describe each chapter individually as follows.

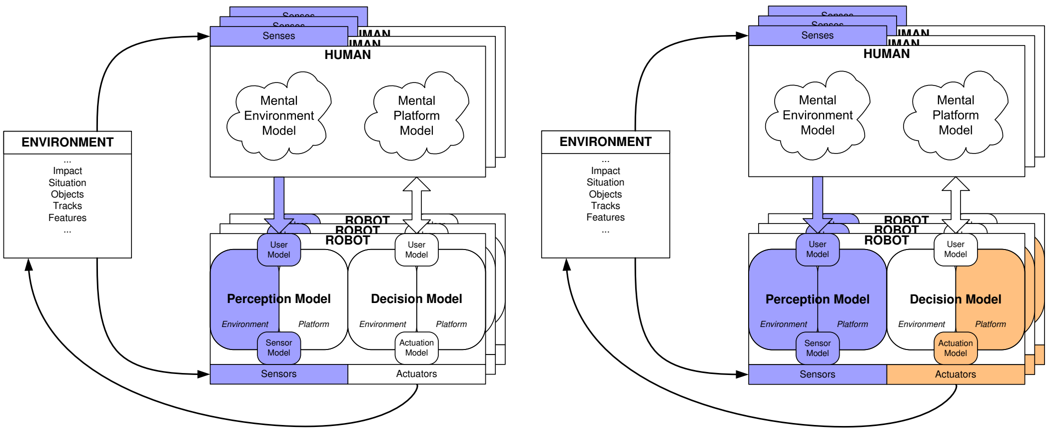
Chapter 2 presents the background on probabilistic representations applied to the robotics problems of perception, decision making, and planning (Figure 1.5(a)). The focus is on the types of representations which can be used for human-robot communication and information fusion.

Chapter 3 serves as a foundation for subsequent chapters. It first discusses how probabilistic data types can be utilised for bidirectional human-robot communication. Subsequently, the



(a) Chapter 2: Probabilistic Representations

(b) Chapter 3: Humans as Information Sources



(c) Chapter 4: Scalable Human-Robot Information Gathering

(d) Chapter 5: Cooperative Human-Robot Decision Making

Figure 1.5: Graphical thesis overview: each chapter focusses on the highlighted parts of the overall system's picture (see Figure 1.3). Chapter 2 presents technical background on probabilistic representations for perception, decision making, and planning. Chapter 3 introduces humans as information sources. Chapter 4 presents scalable interactions as part of a human-robot information gathering mission. Chapter 5 discusses cooperative human-robot decision making.

focus is on human-to-robot information flow which leads to the notion of regarding human operators as information sources. This is visualised by the blue arrow in Figure 1.5(b). Differences between human and robotic sensors are listed and the need for modelling human operators is discussed. Human Sensor Models (HSMs) are introduced as a user modelling technique for this work. The potential limitations of the overall approach are also discussed. Simulation results show how human and robotic information can be fused using multi-level representations. An experimentally acquired HSM is also presented. Publications relevant to this chapter include

- T. Kaupp, A. Makarenko, F. Ramos, and H. Durrant-Whyte, “Human sensor model for range observations,” in *IJCAI Workshop Reasoning with Uncertainty in Robotics (RUR’05)*, Edinburgh, Scotland, 2005.
- T. Kaupp, A. Makarenko, F. Ramos, B. Upcroft, S. Williams, and H. Durrant-Whyte, “Adaptive human sensor model in sensor networks,” in *8th International Conference on Information Fusion (Fusion’05)*, vol. 1, Philadelphia, USA, 2005.
- T. Kaupp, A. Makarenko, S. Kumar, B. Upcroft, and S. Williams, “Operators as information sources in sensor networks,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS’05)*, Edmonton, Canada, 2005, pp. 936–941.

Chapter 4 presents results from an information gathering mission conducted by a human-robot team. The team consists of multiple humans and robots as visualised in Figure 1.5(c). The system uses fully decentralised fusion algorithms and is constrained by the requirement to maintain scalability. This requirement restricts the available options for human-robot communication and the roles human operators can play. A multi-level shared environment representation is used to fuse information from humans and robots using the human-push communication pattern. Experimental results obtained by deploying a human-robot team in an outdoor environment demonstrate the benefits of human-robot information fusion qualitatively. Publications relevant to this chapter include

- T. Kaupp, B. Douillard, B. Upcroft, and A. Makarenko, “Hierarchical environment model for fusing information from human operators and robots,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS’06)*, Beijing, China, 2006, pp. 5837–5842.
- T. Kaupp, A. Brooks, B. Upcroft, and A. Makarenko, “Building a software architecture for a human-robot team using the Orca framework,” in *IEEE International Conference on Robotics and Automation (ICRA’07)*, Rome, Italy, 2007, pp. 3736–3741.

- T. Kaupp, B. Douillard, F. Ramos, A. Makarenko, and B. Upcroft, "Shared environment representation for a human-robot team performing information fusion," *Journal of Field Robotics: Special Issue on Teamwork*, vol. 24, no. 11–12, pp. 911–942, 2007.

Chapter 5 extends information gathering by adding decision making to the task. Only platform decisions are considered as visualised in Figure 1.5(d) which excludes manipulation tasks. Robots make platform decisions based on information collected by their sensors and from human operators. This mechanism is referred to as *cooperative* human-robot decision making. Operators are regarded as a sparse resource in this context which motivates the use of the robot-pull communication pattern. Robots query operators only if the expected benefit from their information exceeds the cost of obtaining it which is computed using value-of-information theory. This mechanism can be interpreted as adjusting the robot's autonomy based on the uncertainty of its beliefs. A navigation task is used to demonstrate an implementation of the adjustable autonomy system. Experiments with the navigation system include a quantitative evaluation of human-robot team effectiveness, and an extensive user study. Publications relevant to this chapter include

- T. Kaupp and A. Makarenko, "Decision-theoretic human-robot communication," in *3rd ACM/IEEE Conference on Human-Robot Interaction (HRI'08)*, Amsterdam, The Netherlands, 2008, pp. 89–96.
- T. Kaupp and A. Makarenko, "Measuring human-robot team effectiveness to determine an appropriate autonomy level," in *IEEE International Conference on Robotics and Automation (ICRA'08)*, Pasadena, CA, USA, 2008, pp. 2146–2151.

Chapter 6 summarises the content of the thesis and presents opportunities for ongoing research based on this work. Furthermore, it is discussed how this work can be utilised for a number of real-world robotics applications.

A thesis overview complementary to Figure 1.5 is presented in Table 1.7.

Chapter 2

Probabilistic Representations

This work uses probabilistic robotics representations as a foundation to design human-robot communication schemes as visualised in Figure 2.1. This chapter motivates that approach by first discussing probabilistic algorithms which have significantly advanced the field of mobile robotics. Second, the technical background on probabilistic representations is presented. The main purpose of this chapter is the introduction of notation and terms which are required for the understanding of subsequently presented material.

The chapter is organised as follows: Section 2.1 briefly discusses probabilistic robotics. Section 2.2 introduces a class of graphical model representations called *Bayesian Networks* (BNs). Their extension to the time domain is called *Dynamic Bayesian Networks* (DBNs) which are presented in Section 2.3. Section 2.4 discusses *Influence Diagrams* (IDs) before Section 2.5 summarises the content of this chapter.

2.1 Probabilistic Robotics

In mobile robotics, probabilistic representations have gained wide acceptance due to their suitability to address the main research problems of perception, decision making, and planning [176]. Probability theory allows to explicitly accommodate the uncertainties of the real world. Uncertainties stem from both noisy measurements and inaccurate world models. Probability theory provides a unifying mathematical framework for reasoning and decision making under uncertainty. Probabilistic algorithms for perception, control, and planning problems are briefly discussed below.

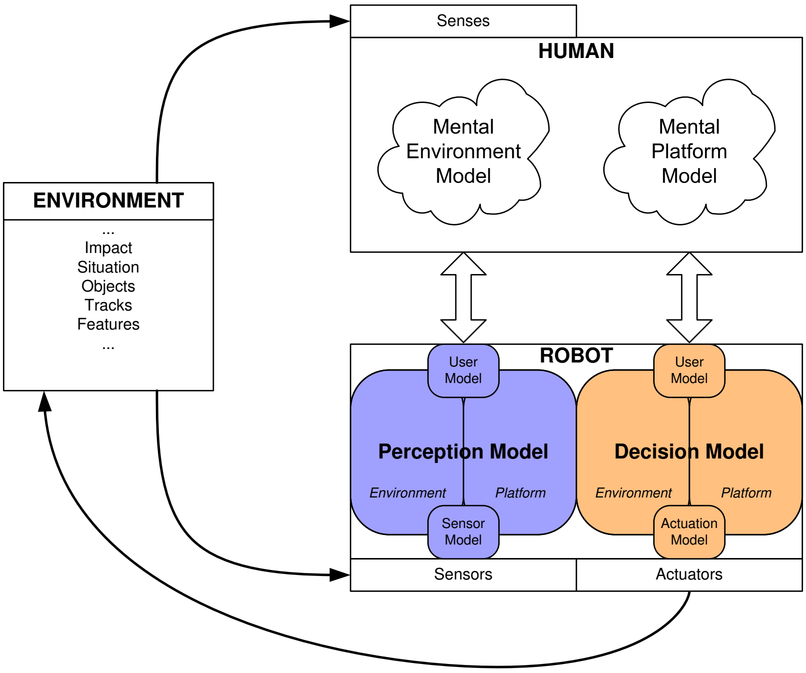


Figure 2.1: This chapter provides technical background: probabilistic robotics representations for perception, decision making, and planning. Human-robot communication schemes are derived from these representations.

2.1.1 Perception, Control and Planning

An important class of robotic perception algorithms is mapping [175]. The main reason for map building is to assist the task of localisation: the estimation of the robot's pose in the world. Localisation and mapping are coupled problems and can be solved simultaneously using SLAM algorithms [47][7]. The objective of mapping can also be the generation of maps for human consumption, *e.g.* in information gathering missions [113]. Platform control is also part of information gathering where the maximisation of information determines the action selection [71].

Robotic control architectures typically consist of several layers and can also be implemented using probabilistic representations [107][40][174][58][38]. Two alternatives exist to make decisions when using a probabilistic model: (1) use perception states only and choose actions by either threshholding or sampling from beliefs [107][40], or (2) explicitly represent the decision maker's choices and preferences [174][58][38]. Preferences are encoded as utility functions which also allows for planning. The planning problem can be described by a

Partially Observable Markov Decision Process (POMDP) which is an active research area in robotics [20].

2.1.2 Bayesian Networks and Extensions

All of the perception and decision-making problems mentioned above can be represented by (Dynamic) Bayesian Networks, or if decisions and utilities are represented explicitly, Influence Diagrams. These representations have properties which are exploited in subsequent chapters for human-robot communication. In a single representation, they are capable of representing

- variables related to perception and decision making
- both correlations and the lack of correlations between random variables
- observed and unobserved random variables
- discrete and continuous random variables
- static and dynamic processes
- multiple abstraction levels
- different time scales

Besides this flexibility for modelling purposes, there are many prominent algorithms for efficient inference and learning for these representations. The properties and usage of BNs, DBNs, and IDs are presented next.

2.2 Bayesian Networks

A probabilistic representation should encode the relationships between random variables qualitatively (model structure) and quantitatively (model parameters), and allow efficient inference and learning. A class of graphical models fulfilling these requirements are *Bayesian Networks* (BNs) [148]. Like other graphical model representations, BNs encode a joint probability distribution of a set of random variables in a compact form by exploiting conditional independence assumptions.

Each random variable is called a *chance node* and denoted by a capital letter, *e.g.* X. The realisation of a random variable is denoted by small letters, *e.g.* x. If the random variable is a vector as opposed to a scalar, it is printed in bold, *e.g.* $\mathbf{x}$. Chance nodes can be either discrete or continuous. All equations in this section assume discrete nodes which can be rewritten for continuous nodes by replacing sums with integrals.

2.2.1 Representation

A BN is a directed acyclic graph (DAG) consisting of chance nodes and edges which connect the nodes[1]. If there is an edge from node X^1 to X^2, X^1 is called a *parent* of X^2 and X^2 is called a *child* of X^1. Each node X^i has a conditional probability distribution (CPD) $p(\mathbf{x}^i|parents(\mathbf{x}^i))$ encoding the effect of the parents on that node. An edge from parent to child can be interpreted as a cause-effect relationship [140].

The set of nodes and edges form the topology of the BN which specifies the conditional independence assumptions. Two equivalent statements can be used to define conditional independence in BNs: (1) a node is conditionally independent of its non-descendants[2] given its parents, or (2) a node is conditionally independent of all other nodes in the network given its parents, children, and children's parents (the so-called *Markov blanket*). To determine the global independence relationships of three sets of nodes (is X^1 conditionally independent of X^2 given X^3?) the *d-separation* criterion can be used [140] which is beyond the scope of this short introduction.

The full joint distribution of a set of random variables $X^0, ..., X^n$ as part of a BN can be written as

$$p(\mathbf{x}^0, ..., \mathbf{x}^n) = \prod_{i=1}^{n} p(\mathbf{x}^i|parents(\mathbf{x}^i)) \tag{2.1}$$

An example of a BN is shown in Figure 2.2. According to Equation 2.1, the joint distribution for this example can be written as

$$p(\mathbf{x}^0, \mathbf{x}^1, \mathbf{x}^2, \mathbf{x}^3) = p(\mathbf{x}^0)p(\mathbf{x}^1)p(\mathbf{x}^2|\mathbf{x}^0, \mathbf{x}^1)p(\mathbf{x}^3|\mathbf{x}^2) \tag{2.2}$$

[1] Nodes are numbered with superscripts here; subscripts are reserved for time steps (see Section 2.3).
[2] Descendants of a node are all its children, its children's children *etc.*

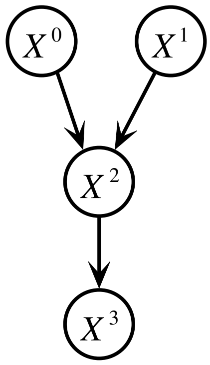


Figure 2.2: Example of a BN with four random variables (chance nodes).

2.2.2 Inference

Since BNs encode the full joint probability distribution of a domain, any query can be answered. A typical query asks for a probability distribution of a set of *query* nodes X given some *evidence* nodes Z which is referred to as *probabilistic inference* [148]. In mathematical terms, a query is written as $p(\mathbf{x}|\overline{\mathbf{z}})$ where $\overline{\mathbf{z}}$ emphasises the instantiation of Z with a value. Nodes in the BN which are neither query nor evidence nodes are denoted by Y. The answer to a query can be computed by using the joint distribution:

$$p(\mathbf{x}|\overline{\mathbf{z}}) = \frac{p(\mathbf{x}, \overline{\mathbf{z}})}{p(\overline{\mathbf{z}})} = \frac{\sum_{\mathbf{y}} p(\mathbf{x}, \mathbf{y}, \overline{\mathbf{z}})}{\sum_{\mathbf{x}} p(\mathbf{x}, \overline{\mathbf{z}})} \tag{2.3}$$

Expressions $\sum_{\mathbf{y}} p(\mathbf{x}, \mathbf{y}, \overline{\mathbf{z}})$ and $\sum_{\mathbf{x}} p(\mathbf{x}, \overline{\mathbf{z}})$ are called *marginalisations* of Y and X, respectively.

A special case of the general problem is the BN shown in Fig. 2.3 which is used to introduce a number of important terms. The BN contains two sets of nodes X and Z. Initial knowledge about $\mathbf{x}$ is represented as a *prior* probability distribution $p(\mathbf{x})$. The evidence entered on Z is called *observation* $\overline{\mathbf{z}}$. For this particular problem, Bayes' theorem can be applied:

$$p(\mathbf{x}|\overline{\mathbf{z}}) = \frac{p(\overline{\mathbf{z}}|\mathbf{x})p(\mathbf{x})}{p(\overline{\mathbf{z}})} \tag{2.4}$$

The answer to the query $p(\mathbf{x}|\overline{\mathbf{z}})$ is called the *posterior* encoding the belief of $\mathbf{x}$ after incorporation of observation $\overline{\mathbf{z}}$. Observation $\mathbf{z} = \overline{\mathbf{z}}$ is substituted into the CPD $p(\mathbf{z}|\mathbf{x})$ to yield a

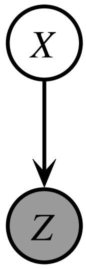


Figure 2.3: A simple BN to demonstrate Bayes' theorem and define important terms. The shaded node is observed.

likelihood function $p(\overline{\mathbf{z}}|\mathbf{x})$. The function represents a distribution over the values of the true state $\mathbf{x}$ and can be understood as a slice through the $\mathbf{x}-\mathbf{z}$ space. A likelihood function is simply referred to as a *likelihood* throughout the remainder of the thesis. Finally, $p(\overline{\mathbf{z}})$ acts as a normalisation constant.

For more complex BNs with many nodes, the conditional independence assumptions encoded in the graph topology can be exploited to design efficient inference algorithms. The simplest inference algorithm is *variable enumeration* which is demonstrated here using the BN shown in Fig. 2.2. The example query we want to compute is $p(\mathbf{x}^0|\overline{\mathbf{x}}^2)$. Applying Equation 2.3 to the joint of Equation 2.2 yields

$$p(\mathbf{x}^0|\overline{\mathbf{x}}^2) = \alpha \sum_{\mathbf{x}^1} \sum_{\mathbf{x}^3} p(\mathbf{x}^0) p(\mathbf{x}^1) p(\overline{\mathbf{x}}^2|\mathbf{x}^0, \mathbf{x}^1) p(\mathbf{x}^3|\overline{\mathbf{x}}^2) \tag{2.5}$$

with α being a normalisation constant. One simplification is to "push sums in" as far as possible which yields

$$p(\mathbf{x}^0|\overline{\mathbf{x}}^2) = \alpha p(\mathbf{x}^0) \sum_{\mathbf{x}^1} p(\mathbf{x}^1) p(\overline{\mathbf{x}}^2|\mathbf{x}^0, \mathbf{x}^1) \sum_{\mathbf{x}^3} p(\mathbf{x}^3|\overline{\mathbf{x}}^2) \tag{2.6}$$

In this case (but not in general), the last term sums up to 1. Variable enumeration has the problem that some computations are unnecessarily repeated. The *variable elimination* algorithm addresses this problem [148]. It implicitly creates a *junction tree* which is a data structure useful for efficient message passing [126].

All the algorithms mentioned so far are *exact* methods which are tractable for *polytrees* where there is at most one undirected path between any two nodes in the network. For

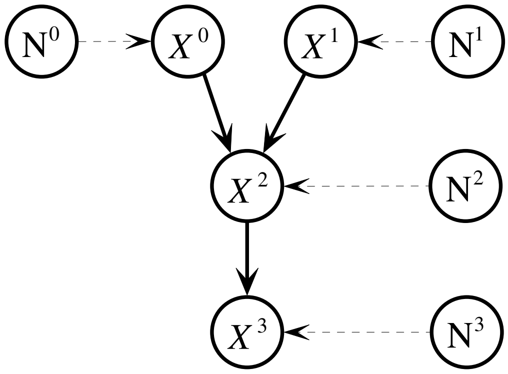


Figure 2.4: Parameters are added to the BN representation shown in Fig. 2.2. Dashed edges are used to distinguish between parameters and state variables.

larger, *multiply connected* networks, approximate inference methods are sometimes required. Among them are *Loopy Belief Propagation* (LBP), variational methods, and sampling techniques [126]. The discussion of these is beyond the scope of this chapter.

2.2.3 Learning and Expert Elicitation

Both the structure and the parameters of a BN can be learned from data. The problem can be approached by using Bayesian learning techniques and approximations of them. Here, parameter learning is briefly discussed.

Parameter learning can be formulated as *inference* on the parameters by adding them to the representation as shown in Figure 2.4. This is referred to a full Bayesian learning and yields a *distribution* over parameters.

Full Bayesian learning is often intractable. If data from all variables in the domain is available (full observability), *Maximum Likelihood* (ML) or *Maximum A Posteriori* (MAP) methods can be used as an approximation to full Bayesian learning. If some variables are unobserved during learning (partial observability), *Expectation Maximisation* (EM) can be applied. All three methods yield a point estimate rather than a distribution over parameters.

Besides learning from data, another possibility exists to find the structure and parameters of a BN model: elicitation from domain experts and the literature [45]. BNs are considered suitable for this method for two reasons: (1) it is natural for people to think in cause-effect relationships (structure), and (2) only local CPDs need to be specified (parameters). The number of CPDs which have to be specified is small if a causal model is strictly applied [148].

After a model has been fully specified, it is important to verify its validity, especially if the expert elicitation method is applied. Two prominent methods are *sensitivity analysis* [148] and *conflict analysis* [86]. Conflict analysis is briefly discussed next.

2.2.4 Data Conflict Analysis

According to Jensen, *"conflict analysis is the activity of detecting, tracing, and explaining possible conflicts among observations of variable values"* [86]. Jensen's conflict measure is adopted here: given a set of evidence nodes $Z = \{Z^0, ..., Z^n\}$, the conflict measure is defined as[3]

$$conf(\overline{\mathbf{z}}) = \log \frac{\prod_{i=0}^{n} p(\overline{\mathbf{z}}^i)}{p(\overline{\mathbf{z}})} \tag{2.7}$$

A potential conflict exists if the measure is positive which means that $\overline{\mathbf{z}}^0, ..., \overline{\mathbf{z}}^n$ are negatively correlated. A negative correlation between two variables $\overline{\mathbf{z}}^a$ and $\overline{\mathbf{z}}^b$ means that $p(\overline{\mathbf{z}}^a, \overline{\mathbf{z}}^b) < p(\overline{\mathbf{z}}^a)p(\overline{\mathbf{z}}^b)$ or $p(\overline{\mathbf{z}}^a|\overline{\mathbf{z}}^b) < p(\overline{\mathbf{z}}^a)$ and vice versa. In other words, observing $\overline{\mathbf{z}}^b$ makes it *less* likely to also observe $\overline{\mathbf{z}}^a$ which should not occur if the evidence supported each other.

2.3 Dynamic Bayesian Networks

BNs can be extended to Dynamic Bayesian Networks (DBNs) to model dynamic processes. DBNs are relevant to the robotics field because both the states of the environment and the platforms change over time. Murphy [126] showed how many common temporal estimation algorithms can be represented as DBNs, among them Hidden Markov Models (HMMs), Kalman filters, and their numerous extensions.

2.3.1 Representation

DBNs represent random variables evolving over time with each time slice containing a snapshot of the current state. DBNs make two assumptions: (1) the state changes according

[3] Logarithms to the base e are used here.

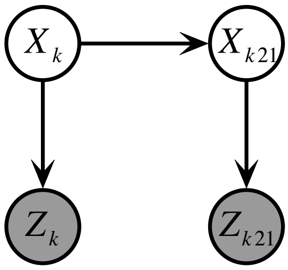


Figure 2.5: DBN example: first-order Markov model representing an HMM or KF dependent on whether X is discrete or continuous. $p(\mathbf{x}_{k+1}|\mathbf{x}_k)$ and $p(\mathbf{z}_k|\mathbf{x}_k)$ are transition and sensor model, respectively.

to a *stationary process*, and (2) the current state only depends on a finite history (*Markov assumption*).

The first assumption means that parameters of random variables do not change over time which avoids having to specify an unbounded number of CPDs. This is also referred to as *parameter tying* [126]. The second assumption limits the number of parents variables can have. The most commonly used Markov process is *first-order*: the future is independent of the past given the present.

A first-order Markov models is shown in Figure 2.5. As a result of parameter tying and the Markov assumption, only 2 time slices need to be shown to represent all CPDs in the model. States of two consecutive time slices are denoted by subscripts k and $k+1$, respectively. The evolving state is modelled using the *transition model* $p(\mathbf{x}_{k+1}|\mathbf{x}_k)$. The other CPD in Figure 2.5 is $p(\mathbf{z}_k|\mathbf{x}_k)$=$p(\mathbf{z}_{k+1}|\mathbf{x}_{k+1})$ and is known as the *observation model* or *sensor model*.

Two important classes of DBNs use the same conditional independence assumptions encoded in Figure 2.5: Hidden Markov Models (HMMs) and Kalman Filters (KFs). The difference is that X is discrete in HMMs while it is continuous Gaussian for KFs.

2.3.2 Inference and Learning

There are several kinds of inference which are of interest in DBNs. The major ones include *filtering*, *prediction*, and *smoothing* [126]. All of them can be expressed by the posterior $p(\mathbf{x}_{k+1}|\mathbf{z}_{0:\kappa})$: for filtering $\kappa = k+1$, for prediction $\kappa < k+1$, and for smoothing $\kappa > k+1$.

Filtering is used to estimate the current belief state based on all observations to date. Prediction is the task of computing the posterior distribution over future states given all observations to date. Smoothing is the task of computing a posterior distribution over a past state given all observations to date. It is used for parameter learning as described below.

All the algorithms above can be formulated in terms of forwards and backwards operators [126]. Different implementation of these operators exist which trade off accuracy with speed. In robotics, the most common inference problem is filtering which is presented next.

Recursive Filtering Filtering can be formulated recursively using Bayes' theorem (compare to Equation 2.4). The Bayesian filtering operation consists of two steps: a one-step prediction ($\kappa = k+1$) using the transition model followed by an update using the sensor model.

The update yields the posterior and is computed as follows:

$$p(\mathbf{x}_{k+1}|\mathbf{z}_{0:k+1}) = \frac{p(\overline{\mathbf{z}}_{k+1}|\mathbf{x}_{k+1})p(\mathbf{x}_{k+1}|\mathbf{z}_{0:k})}{p(\overline{\mathbf{z}}_{k+1}|\mathbf{z}_{0:k})} \tag{2.8}$$

where $\overline{\mathbf{z}}_{k+1}$ is an observation of state $\mathbf{x}$ at time step $k+1$, $p(\mathbf{x}_{k+1}|\mathbf{z}_{0:k})$ is the prediction of the state from the previous time step (the prior), and $\mathbf{z}_{0:k+1}$ summarises all observations up to time step $k+1$. $p(\overline{\mathbf{z}}_{k+1}|\mathbf{x}_{k+1})$ is a likelihood computed by substituting an observation $\overline{\mathbf{z}}_{k+1}$ into the sensor model $p(\mathbf{z}|\mathbf{x})$.

The prediction step is given by the Chapman-Kolmogorov equation:

$$p(\mathbf{x}_{k+1}|\mathbf{z}_{0:k}) = \sum_{\mathbf{x}_k} p(\mathbf{x}_{k+1}|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{z}_{0:k}, \mathbf{x}_0) \tag{2.9}$$

where $p(\mathbf{x}_k|\mathbf{z}_{0:k}, \mathbf{x}_0)$ is the estimate from the previous time step.

Learning Techniques for learning the parameters of a DBN are straight-forward extensions to the techniques presented in Section 2.2.3. For *offline* learning, smoothing algorithms are used which take all observations into account ("learning with hindsight"), *i.e.* $p(\mathbf{x}_k|\mathbf{z}_{1:K}, \Theta)$ is computed for all k with K being the length of the entire time sequence.

For *online* learning, the parameters can be added to the representation and the filtering algorithm estimates both the states and the parameters, *i.e.* $p(\mathbf{x}_k, \Theta|\mathbf{z}_{1:k})$ is computed.

2.4 Influence Diagrams

BNs can be extended to *Influence Diagrams* (IDs) to model decision making under uncertainty [81]. IDs are generally able to represent information about the current state, possible actions, the state resulting from the action, and the utility of that state [148]. IDs require the decision and chance nodes to be discrete [85].

2.4.1 Representation

IDs extend BNs by adding *decision* and *utility* nodes. Decision nodes represent choices available to the decision-maker – a set of possible actions. Utility nodes encode a utility function: the usefulness of the consequences of decisions using a scalar called utility[4].

In addition to the DAG property inherited from BNs, two more structural properties are part of an ID: (1) there is a directed path comprising all decision nodes, and (2) utility nodes do not have children. The directed path property is required to ensure a temporal sequence of decisions. This property applies even if the decisions are independent which is hard to determine [85].

For visualisation, standard conventions are adopted here: chance nodes are drawn as circles, decision nodes as squares, and utility nodes as diamonds. An example of an ID is shown in Figure 2.6. X and $Y = \{Y^0, ..., Y^n\}$ denote world states and $Z = \{Z^0, ..., Z^m\}$ denotes observations. The ID contains a single decision node D with no chance nodes as children implying that the decision has no effect on the world states. In this case, the decision is also referred to as a *non-intervening* decision. The utility node U depends on its parents, decision node D and chance node X. The network topology within the "supernode" containing Y and Z nodes is arbitrary.

[4]How to construct utility functions is not covered here, see *e.g.* [194][14].

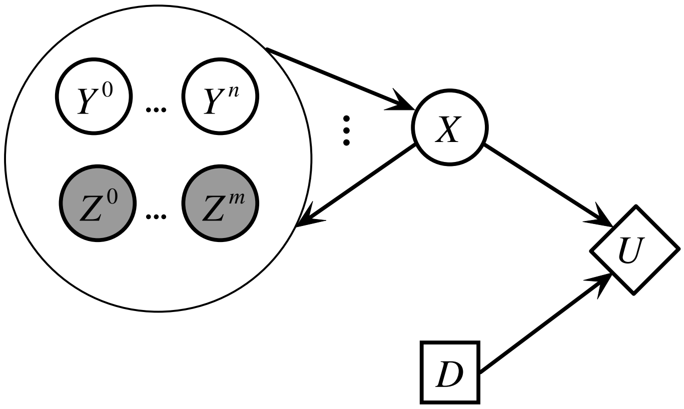


Figure 2.6: Influence diagram with a single non-intervening decision node D and single utility node U. The world is represented by X and $Y = \{Y^0, ..., Y^n\}$ while observations are represented by $Z = \{Z^0, ..., Z^m\}$. Z nodes are shown shaded. A "supernode" contains all Y and Z nodes with an arbitrary network topology.

2.4.2 Making Decisions

For the case shown in Figure 2.6, the best action can be found as follows. First, compute the *expected utility* (EU) of *action* vector $\mathbf{d}$ given observations $\overline{\mathbf{z}}$:

$$EU(\mathbf{d}|\overline{\mathbf{z}}) = E^{p(\mathbf{x}|\overline{\mathbf{z}})}\{U(\mathbf{x}, \mathbf{d})\} = \sum_{\mathbf{x}} U(\mathbf{x}, \mathbf{d}) p(\mathbf{x}|\overline{\mathbf{z}}) \tag{2.10}$$

where the computation of $p(\mathbf{x}|\overline{\mathbf{z}})$ marginalises out Y. Second, choose the action d^* which maximises the EU:

$$d^* = \underset{\mathbf{d}}{\operatorname{argmax}}\, EU(\mathbf{d}|\overline{\mathbf{z}}) \tag{2.11}$$

The EU for a single decision node can also be computed without making the assumptions of a non-intervening decision and a single utility node. For i utility nodes each having a parent X^i, the EU is

$$EU(\mathbf{d}|\overline{\mathbf{z}}) = \sum_{i} \sum_{\mathbf{x}^i} U^i(\mathbf{x}^i) p(\mathbf{x}^i|\mathbf{d}, \overline{\mathbf{z}}) \tag{2.12}$$

where $p(\mathbf{x}^i|\mathbf{d}, \overline{\mathbf{z}})$ now also depends on the decision and thus describes its effects on the state $\mathbf{x}^i$.

The general decision problem with *multiple* decision nodes is to find the best *sequence* of decisions. The solution to the problem is a sequence that maximises the expected utility. One possibility is to convert the ID to a decision-tree [81]. In general, however, the tree is of exponential size.

Shachter developed an alternative algorithm which transforms the ID by a series of node-removal and arc-reversal operations [157]. Refinements of that algorithm increased its computational efficiency [158]. Jensen describes an efficient method for solving IDs using strong junction trees [85]. It represents an extension to the junction trees used in pure Bayesian decision analysis. Work by Lauritzen & Nilsson links algorithms for IDs to developments in clustering algorithms for Bayesian networks [105].

2.4.3 Value of Information

Rather than taking the action d^* in Equation 2.11, a decision-maker might have the choice of consulting one of its information sources $\{I^0, ..., I^p\}$ in order to generate a more informed decision. Consulting an information source I^i is equivalent to obtaining the state of that chance node. For this thesis, it is assumed that only a single information source can be consulted at any given time which is referred to as *myopic* information gathering [148].

It is possible to calculate what we can *expect* to gain from consulting the information source *before* observing that node by using its belief given all current evidence $p(\mathbf{i}^i|\overline{\mathbf{z}})$ [42]. The expected utility of the optimal action (EUO) *after* having observed I^i is

$$EUO(\mathbf{i}^i, \mathbf{d}|\overline{\mathbf{z}}) = \sum_{\mathbf{i}^i} p(\mathbf{i}^i|\overline{\mathbf{z}}) \max_{\mathbf{d}} EU(\mathbf{d}|\overline{\mathbf{z}}, \mathbf{i}^i) \tag{2.13}$$

The value of observing I^i is called the *Value Of Information* (VOI). It is calculated as the difference between the expected utility after having observed I^i and the currently available maximum expected utility:

$$VOI(\mathbf{i}^i, \mathbf{d}|\overline{\mathbf{z}}) = EUO(\mathbf{i}^i, \mathbf{d}|\overline{\mathbf{z}}) - \max_{\mathbf{d}} EU(\mathbf{d}|\overline{\mathbf{z}}) \tag{2.14}$$

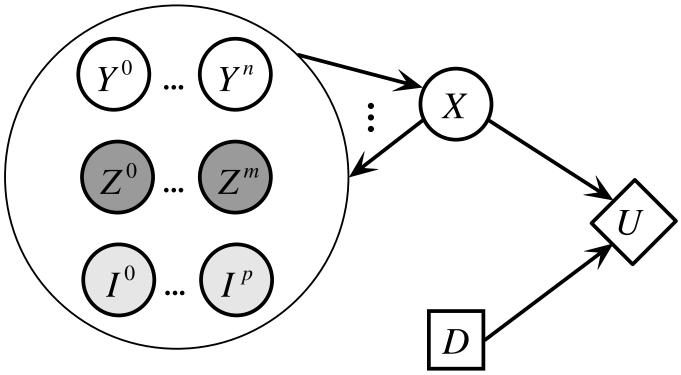


Figure 2.7: Extension of Figure 2.6 by a set of information sources $I = \{I^1, ..., I^p\}$ shown in light grey.

Equation 2.14 is valid for all network topologies. If there are several decision nodes, the directed path property determines the temporal order as mentioned above. Before each decision is made, Equation 2.14 can be used to decide whether to obtain more information first.

Figure 2.7 shows an extended version of Figure 2.6 with potential information sources added to the representation. For this case, Equation 2.14 can be computed as:

$$\begin{aligned} VOI(\mathbf{i}^i, \mathbf{d}|\overline{\mathbf{z}}) &= \sum_{\mathbf{i}^i} p(\mathbf{i}^i|\overline{\mathbf{z}}) \max_{\mathbf{d}} \left(\sum_{\mathbf{x}} p(\mathbf{x}|\mathbf{i}^i, \overline{\mathbf{z}}) U(\mathbf{d}, \mathbf{x}) \right) \\ &\quad - \max_{\mathbf{d}} \left(\sum_{\mathbf{x}} p(\mathbf{x}|\overline{\mathbf{z}}) U(\mathbf{d}, \mathbf{x}) \right) \end{aligned} \tag{2.15}$$

Computing the VOI is relevant for intelligent information gathering systems where the goal is to maximise the amount of information collected [196]. Consulting an information source comes at a cost, so a sensible strategy is to consult that source only if the expected benefit is higher than the cost $C(\mathbf{i}^i)$:

$$VOI(\mathbf{i}^i, \mathbf{d}|\overline{\mathbf{z}}) - C(\mathbf{i}^i) > 0 \tag{2.16}$$

A central aspect of this thesis is to regard human operators as information sources for human-robot information fusion as presented in the next chapter. In this context, Equation 2.16 can be interpreted as follows: if the cost for obtaining information from a human operator is smaller than the value this information is expected to add, then ask the operator for information. This mechanism will be experimentally demonstrated in Chapter 5.

2.5 Summary

This chapter presented the background on probabilistic methods for representing the perception and decision making of mobile robots. The focus was on representations considered useful for human-robot information exchange, namely Bayesian Networks (BNs), Dynamic Bayesian Networks (DBNs), and Influence Diagrams (IDs). The literature for these representations is vast and only aspects which are required to understand subsequent experimental chapters were presented.

In the context of BNs and IDs, probabilistic data types which are important throughout the rest of the thesis were introduced: they include *observation*, *prior*, *likelihood*, *posterior*, *parameter*, *action*, and *utility function*.

Figure 2.8 shows how the internals of the perception and decision models (previously shown in Figure 2.1) can be realised using an ID. Both the environment and the platform is represented by two abstraction levels (four white chance nodes of the perception model). The sensor and user models for perceiving environment and platform states are represented by the CPDs attached to the grey evidence nodes. Decisions are made based on the beliefs of the chance nodes, and the corresponding utility functions. The actuation model encodes the effects of a platform decision in the example (bottom right). Operators can directly enter actions as indicated by the decision node in the user model box (top right). The next chapter will show how to utilise such a representation for human-robot information fusion.

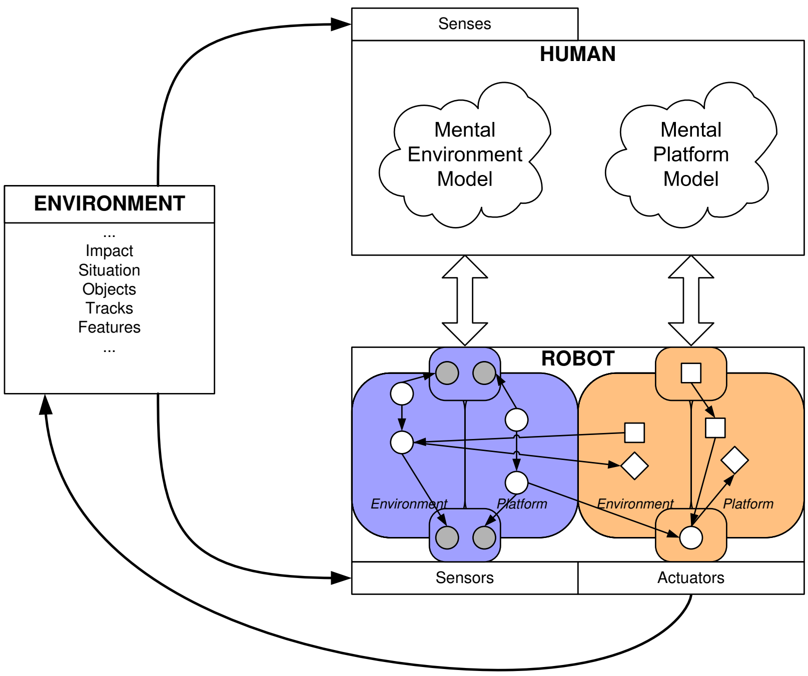


Figure 2.8: Influence diagram representation for robotic perception and decision making (compare to Figure 2.1): environment and platform are represented by two nodes each (white circles) which are observed by robotic sensors and human operators (grey circles). Decisions (squares) are made according to the beliefs of the chance nodes, and the utility functions (diamonds).

Chapter 3

Humans as Information Sources

As discussed in the previous chapter, many aspects of critical algorithms in today's robotic systems are probabilistic. This chapter starts by exploring the suitability of probabilistic data types for *bidirectional* human-robot communication. Subsequently, the communication direction is restricted to a human-to-robot flow which introduces the concept of treating humans as *information sources* (see Figure 3.1). Information is defined as one of two probabilistic data types: *observations* and *likelihoods*. The purpose of letting operators submit information is to enable the probabilistic fusion with information acquired by robotic sensors. Human-robot information fusion is used to fulfil the objective of combining the strengths of the two complementary information sources.

Arguments are presented as to why information is considered an adequate data type for human data entry. It is shown that for a large class of semi-autonomous systems, this type of human-robot communication is sufficient. Two questions arise: (1) what representations are suitable to incorporate information from human operators, and (2) how to treat human input. The first question is addressed by discussing the suitability of multi-level probabilistic representations. The second question is addressed by showing how probabilistic user models can be incorporated into a multi-level representation.

The concept of treating humans as information sources as part of probabilistic reasoning and decision making is a novel contribution to the field of HRI. The potential benefit of this concept has recently gained attention in the context of crisis management [139][122] where people in the field may be able to contribute valuable information after natural or man-made disasters have occurred. Another application area where operators may add information to

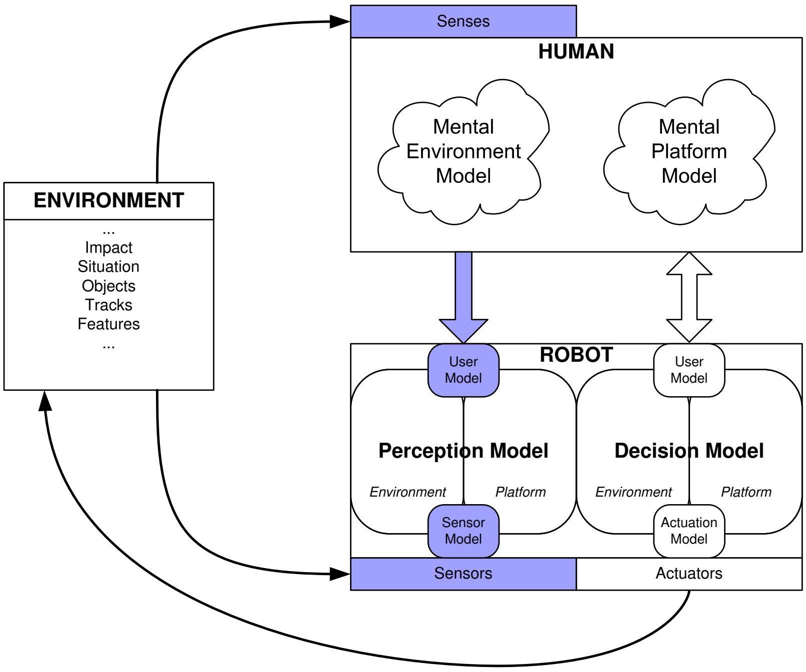


Figure 3.1: This chapter introduces the concept of regarding humans as information sources. Human operators submit perceptual information which is transformed by a user model and subsequently fused with robotic sensor information.

a semi-automated reasoning system is command-and-control [146][15]. In all of this work, the integration of operators is often discussed conceptually only, and results from fusing human and sensor information are limited to date. The second contribution of this chapter is the presentation of methods for probabilistic modelling of humans as sensors. Some of the work mentioned above recognises the need for modelling the reliability of human input [122] but no results have been reported. The third contribution is the utilisation of probabilistic multi-level representations for human-robot information fusion. In the literature, multi-level models are prominent for many application areas, but typically, input at higher abstraction levels is not considered [109]. A detailed literature review is presented at the end of the chapter.

The remainder of the chapter is organised as follows. Section 3.1 presents use cases of probabilistic data types for typical robotics applications. Human entry of information is contrasted with human entry of actions. Section 3.2 discusses the implications of regarding human operators as sensors. The differences between robotic and human sensors are listed and probabilistic representations are proposed to incorporate the differences. The *Human*

Sensor Model (HSM) is introduced as a probabilistic method to model user uncertainties. Section 3.3 discusses the limitations of regarding human operators as information sources. Section 3.4 presents simulated results from a set of example models demonstrating both the advantages and limitations of the approach. Section 3.5 presents results from a calibration experiment which employed human subjects to build a HSM for range observations. Section 3.6 presents related work before Section 3.7 summarises.

3.1 Probabilistic Human-Robot Interactions

The approach advocated in this thesis is to make use of established probabilistic representations for bidirectional human-robot communication. This can be seen as a *robot-centred* approach: given current robotics representations, methods to incorporate human operators are investigated. As Fong pointed out in [57], robot-centred does not mean that the robot is "in charge": the human is rather seen as a resource or collaborator to the robot. Furthermore, robot-centred does not contradict the goal of *human-centred* robotics which takes *"a philosophical stance on building technology that serves human needs"* [26]. More specifically, a robot-centred approach to human-robot communication is defined here as taking established probabilistic representations as a starting point for developing communication schemes.

3.1.1 Probabilistic Data Types

The type of data to be communicated between humans and robots is well-defined using our robot-centred approach: all relevant information is encoded in the probabilistic representation. As presented in Chapter 2, Bayesian inference includes the following data types: *prior belief*, *observation*, *likelihood*, *posterior belief*, and *parameter* (which include sensor and transition model). If decisions need to be made, additional data types are *action* and *utility function*. Figure 3.2 summarises the available data types and emphasises what is referred to as *information* in this chapter.

Given a probabilistic representation of the state of the environment and the robotic platforms, all communication patterns mentioned in Section 1.3.2 can be realised. Table 3.1 lists examples of robotics applications with corresponding communication patterns and messages. The examples are briefly described next.

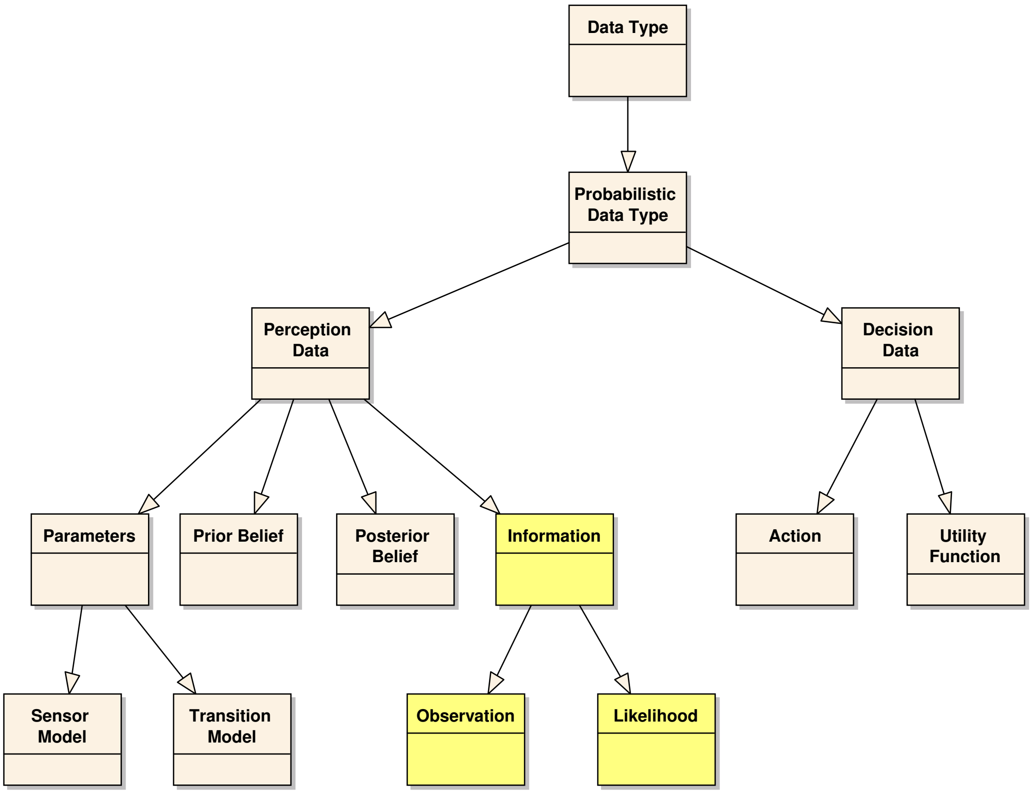


Figure 3.2: Probabilistic data types as a UML class diagram. Arrows indicate a “is-a” relationship. The “root” node also appears in Figure 1.4. The highlighted data types are submitted by human operators when acting as information sources.

Teleoperation resembles a simple human-push pattern with human operators constantly sending actions to the robot. Similarly, in a manipulation task, a robot may be used to change the environment by submitting low-level or high-level actions (*e.g.* “move that chair”). In many Adjustable Autonomy (AA) systems, human operators may specify actions at different abstraction levels when taking over control. Examples include safe-guarded teleoperation and the specification of high-level task goals [23]. Section 5.4.3 discusses related work in AA in more detail.

The human role of monitoring the robots’ task and health can be implemented using robot-push or human-pull. Robot-push is more commonly applied, *e.g.* frequent updates of a feature map on a GUI display during a map building scenario. Relaying current beliefs including uncertainty to operators can contribute significantly to improve their Situation Awareness (SA) [49]. In the map building example mentioned above, the (Gaussian) un-

Application	Comms. patt.	Message	Prob. data type
Teleoperation	human-push	plat.-dec.	action
Manipulation	human-push	env.-dec.	action
Adjustable autonomy	human-push, robot-pull	plat.-dec.	action
Task monitoring	robot-push, human-pull	env.-perc.	belief, parameter
Health monitoring	robot-push, human-pull	plat.-perc.	belief, action, parameter
Human-robot fusion	***human-push, robot-pull***	***env.-perc., plat.-perc.***	***observation, likelihood***

Table 3.1: Examples for human-robot communication using probabilistic data types. The focus of this thesis is emphasised. Abbreviations are: env. = environment, plat. = platform, perc. = perception, dec. = decision.

certainty of point features are often displayed using ellipses. For monitoring purposes, operators may also be interested in the robots' current actions, *e.g.* their speeds and turn-rates. If online learning is part of the system, operators may also want to receive updated parameter values.

This thesis focuses on the application highlighted in Table 3.1: human-robot fusion. Observations and likelihoods are identified as appropriate data types for this application. Human-robot information fusion can be implemented by either human-push (Chapter 4) or robot-pull (Chapter 5). In the following, it is argued that submitting information can be superior to submitting actions for robot control.

3.1.2 Information Vs. Action

The benefits of submitting information rather than actions for robot control are related to modularity, scalability, reuse, and peer-to-peer interaction. Each one is discussed below while limitations of this approach are presented in Section 3.3.

Modularity Submitting actions is equivalent to making decisions. For operators to make "good" decisions, complete SA may be required. In contrast, observations can be made on smaller parts of the represented, potentially complex, scenario.

Graphical models are well-suited to represent complex scenarios. According to Jordan [89], a fundamental concept of graphical models is the notion of modularity: a complex system is built by combining simpler parts. In this work, BNs are used to minimise complexity by assuming conditional independence between variables. Modularity of a representation can be beneficial for human information submission: operators are only required to understand a subproblem, not the entire model.

As a consequence, by using the approach advocated in this work, non-experts can add information about simple, intuitive subproblems which do not require the understanding of an entire situation. The information entered in this way propagates throughout the representation using inference mechanisms as described in Section 2.2.2, and ultimately leads to well-informed decisions.

Reuse If more than one decision needs to be made, information supplied to the system may influence all of the decisions depending on the representation's structure. Thus, a single piece of information is "reused" to make several decisions simultaneously which is not achievable if operators were to specify actions directly.

Information can also be reused in a temporal sense. If the representation keeps a history, which is typically done for evolving phenomena, fused information remains in the system and influences actions at all future time steps. Filtering using a DBN as discussed in Section 2.3 is an example of how to maintain a history. In contrast, actions cannot be reused in the future.

Scalability In robotic systems, actions are typically related to a specific robotic platform. This includes higher-level actions such as path planning decisions. Specifying actions for individual platforms does not scale to multi-robot systems, and actions cannot be "shared" among platforms.

In contrast, beliefs related to the environment can be shared among all platforms, and thus do scale well [115]. To share beliefs, state estimation needs to be decentralised which is presented in Chapter 4. Based on the beliefs of the shared environment states, actions can be triggered for all platforms in the system [71].

Robot as a Partner Letting operators submit information rather than actions reinforces the notion of acting on the same level of authority: humans and robots collect information cooperatively and the model produces the decision. This approach is different from many

AA systems where operators take over control (see Section 5.4.3). Taking over control can be seen as overriding the robot's decisions whereas our approach resembles true peer-to-peer interaction. Viewing the robot as a partner rather than a tool is considered a prerequisite for truly intelligent human-robot systems by many HRI researchers [52][117].

3.2 Human as a Sensor

This section analyses the implications of regarding human operators as sensors. First, the applications of human-robot fusion presented in this thesis are outlined. Then, the differences between robotic and human sensors are presented which impose certain challenges on the representation employed for human-robot information fusion. Multi-level Bayesian Networks (BNs) are able to address these challenges and are capable of incorporating user models. User models are named *Human Sensor Models* (HSMs) and are introduced afterwards.

3.2.1 Applications

Two robotics problems are considered in this thesis: (1) cooperative information gathering, and (2) cooperative decision making.

The objective for information gathering is to build a representation of a spatially distributed phenomenon. Information gathering can be conducted by a single robot, a team of networked robots, or a sensor network [113]. Independent of the system type, information from multiple sources is usually combined into a single representation. Typically, human operators either engage in a monitoring activity as part of these systems [100], or make decisions based on the collected information [34][15]. The novel approach presented here is to let operators contribute to the information gathering task by submitting information in a similar way to robotic sensors. Thus, humans become an integral part of the information gathering system. Chapter 4 presents results from an experiment using a heterogeneous human-robot team deployed to gather information in an outdoor environment.

The second problem is cooperative decision making: robots have a task to fulfil and therefore need to make certain decisions. In this scenario, there are more options for human involvement compared to the task of "pure" information gathering. The most straightforward interaction is teleoperation as described in Section 3.1.1. Operators can also make higher-level decisions and enter them into the representation as commonly done in systems employing supervisory control [159]. The approach presented here is different: as for information gathering, operators submit information and let the robot decide what to do based on the additional information. Chapter 5 presents results from robotic navigation experiments employing this mechanism.

3.2.2 Robotic Vs. Human Sensor

The qualities of human sensors differ vastly from those of robotic sensors. This section discusses the differences and shows how a probabilistic representation can address them. Differences are identified in the following categories: abstraction, resolution, time scale, uncertainty, and reliability.

Abstraction Robotic sensors perform well in low-level descriptions such as geometric properties. Measuring the range and bearing to a point feature using a laser scanner is an example. In contrast, human operators are valuable in contributing more abstract properties. An example is object recognition which is trivial to humans but as a general problem, remains unsolved using vision algorithms [37].

Accuracy Robotic sensors are typically very accurate in the property they measure whereas humans function on a more qualitative level. A laser range scanner, for example, can measure the distance to an object with an accuracy in the order of centimetres. For humans, it is more natural to categorise coarsely. Examples of human distance observations are "between 1 and 2 meters" or "close".

Time scale Robotic sensors can make periodic observations at high frequency. For humans, it is more natural to asynchronously submit observations whenever there is spare capacity to do so[1].

[1]For control tasks, human capacity is limited to a bandwidth below $1Hz$ [160].

Uncertainty Both robotic sensors and human operators are uncertain in their measurements. The peculiarity of humans is their individual difference: while robotic sensors perform similarly across a product range, individual humans differ greatly in their perception.

Variability Related to the previous point is the variability of observations. Variability is not only caused by differences *between* individuals but can also stem from the *same* individual. Many external factors make an individual perform differently under different circumstances. This is less of a problem for robotic sensors which may show variability on a smaller scale due to temperature changes for instance.

The differences listed above illustrate that robots and humans have complementary perceptual abilities. Combining these abilities offers an opportunity for effective information fusion. The approach presented in this work is to use a BN representation for human-robot information fusion which is discussed next.

3.2.3 Suitability of BNs

BNs are capable of incorporating the differences between human and robotic sensors. Multiple *abstraction* levels are naturally encoded by the cause-effect structure of BNs. Hierarchical Hidden Markov Models (HHMM) are an example of a model addressing processes with a hierarchical multi-level structure [126].

An important advantage of a hierarchical BN is the information flow between levels in the hierarchy: lower level observations propagate to higher levels via inference, and thus change the beliefs of higher level states. This information can be communicated to human operators for monitoring purposes (human as information sink). Information flow also works in reverse: high-level human information entry changes lower-level beliefs (human as information source).

Different *accuracy* levels can be represented by continuous and discrete random variables, respectively. These networks are often referred to as *Hybrid* BNs [148]. Multiple *time scales* can be modelled using *Dynamic* BNs such as a variable-duration semi-Markov HMMs [126]. *Uncertainty* and *variability* of the human sensor can be addressed by user modelling techniques as described next.

3.2.4 Human Sensor Model

Two probabilistic data types are considered suitable for human-robot information fusion as shown in Table 3.1: likelihoods and observations. The human entry modes using likelihoods and observations are subsequently called *likelihood mode* and *raw observation mode*. The former is model-free while the latter requires a user model to transform between raw observations and the probabilistic representation. The probabilistic user model is called *Human Sensor Model* (HSM) in this work. Both of the modes mentioned above were applied in the experiments presented in Chapters 4 & 5 and are described next.

Likelihood mode In the likelihood mode, human information is entered as likelihood functions which include uncertainty. Humans can be seen as operating in state space using this mode. If a Gaussian representation is used for example, a likelihood function would consist of a mean and a standard deviation. For a binary representation, a ratio would be specified, *e.g.* $5:1$.

In this mode, operators are explicitly allowed to add uncertainty to their observations. Operators apply their mental models of the environment and platforms (see Figure 1.3) to estimate the uncertainty of their observations. The problem with this approach is that these estimates may be erroneous. Furthermore, as discussed above, individual users may have very different mental models which depend on many parameters that are time-variant and situation-dependent. Therefore, it is desirable to let the system decide how much uncertainty is associated to a raw human observation. This motivates the idea of modelling human operators as sensors which is discussed next.

Raw observation mode In the raw observation mode, operators make observations which the system converts into likelihoods by applying a HSM. If an n-dimensional Gaussian representation is used, a raw observation consists of a point in n-dimensional space which is converted into an (unnormalised) n-dimensional Gaussian using the HSM. Section 3.5 presents an example of how to build a Gaussian HSM using a calibration experiment. For a binary representation, one of the two binary states is specified. The HSM adds uncertainty to the observation which yields a binary likelihood, *e.g.* $3:1$.

3.3 Limitations

Human-robot information fusion as advocated in this work requires an *a priori* and fixed probabilistic representation. In this section, some of the limitations with this approach are identified. Suggestions are made of how to overcome the limitations in the future.

3.3.1 Model Complexity and Accuracy

Fusing human and robotic information relies heavily on the quality of the probabilistic fusion model. This section discusses the impact of model complexity and accuracy on the expected fusion results.

Complexity According to Occam's razor, the simplest model is the most preferable one [193]. The method advocated here may require complex models since higher-level states are needed to enter human observations. For some problems, there may not be a need to model these states for several reasons: (1) there is no interest in them, (2) no decisions depend on them, (3) the states are hard to model. The last point is a difficult problem, especially if little training data is available. An overly complex model may produce worse fusion results due to overfitting [46].

Accuracy It is widely acknowledged that probabilistic models do not have to be overly accurate in terms of the parameter values. In fact, "*probability provides a way of summarising the uncertainty that comes from our laziness and ignorance*" [148]. Despite this favourable property, human input cannot improve the accuracy of the fusion results if the model's parameters or structure are inappropriate. In fact, it may be feasible to refine the model's parameters and structure online using human input. This is a research area subject to future work as discussed in Section 6.2.1.

3.3.2 Conflicting Evidence

User modelling is a hard problem, mostly due to the individuality of humans and their variation in performance. There are ways of mitigating the problem of individuality, *e.g.* by using stereotypical profiles as presented in Chapter 5 or adapting models online as presented in [96].

Even if an appropriate user model exists, the danger remains that information entered by humans is inconsistent with each other or with information collected by robotic sensors. The problem is especially severe in a system where multiple operators and sensors are involved. A way to identify conflicting evidence in discrete BNs was presented in Section 2.2.4. Conflict resolution is often done manually by an analyst [86]. To automate the process is an avenue of future research: one possibility is to "gate" suspicious observations which is a common method in classical data association applications [11].

3.3.3 Human Factors

Section 3.1.2 argued that letting operators enter perceptual information rather than actions is a more peer-like interaction mode. Information propagates throughout the BN and decisions are made based on the inferred beliefs. As a consequence, the behaviour of the robot is not predictable to an operator who does not have a valid mental model of the system [130]. A feeling of "not being in control" may irritate users and may lead to less trust towards the automated system. This is an interesting hypothesis which could be validated in a user study. Future work as presented in Section 6.2.2 will investigate how to generate good mental models based on explaining internal reasoning and decision making to operators.

For some applications, a supervisory interaction scheme may be more appropriate [68]. A trade-off between simplicity of interaction (achieved *e.g.* by specifying actions) and building more intelligent human-robot systems is acknowledged here [117]. In practice, a combination of supervisory and peer-to-peer interaction modes is expected to be most effective for many application domains, given the state-of-the-art in robotics [68].

Other benefits of exploiting human decision-making skills (as opposed to perceptual skills) are presented in Section 6.2.1.

3.4 Example Models

This section presents simulation results from three simple human-robot fusion models. The simulations are conducted to demonstrate both the benefits and limitations of the approach.

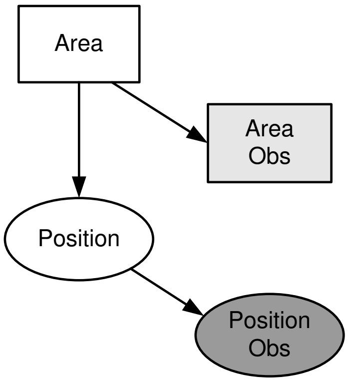


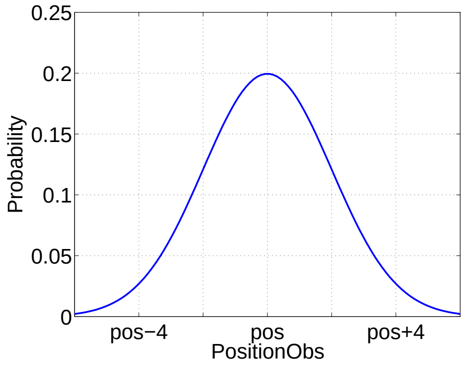
Figure 3.3: Static example model for human-robot information fusion. *Position* is continuous and observed by the robot (*PositionObs* in dark grey) while *Area* is discrete and human-observable (*AreaObs* in light grey).

3.4.1 Static Model

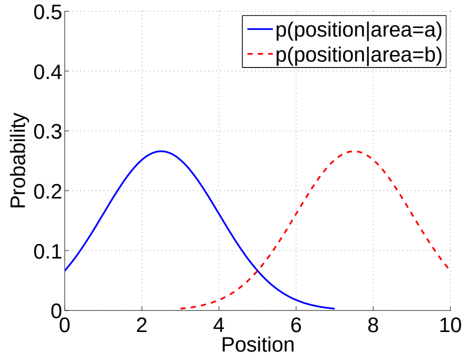
Figure 3.3 shows a two-level, static, hybrid BN for human-robot information fusion. The low-level *Position* variable represents the continuous one-dimensional position of a robot. An absolute position sensor (*e.g.* GPS) is used to observe *Position*. The sensor is erroneous and modelled as shown in Figure 3.4(a). The high-level *Area* variable represents two areas a and b the robot can be located in. Roughly, *Area* a reaches from 0 to 5 while *Area* b reaches from 5 to 10 which is modelled using two conditional Gaussian distributions. Their means are located in the middle of each area as shown in Figure 3.4(b). The *Area* variable is observed by a human using both the raw observation and likelihood modes as discussed in Section 3.2.4. The HSM required for the raw observation mode is encoded as a Conditional Probability Table (CPT) as shown in Figure 3.4(c).

The first property of human-robot information fusion demonstrated here is the information flow between abstraction levels. Figure 3.5 visualises results for different sets of observations entered into the static model shown in Figure 3.3: no observations, human-only observation, robot-only observation, and human-robot observations. Figure 3.5(a) shows the result of inference for *Area* and *Position* when no observations have been entered: areas a and b are equally likely and the Gaussian distribution is flat across all positions. The two distributions represent a non-informative prior. All following figures are to be compared to this situation.

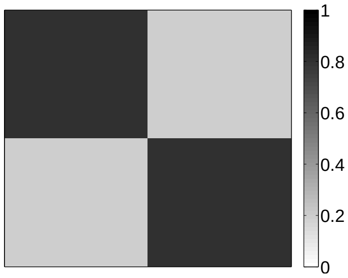
Figure 3.5(b) shows the result when the human operator makes the observation $\overline{areaObs} = a$: the probability for a increases to 0.8 (using the HSM) while the information also flows to



(a) Robotic SM: $\mathcal{N}(position, \sigma = 2)$



(b) $p(position|area)$



(c) HSM: $p(areaObs|area)$

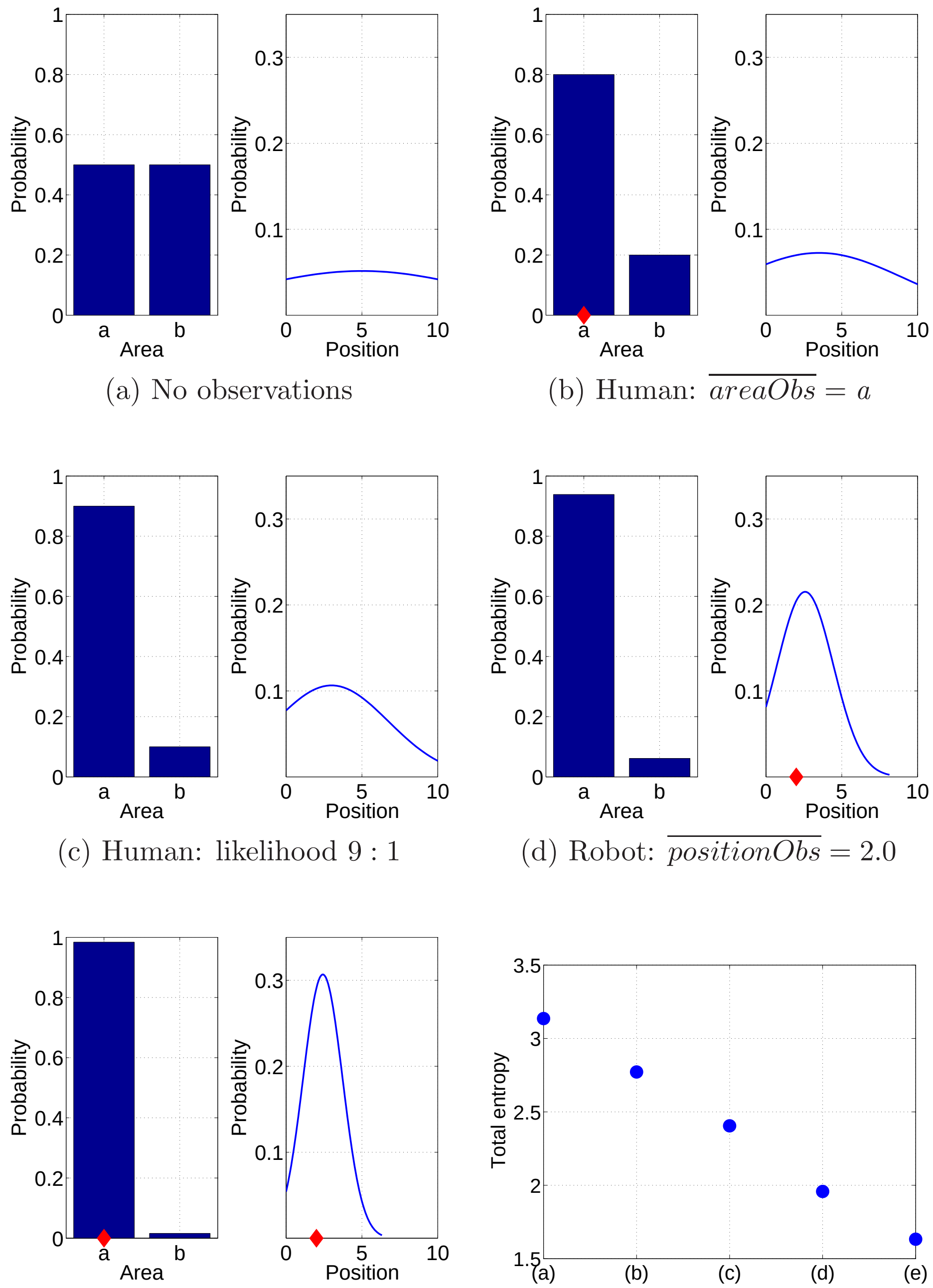
Figure 3.4: Parameters of the static example model: (a) the robotic sensor model represented as a conditional Gaussian distribution; (b) the conditional probability distribution $p(position|area)$; (c) the HSM represented as a $2x2$ CPT. Rows are probability distributions and add up to 1.0 while columns represent likelihood functions $p(\overline{areaObs}|area)$.

the *Position* node which makes the Gaussian become more compact. Figure 3.5(c) shows the result from using the other human input mode: the operator specifies a likelihood ratio[2] of $a : b = 9 : 1$ which increases the probability for a to 0.9. As before, information also flows to the *Position* node. The reverse information flow is shown in Figure 3.5(d): the robotic sensor makes an observation $\overline{positionObs} = 2.0$ which changes both the *Position* and the *Area* distributions.

So far, the resulting distributions for robot-only and human-only information entry have been presented. The second property of human-robot information fusion demonstrated here is the improvement of the fusion results. Figure 3.5(e) shows the result of combining human and robotic observations (scenarios (b) & (d)): the distributions are more certain[3].

[2] This is also referred to as entering *soft* or *virtual evidence* on the *Area* node [140].

[3] Scenarios (c) & (d) could be combined accordingly which is omitted for brevity.



(a) No observations

(b) Human: $\overline{areaObs} = a$

(c) Human: likelihood $9:1$

(d) Robot: $\overline{positionObs} = 2.0$

(e) Human-robot fusion

(f) Entropy

Figure 3.5: Simulation results from the static example model. (a) No evidence is entered into the network, beliefs are "uniform". (b) Human operator enters a raw observation indicated by red marker. Information also flows to the *Position* variable. (c) Human operator enters likelihood ratio $9:1$. Information also flows to the *Position* variable. (d) Robot makes an observation indicated by the red marker. Information also flows to the *Area* variable. (e) Human-robot fusion indicated by two red markers gives the most compact distributions. (f) The total entropy for the scenarios (a)–(e).

A quantitative measure of how much information is contained in a distribution is the *Entropy* or *Shannon information* [135]. The more compact a distribution, the more information it contains. For a discrete distribution, entropy is defined as[4]

$$H_p(\mathbf{x}) = -\sum_{\mathbf{x}} p(\mathbf{x}) \log p(\mathbf{x}) \tag{3.1}$$

For a continuous distribution, entropy is defined as

$$H_p(\mathbf{x}) = -\int p(\mathbf{x}) \log p(\mathbf{x}) d\mathbf{x} \tag{3.2}$$

which resolves to the following expression if the distribution is a n-dimensional Gaussian with a covariance matrix P:

$$H_p(\mathbf{x}) = \frac{1}{2} \log[(2\pi e)^n |P|] \tag{3.3}$$

The sum of the *Area* and *Position* entropies are computed for the five scenarios shown in Figure 3.5(a)–(e). The result is shown in Figure 3.5(f): most information is contained in the distribution resulting from human-robot information fusion.

3.4.2 Dynamic Model

The model presented above is static, *i.e.* it does not take any history into account. For many robotic processes, dynamic models are more appropriate as discussed in Section 2.3. Figure 3.6 shows the extension of the static BN from Figure 3.3 to a DBN.

Figure 3.7 shows results of the filtering algorithm when the robot drives from *Position* 1.0 to 10.0 for three different scenarios: human-only observations, robot-only observations, and human-robot fusion. The figure shows the beliefs and their uncertainties evolving over time. The leftmost plot shows the discrete *Area* distributions with the shading level of the marker proportional to the probability mass. The centre plot shows a ROC curve using the beliefs of the *Area* node. It is shown here to indicate the accuracy of the area classification – the

[4]Logarithms to the base e are used here.

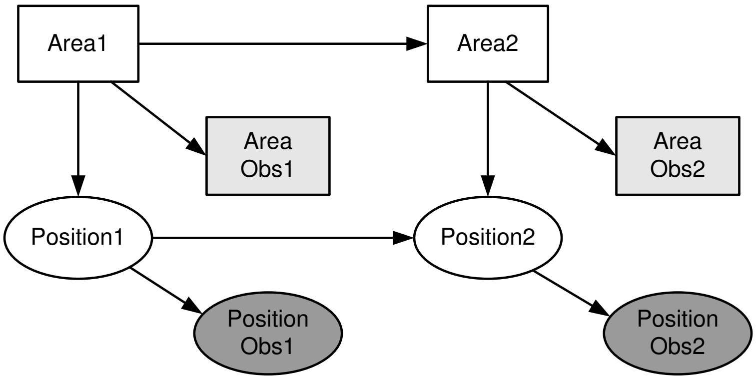


Figure 3.6: Extension of the model shown in Figure 3.3 to a dynamic model.

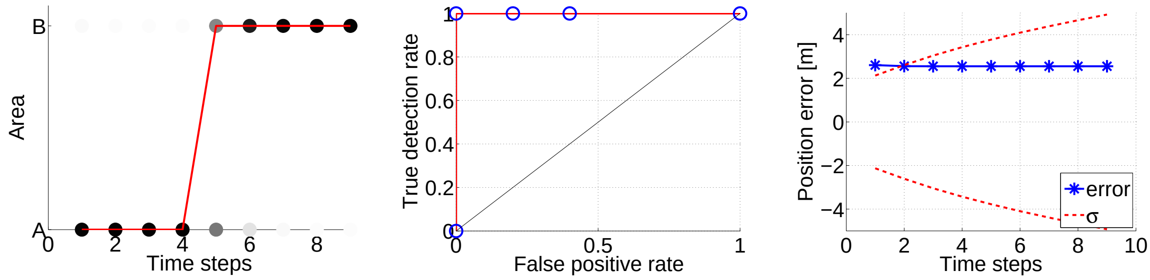
more area under the ROC curve, the more accurate. The rightmost plot shows the robot's *Position* error including the estimated one-σ standard deviation.

The figure shows how the perceptual strengths of humans and robots are combined. In human-only mode, in which the operator only makes discrete *AreaObs* observations, the classification results are excellent. However, the *Position* state can not be estimated well and the uncertainty in the estimate grows over time as shown by the increasing standard deviation. In contrast, the robot by itself tracks the *Position* state well but the estimates of the *Area* state are not as accurate. As for the static model, the best results are achieved when humans and robots fuse their observations.

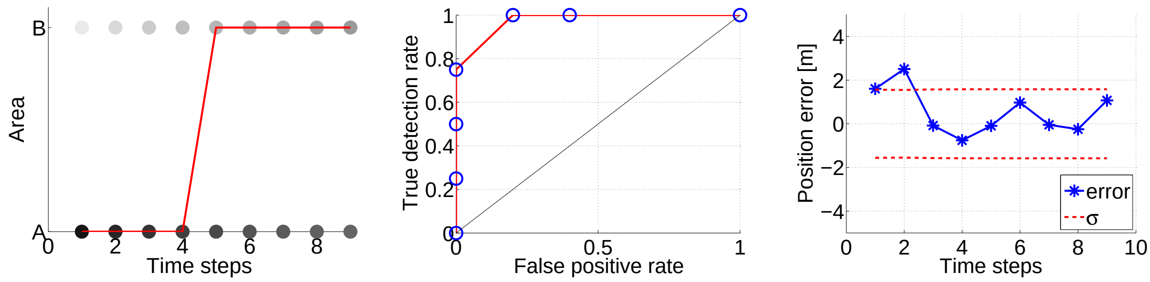
The dynamic example model uses two assumptions related to the time scale: (1) *Area* and *Position* states change at the same rate, and (2) humans and robots add observations at each time step. These assumptions can be avoided:

1. States changing at different rates can be modelled using a semi-Markov model [126] as shown in Figure 3.8. The *Counter* variable represents the duration of the *Area* state: whenever it expires, the *Switch* variable is turned on which lets the *Area* node change states using its transition model. The *Counter* variable then resets according to its model distribution (*e.g.* geometric or encoded as a table).

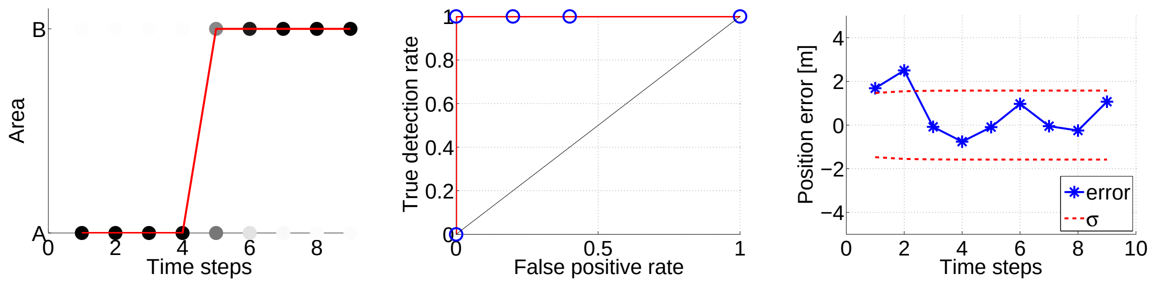
2. Humans are unlikely to submit observations at the same rate as robots. While robotic observations arrive at each time step, human observations are only added occasionally. It is straight-forward to leave out or add observations in a recursive Bayesian filtering algorithm.



(a) Human-only observations



(b) Robot-only observations



(c) Human-robot fusion

Figure 3.7: Simulation results from the dynamic example model. There are three plots per scenario: the plot on the left shows the belief in which *Area* the robot is currently located. The probability mass is proportional to the shading level of the marker and the truth is drawn as a red line. The plot in the centre shows a ROC curve as an indicator of how well the *Area* classification works. The plot on the right shows the *Position* error of the robot including the estimated one-σ standard deviation.

3.4.3 Decision-Making Model

So far, no decision making has been considered. Figure 3.9 shows an Influence Diagram (ID) encoding a simple decision model for an intelligent vehicle. The decision under consideration is whether to drive *fast* or *slow*. In this simple model, it is assumed that the decision only

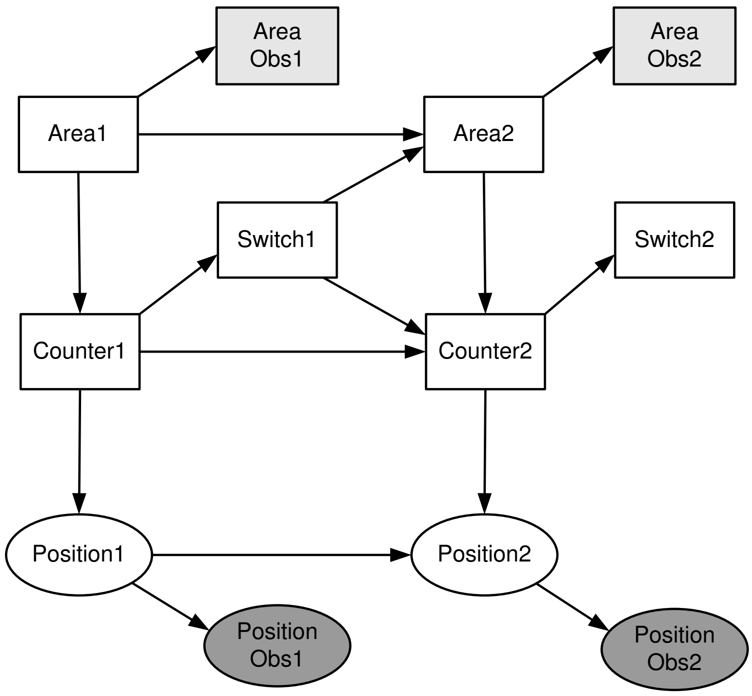


Figure 3.8: Semi-Markov example model for human-robot information fusion.

depends on the *StreetType* (*highway/urban*) which the robot observes with some uncertainty. A higher abstraction level is the *CollisionDanger* encoding the current danger of a collision[5]. This node can be observed by human operators, either an expert or a novice which are represented by a HSM each as shown in Figure 3.10.

The example is aimed at contrasting the approach presented here to systems where human operators take over control as mentioned in Section 3.1.2. Rather than having to "intervene" and directly make the decision which speed to go, human operators make a higher-level observation of a collision danger. The information propagates to the lower level where it is fused with the robot's sensor observation before a decision is made.

It can be argued that observing the collision danger is more suitable to the perceptual capabilities of humans and does not require any technical knowledge. Furthermore, the model may become more complex by adding other variables which influence the speed decision. Observing a subproblem like the collision danger may be more accessible to

[5]Note that the direction of the edge is from *StreetType* to *CollisionDanger* which better reflects the cause-effect relationship between the two variables. In terms of abstraction, however, *CollisionDanger* is higher.

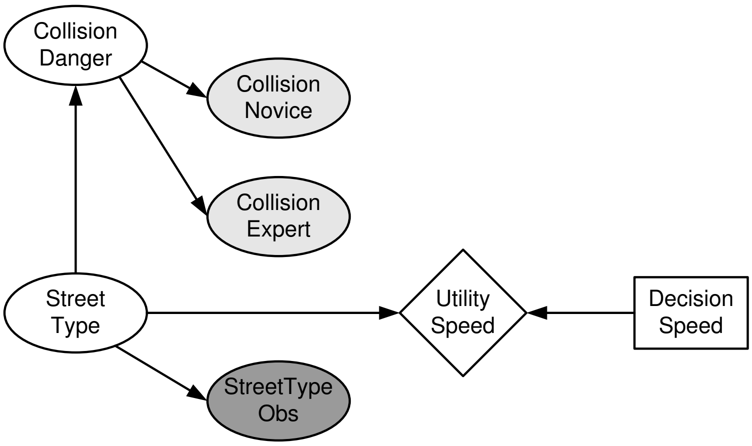
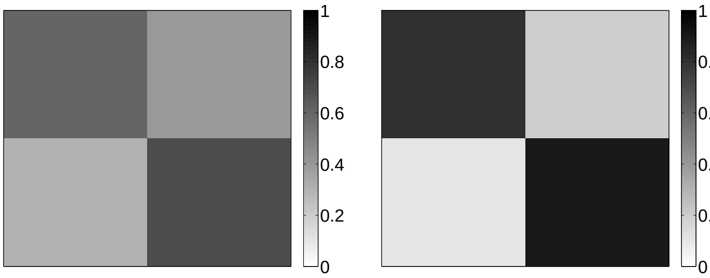


Figure 3.9: Example decision model for human-robot information fusion. The decision is what speed to move which depends on the *StreetType* observed by the robot. In terms of abstraction, a higher-level node is *CollisionDanger* which can be observed by two different types of human operators: novices and experts.



(a) Novice (b) Expert

Figure 3.10: Two discrete HSMs represented as $2x2$ CPTs. The rows are probability distributions and add up to 1.0 while columns represent likelihood functions $p(\overline{collisionNovice}|collisionDanger)$ and $p(\overline{collisionExpert}|collisionDanger)$.

humans than taking all possible variables into account to make a global decision. However, as mentioned in Section 3.3, the approach requires a good model.

Another purpose of the example is to demonstrate the situation of *conflicting evidence* as mentioned in Section 3.3. Table 3.2 and Figure 3.11 show potential conflicts generated by the three available information sources: expert/novice human operators, and the robot. The conflict of evidence is computed using Equation 2.7.

As can be seen from the table and the figure, the lowest conflicts occur when both operators and the sensor "agree" as in the first and last case. The highest conflict occurs when the expert "disagrees" with the novice and the sensor.

Index	Collision-Expert	Collision-Novice	StreetType-Obs	Conflict
1	*low*	*low*	*highway*	−0.87
2	*high*	*low*	*highway*	0.33
3	*low*	*high*	*highway*	−0.11
4	*high*	*high*	*highway*	0.23
5	*low*	*low*	*urban*	0.10
6	*high*	*low*	*urban*	0.09
7	*low*	*high*	*urban*	0.50
8	*high*	*high*	*urban*	−0.21

Table 3.2: Results of the conflict of evidence analysis conducted on the decision model shown in Figure 3.9. Each row represents 3 simultaneous observations by an expert, a novice, and the robot. Positive conflict measures indicate potential conflicts.

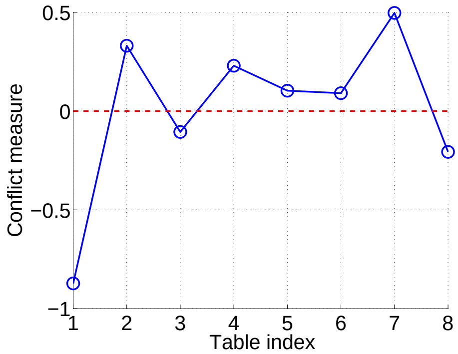


Figure 3.11: Conflict of evidence results from Table 3.2. Positive conflict measures indicate potential conflicts.

3.5 Experiments

This section describes a calibration experiment with the purpose of developing a HSM for range observations. While it was argued earlier that humans are better suited for identifying higher-level properties of an object, operator observations of the object's location are also required under some circumstances. This will be discussed in Section 4.2.1 where the HSM for range is used as part of a larger experimental demonstration.

3.5.1 Procedure and Results

In the calibration experiment, eight poles (length 170cm) were placed in an open space with known coordinates. Twenty-one participants were asked to estimate range and bearing to each pole using direct line-of-sight vision. The experimental setup is shown in Figure 3.12(a).

Figure 3.12(b) visualises the data obtained from the experiment: both single observations of all participants and mean estimates are shown. The ellipses in the graph assume a Gaussian distribution, a two-σ standard deviation is drawn.

The evaluation of the data focuses on the range estimates. Decoupling range and bearing estimates implies the assumption that there is no correlation between them. The results, both qualitatively and quantitatively, can be summarised as follows:

1. The standard deviations of the range estimates can be approximated as a linear function of the true range, *i.e.* uncertainty increases with range which is shown in Figure 3.12(c).

2. The mean range estimates can be approximated as a linear function of the true range. On average, the range was underestimated for all the poles as shown in Figure 3.12(d). The increasing standard deviation is also visualised via error bars.

3. Range estimates can be approximated by a Gaussian distribution which is verified as follows. First, range observations for all poles are pooled into a single standard score z. It is computed by $z = \frac{x-\mu_p}{\sigma_p}$ where x is the raw score to be standardised (a range observation), and μ_p and σ_p are the range mean and standard deviation of the corresponding pole. Second, a histogram and normal probability graph are plotted as shown in Figures 3.12(e) & (f). Since the z-score follows the line of the normal probability graph, the assumption of a Gaussian distribution is reasonable.

Researchers in the field of psychology have also investigated the topic of human distance estimation with similar results to ours. Baird *et al.* demonstrated that the relationship of perceived distance and actual distance, on average, can be described by a power function with an exponent close to 1.0 [9]. Da Silva *et al.* found that it is slightly greater than 1.0 with indoor observations and generally slightly less than 1.0 with outdoor observations [36].

3.5.2 HSM for Range

Based on the experimental data, it is possible to build a probabilistic sensor model[6]. For a HSM, the goal is to produce a likelihood function of x given a human observation z. In the

[6] Another method would be to use the results of the psychologists' research directly.

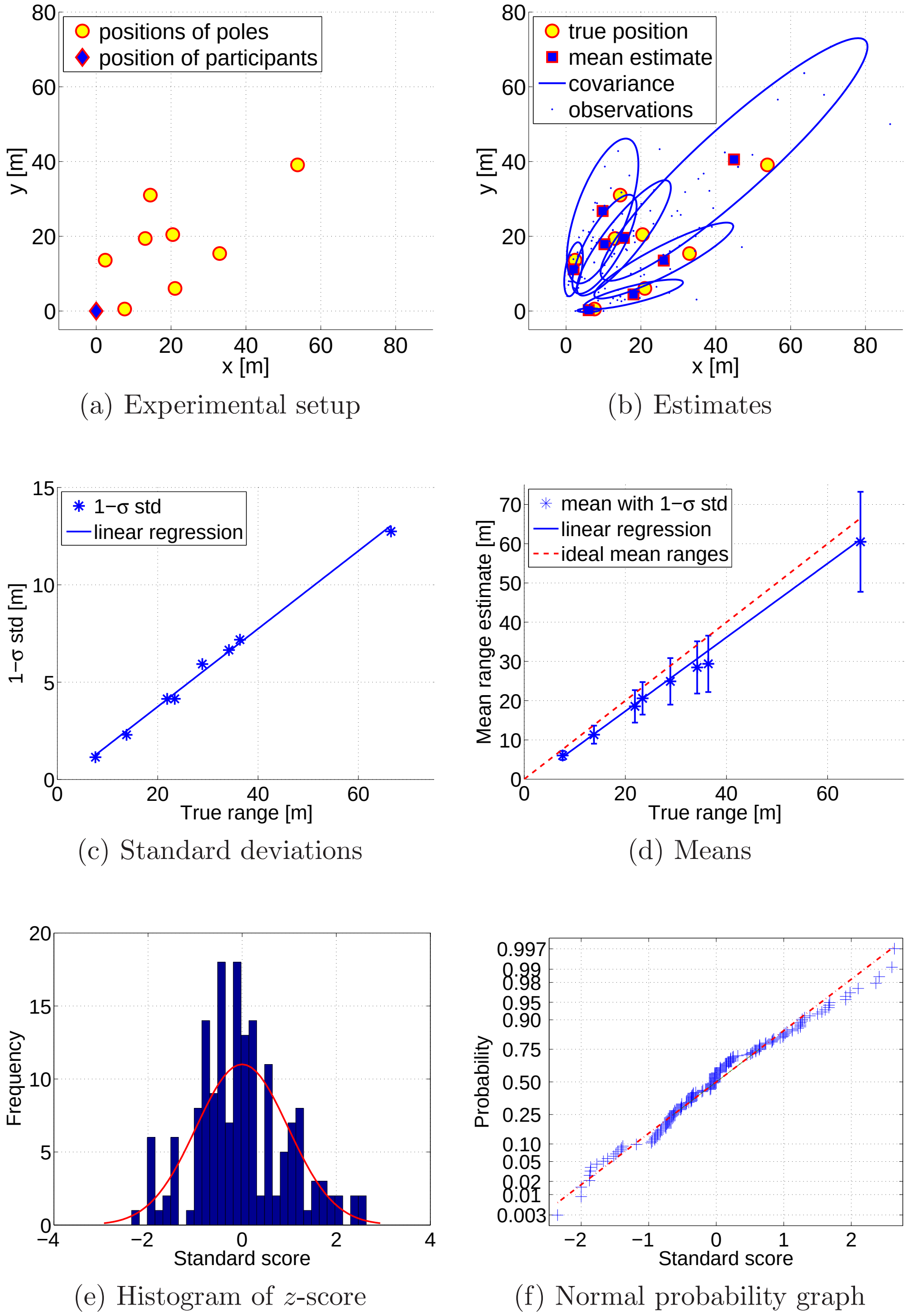


Figure 3.12: Results of the calibration experiment: (a) experimental setup; (b) all participants' range/bearing observations, Gaussian uncertainty ellipses around means, true pole locations are repeated for reference; (c) range standard deviation increases linearly with range; (d) mean range estimates are linear to true ranges, ranges are underestimated, error bars indicate increasing standard deviation; (e) histogram of z-score using 30 bins; (f) normal probability graph verifying the Gaussian assumption for range.

specific range model, both variables[7] are in the form of ranges measured in metres. The HSM can be represented as a simple Bayesian Network (BN) as shown in Figure 3.13.

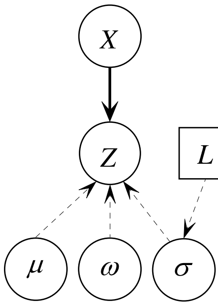


Figure 3.13: A BN representation of the HSM for range explicitly showing the parameters (nodes with outgoing dashed edges). Round/square nodes are continuous/discrete. X is realised as a Gaussian, Z as a conditional Gaussian with parameters μ, ω and σ. The discrete node L encodes the range-dependent uncertainties.

The BN consists of the following nodes: node X represents the true range to the target and is encoded as a Gaussian. Node Z represents the human range observation and is encoded as a conditional Gaussian. Nodes μ, σ and ω are the parameters of the conditional Gaussian sensor model. Node L stands for *Location* and is required to encode the range-dependent uncertainties of the range observations. It is a continuous variable discretised into several bins for implementation purposes. This one-dimensional sensor model can also be formulated as:

$$z = \omega x + \mu + \mathcal{N}(0, \sigma(L)) \tag{3.4}$$

where z is the human range observation, x is the true range, ω and μ are regression coefficient and mean, and σ is the standard deviation of the observation noise which is normally distributed with zero mean. The standard deviation is range-dependent which is expressed by the dependency on the L node.

The model can now be used for inference of range x at run-time as will be shown in Section 4.3.3. The parameters based on the results of the calibration experiment are: $\hat{\mu} = 0$, $\hat{\sigma} = \{1; 2; 3; 4; 5; 6\}$ (dependent on value of L node), and $\hat{\omega} = 0.9$.

[7]Note that x and z should not be confused with their earlier use for raw and the standard scores.

3.5.3 Discussion

The HSM developed above represents an average human operator's ability to estimate the range to an object in an open space. Given the modularity property of BNs, the simple BN shown in 3.12(d) can be integrated into a larger representation as will be demonstrated in Section 4.2.1.

Another advantage of the BN representation is the extensibility to more complex models. In the present form, the model does not address the individuality of human operators as discussed in Section 3.2.2. More variables can be added to the user model, *e.g.* time of the day, current workload *etc.*

A second possibility to address the individuality is to adapt the HSM to individual operators at run-time. This requires the truth of the range being available online. We reported preliminary results of an experiment where such an oracle was available [96].

3.6 Related Work

Related work for this chapter comprises three main topics: humans as information sources, multi-level representations, and probabilistic user modelling. Each one is discussed below.

3.6.1 Humans as Information Sources

Different system functions can be automated when humans interact with automation. Parasuraman *et al.* propose four function classes [136]: (1) information acquisition, (2) information analysis, (3) decision and action selection, and (4) action implementation. Letting human operators contribute perceptual information can be seen as choosing a low automation level for class (1) while classes (2)–(4) are fully automated.

Regarding human operators as information sources is a novel idea and thus, the literature on this topic is limited, especially in the robotics field. Three application areas which may benefit from human perceptual input are crisis management, command-and-control, and cooperative mapping which are reviewed next.

Crisis Management The advantages of fusing information from humans and sensors have recently been recognised in the context of crisis management [139][118][122][110]. Pavlin *et al.* work on integrating human information originating from databases, webpages, mobile phones or interactive querying into their Distributed Perception Network (DPN) [139]. DPNs use a distributed Bayesian network and a representation which can be exploited to generate useful queries addressed at humans in the field. The example provided is a gas detection system where humans contribute information by reporting smell and health symptoms information queried via SMS or the web. The problem of modelling human operators has also been highlighted but no results are reported.

Project Rescue is another crisis management project explicitly acknowledging human operators as sensors [122][6]. First responders' observations and eyewitness accounts may be leveraged to gain a better assessment of the situation using human cognitive abilities. They emphasise the necessity of modelling the reliability of human reports with respect to perceptual and cognitive biases as well as to social background. Probabilistic modelling techniques are proposed as a potential solution. Implementation of the the user modelling ideas to date are limited. The focus is on mapping humans' free text reports containing location information to probability distributions over "domains" which are 2d spatial grids [92]. The system does not perform fusion of human reports with sensor information.

Llinas' crisis management paper is written as a research proposal [110]. It proposes the investigation of fusion algorithms to support decision makers dealing with natural and man-made disasters. Potential information sources include aerial imagery equipment, ground sensors, and humans in the field. Human operators can be either civilians, specifically deployed data collection teams, or rescue staff. This work seems not to have gone beyond this initial proposal.

Command-and-Control In most command-and-control applications, humans remain the final decision maker. Automated systems are typically used to provide aid to decision makers (see *e.g.* [34]).

An exception is work from Ross *et al.* who mention operators as potential information sources in a military global awareness mission [146]. The world is represented as a hierarchical Bayesian Network. Observations can come from image intelligence reports, radar reports, terrain databases or human intelligence reports. They do not address learning the model parameters and entry of human observations is not demonstrated experimentally.

The data fusion community has worked on making human operators an explicit part of the popular JDL data fusion model [15][120]. Blasch presents a simple experiment where users help the fusion system with data association and localisation as part of a group-tracking scenario [15]. Both papers discuss integrating operators on a conceptual level only and do not address the problem of how to represent and fuse information from human operators.

Mapping A human operator and several robots cooperatively build a map of the environment in [25]. It is argued that a common meaningful representation is required for a human-robot team to achieve a common goal. The map contains both lower-level terrain information as well as semantic abstractions provided by a user. A point-and-click mechanism allows the operator to add objects such as victims in a search-and-rescue scenario. The higher-level representation, however, is non-probabilistic and has no relationship to lower-level states.

Another cooperative map building system is presented in [177]. A human operator is explicitly acknowledged for *Human Augmented Mapping*. Operator input is used to make the map human-understandable and to clarify ambiguities using dialogs [103]. A related dialog-based approach is presented in [161] where operators label objects which the robot has segmented previously. Both of these approaches do not fuse information, *i.e.* robotic and human map states are not correlated.

3.6.2 Multi-Level Representations

Figure 1.3 showed the environment as consisting of several levels of abstraction including features, objects, tracks, situation and impact. These terms have been adopted from the standard multi-level process model used in the information fusion community, the JDL Data Fusion Model [187]. It is a functional model motivated by the lack of a common terminology when discussing different elements of information fusion processes.

The JDL model consists of multiple levels but does not prescribe any fixed architecture or choice of representation. In principle, all levels of the JDL model can be represented using a probabilistic framework. BNs are considered suitable because they keep track of inter-level correlations which enables inter-level information flow. A BN implementation of JDL level 2 for a battlefield application is presented in [66]. The objectives are similar to ours: both higher-level nodes such as enemy intent and lower-level nodes such as enemy tracks are represented together. The potential of human input from friendly units is also mentioned but no quantitative results are presented.

Multi-level *probabilistic* models have successfully been applied to a vast number of applications in need of representing information on multiple levels of abstractions. Domains include computer vision [168][127][178][184], speech and text modelling [128][41][144], and activity recognition [109][138][132][180][62]. Multi-level probabilistic models are also useful if decisions need to be made. Examples include intelligent light control [186], and robot localisation [172].

None of these models consider input of evidence on a higher level. Typically, a single low-level sensor stream is used to infer higher-level states. Sensors include laser [172], GPS [109][138], camera [168][127][178][184], microphone [128], and photosensor [186]. Alternatively, multiple low-level sensor streams are fused, again to infer higher-level states. Examples are video and audio [128][41] and audio/video/computer activity [132]. In contrast, this work lets human operators add information on higher-level states which propagates throughout all other levels depending on the correlations encoded in the model.

Since the propagation of information to lower levels is not anticipated in the models cited above, one approach is to isolate levels from each other. The advantage is the decomposition of the modelling problem which simplifies learning and inference for submodels. Oliver *et al.* use this *layered* approach to recognise activities in an office environment [132]. Each layer is represented as a set of independent HMMs which propagate their results to higher levels. A similar approach is used for recognising potential terrorist attacks [180]. The output of lower-level HMMs provides evidence for a high-level BN which integrates all information into a global belief. In contrast to these approaches, this work requires to maintain the correlations between levels to enable inter-level information flow.

3.6.3 Probabilistic User Modelling

User modelling is an important topic in Human-Computer Interaction (HCI), Artificial Intelligence (AI), and more recently, in Human-Robot Interaction (HRI). Work in these fields, which makes use of probabilistic methods, is briefly reviewed next.

When people interact with computers, the objective is often to infer the user's goals or intentions [79]. The purpose is to provide assistance to users who try to solve a task, or adapt user interfaces to the context [84]. Adaptive user interfaces are also used in HRI, *e.g.* to control a robot using voice [121]. Another application area where users' mental states are inferred is Intelligent Tutoring Systems (ITS). These systems typically reason about the student's comprehension of certain concepts the ITS intends to teach [123][28]. In contrast to these applications, the systems presented in this thesis do not necessarily require a complex model of the users' mental states. Operators make simple observations in our systems and the user model's purpose is to add uncertainty to it. In plain terms, the user model encodes how much a human observation can be "trusted".

The major challenge in user modelling is how to address the *individuality* of humans, *i.e.* how to learn something about the individual [84]. This requires a significant amount of data from individual users which can easily be collected for HCI applications. In the robotics domain, it is harder to obtain sufficient data, and our solution so far has been to use stereotypical user profiles (novice/expert). Another option is to use life-long or incremental learning techniques [78] which will be investigated in the future.

Besides the variation *between* users, the individual's performance is also variable and depends on factors such as time of the day, workload, and attention. All these states can be incorporated into a HSM but require some type of sensing of the operator, either actively [167], or passively [73].

3.7 Summary

This chapter introduced the concept of treating human operators as information sources. First, examples were given of how different probabilistic data types can be used for bidirectional human-robot communication. Then, the information flow direction was restricted to human-to-robot suggesting the exploitation of humans' perceptual capabilities. It was proposed to let humans submit *information* which was defined as part of Bayesian inference to be either *likelihoods* or *observations*. It was shown how treating humans as information sources can be applied to both information gathering and decision making. For the latter case, the advantages of using information over *actions* for robot control were also discussed.

It was argued that humans and robots are inherently different in their perceptual capabilities. This lead to certain challenges for the probabilistic representation employed for human-robot information fusion. The conclusion was that BNs are suitable to address the identified challenges. BNs are also able to incorporate a probabilistic model of the operators' perception which is called Human Sensor Model (HSM) in this work. Then, limitations of the approach were identified. A set of simulations were used to verify the arguments for human-robot information fusion, and to demonstrate some of the limitations. Finally, an experiment with human participants was presented which yielded a HSM for range observations.

This chapter provides an important foundation for the two following experimental chapters. Chapter 4 will present experimental results from an information gathering task with human operators submitting environment information (human-push). The HSM for range presented in Section 3.5 will be used in that experiment. Chapter 5 will apply human-robot information fusion to robotic decision making with operators being queried for their perceptual input (robot-pull).

Chapter 4

Scalable Human-Robot Information Gathering

This chapter uses the framework of probabilistic human-robot communication established in the previous chapter to analyse a particular class of applications. The focus of this chapter is on issues arising in systems consisting of multiple humans and multiple robots. Humans and robots are organised into a team tasked with information gathering. The objective is to fuse observations from robotic platforms and human operators into a representation common to all members of the team. A desired property for human-robot teams is the extensibility to an arbitrary number of team members without creating a computational or communication bottleneck. This property is referred to as *scalability* which cannot be jeopardised in any way, *e.g.* by the type of information exchanged between humans and robots. It is argued that only *environment* information is a suitable data type for scalable human-robot communication. Figure 4.1 illustrates the scope of this chapter.

Two main contributions are made in this chapter: (1) demonstration of scalable human-robot communication, and (2) presentation of a shared environment representation for human-robot information fusion. To date, limited work has been done on systematically investigating the scalability of human-robot interactions [171]. Often, the focus is on a single operator actively controlling a small number of robots [32]. In contrast, the approach presented in this chapter assumes fully autonomous control of robotic platforms based on the shared environment representation built up with human cooperation. In related work dealing with shared human-robot representations, the operator does not contribute any

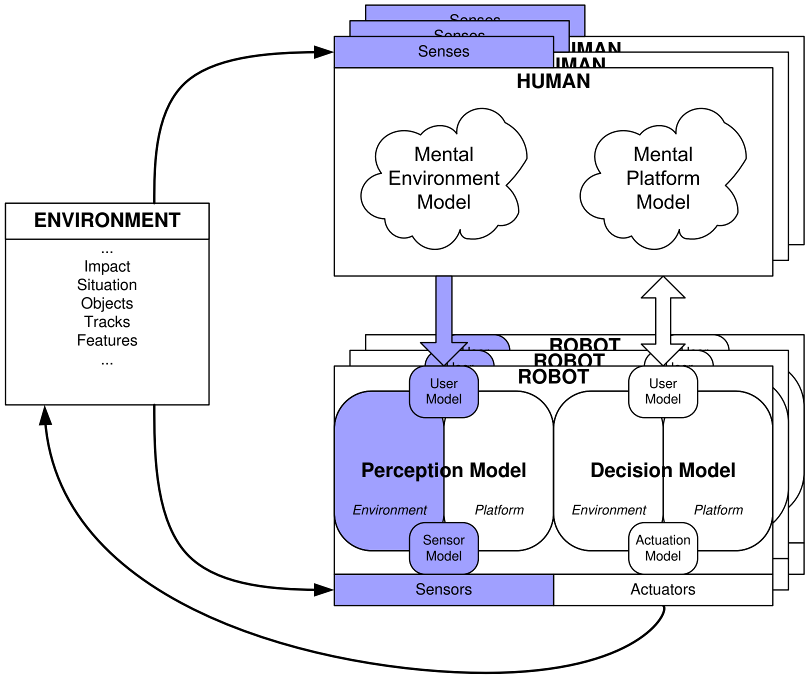


Figure 4.1: This chapter presents human-robot information gathering which is constrained by the requirement of scalability (many-to-many interactions). Only environment information is fused and no decisions are made.

information to the task online [183], or human input is decoupled from robotic information [25], *i.e.* no information fusion occurs. A detailed literature review is presented at the end of the chapter.

The remainder of the chapter is organised as follows. Section 4.1 presents the algorithms used to maintain scalability by decentralising environment state estimation. It also discusses the role of human operators as part of a decentralised system. Section 4.2 presents the shared environment representation and demonstrates the integration of human operators as information sources. In this chapter, the human-push pattern is employed for human information contribution. Section 4.3 presents experimental results obtained by deploying a human-robot team in a natural outdoor environment. The team consists of an unmanned aerial vehicle (UAV), a ground vehicle, and two human operators as shown in Figure 4.2. Fusion results are presented as a cooperatively acquired multi-attribute map, and a set of human-robot information exchange patterns. Related work is presented in Section 4.4 before Section 4.5 summarises the content of this chapter.

Figure 4.2: Human-robot team deployed for information gathering: the Brumby UAV, a ground vehicle, and two human operators.

4.1 Scalable Information Fusion

This section starts by introducing our decentralised approach to information fusion. System properties relevant for the integration of human operators are described here. Second, reasons are provided for regarding human operators as an integral part of decentralised information fusion. Then, the roles operators can play in a system constrained by decentralisation are described.

4.1.1 Decentralised Data Fusion (DDF)

We consider applications where the objective of the human-robot team is to build a common environment representation. The team is broken up into *components* including embodied components (humans and robots) as well as software components. Components can be organised using different network topologies, *e.g.* hierarchical or centrally distributed [76][129]. The usage of a decentralised architecture has several advantages over other solutions [116]:

- *Scalability*: the network can grow to an arbitrary number of components (humans, robots, software components).
- *Survivability*: no component of the system is mission-critical, so the system is survivable in the event of run-time loss of components.
- *Modularity*: all components can be implemented and deployed independently.

These advantages require that (1) no central services, facilities or components exist, and (2) no global knowledge of the network topology is needed. We use the family of *Decentralised*

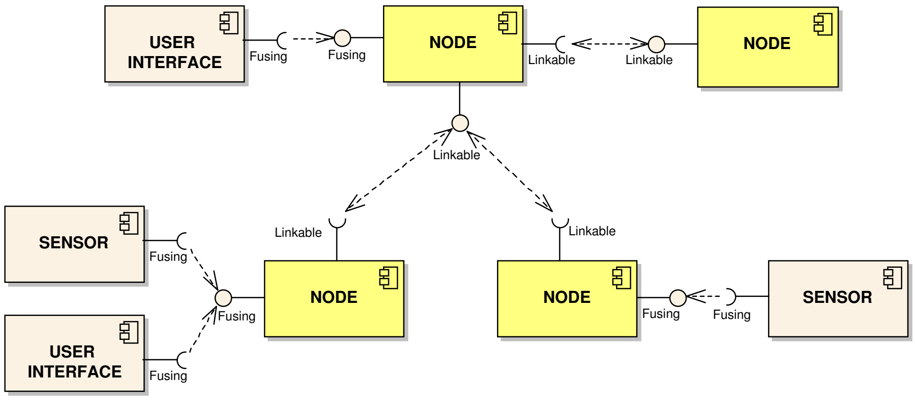


Figure 4.3: A decentralised data fusion system: SENSORs and USER INTERFACEs submit likelihoods to NODEs which form a DDF network (highlighted). Arrows indicate the direction of data flow.

Data Fusion (DDF) algorithms which fulfil these requirements [69]. Scalability of the DDF algorithms has been demonstrated before [69][115] and is not the focus of this work.

An example of a decentralised system represented as a UML component diagram is shown in Figure 4.3. Each software component (capitalised throughout the rest of the chapter) has a set of provided and/or required interfaces visualised as filled circles and open semicircles respectively. Interfaces are used to communicate data between components.

The example shows a DDF network consisting of four NODEs. Each NODE runs the DDF algorithm whose task is to maintain a consistent probabilistic estimate of the environment state using a Bayesian filter as introduced in Section 2.3.2.

In a DDF system, likelihoods are produced by SENSORs and USER INTERFACEs respectively, and communicated to NODEs via the `Fusing` interface as shown in Figure 4.3. To ensure that the estimate is common to all NODEs they have to communicate with each other via the `Linkable` interface. This involves keeping track of previously communicated information to avoid *rumour propagation* [10]. Based on the locality principle which guarantees scalability, every NODE only exchanges information with its immediate neighbours. After communication, all new information propagates throughout the entire DDF network resulting in a common belief of the environment.

Objective	Interface	Comms. patt.	Message	Prob. data type
Human-robot fusion	***Fusing***	***human-push***	***env.-perc.***	***likelihood***
Task monitoring	Informed	robot-push	env.-perc.	belief
Platform control	Controllable	human-push	plat.-dec.	action, utility fct., policy
Status monitoring	Detailed	robot-push	plat.-perc.	belief

Table 4.1: Overview of objectives of human interactions with a DDF network and their realisations as interfaces (compare to Table 3.1). The focus of this work is emphasised.

4.1.2 Roles of Operators in DDF

In a decentralised system, the numeric relationship of humans and robots is potentially *many-to-many* (see Section 1.3.3). Together with the other requirements of decentralisation, this imposes constraints on the interaction of human operators with the other team members. In general, there are two types of information which can be queried or submitted by operators: information related to the *environment* and information related to the *platforms* as described in Section 1.3.2. An example of environment information is the location of a feature. An example of platform information is the health status of a robot.

Only environment information is scalable with respect to the size of the network, *i.e.* querying or submitting environment information is independent of how many platforms are deployed. In DDF, only environment information is fused and shared among the platforms.

To ensure scalability, the main objectives of human-robot interaction have to be related to environment information, namely: (1) to present the user with the global world state (task monitoring); and (2) to allow human operators to contribute environment information to the DDF network (human-robot fusion). Human-robot fusion can be realised by two communication patterns as described in Section 1.3.2: robot-pull and human-push. For the experiments in this chapter, human-push was used exclusively.

Other human-robot interactions which refer to platform information are non-scalable but can be useful in a practical system. A summary of all objectives is given in Table 4.1 and their realisation as a USER INTERFACE is shown in Figure 4.4. Each objective is discussed in more detail below.

Human-Robot Fusion The objective of letting operators contribute information can be achieved by using peer-to-peer interaction (see Section 1.3.3). Using the human-push communication pattern, the network is at no time aware of where the information comes

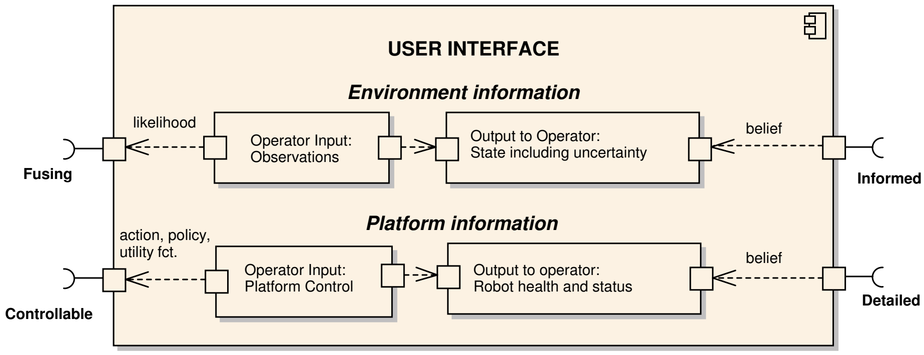


Figure 4.4: A USER INTERFACE with four interfaces and corresponding data types is used to fulfil the requirements of human-robot interaction as part of a DDF system.

from, a physical sensor, or the human operator. This is reflected in Figure 4.3 where both the USER INTERFACE and the SENSOR use the same `Fusing` interface. Operators can either submit raw observations which get translated into a likelihood by a human sensor model, or specify likelihoods directly as discussed in Chapter 3.

Task Monitoring The objective of task monitoring is realised by presenting the world state to the user which is achieved by connecting to the `Informed` interface. In this case, the operator represents an information consumer and simply needs to connect to the most easily accessible NODE. Apart from an information propagation delay, any NODE in the DDF network provides all environment information as a probabilistic belief as described in Section 4.1.1. The type of display can be graphical, textual, or another modality appropriate for the type of information gathered.

Platform Control In a practical system, it is often convenient to have direct control of the active platforms using teleoperation. Another option is to let operators send utility functions or policies which is a supervisory interaction mode [115]. Information related to platform control are submitted through the `Controllable` interface. This *component-centric* control scheme does not scale well with the number of components and thus, if incorporated in a DDF network, the number of platforms controlled at any one time must be limited [115].

Status Monitoring The human-robot team represents a distributed system which has a high degree of inherent complexity. It is therefore useful to directly monitor the status of the robots at run-time. This includes the robots' positions in the world, health status, and the state of individual software components at runtime. Status information is aggregated through the `Detailed` interface which is realised by every software component. If collected centrally, the individual components' status information can be compiled to display the global DDF network topology which is useful for debugging purposes. The display of status information can be switched off to avoid a communication or computational bottleneck.

4.1.3 Avoiding Human Rumour Propagation

The previous section argued that there is no restriction on the number of operators simultaneously connected to the DDF network as long as operators deal with environment information only (scalability). This section discusses the issue of preserving a second important property of DDF – the correctness of the global estimate enforced by avoiding rumour propagation as mentioned in Section 4.1.1.

Rumour propagation between NODEs can be avoided by either adhering to a tree network topology or by fusing conservatively[1] [114]. These mechanisms are sufficient as long as information flow from SENSORs and USER INTERFACEs is one-directional as shown in Figure 4.3. However, when human operators are involved, the following information loops may emerge:

1. Human operators may act as environment information sinks for task monitoring purposes as discussed in Section 4.1.2. Receiving beliefs from the DDF network most likely influences the observations operators make. As a consequence, operators should only be allowed to exclusively act as an environment information source or sink.
2. Human operators may form information loops with each other when working in close proximity. Examples include direct conversations between teammates and the observation of each others' actions.
3. Human operators most definitely form a temporal information loop with themselves, *i.e.* they remember their past observations. This loop can be avoided by prohibiting the reobservation of the same feature.

The three possible information loops need to be ruled out to maintain a correct probabilistic belief of the environment state. However, these rules may prove hard to implement and

[1] In this work, a tree network is used.

enforce in practice. One option is to employ dedicated human NODEs which fuse their information conservatively with the rest of the DDF network, *e.g.* by using covariance intersect methods [90]. A second option is ask human operators to only submit *new* information. Although people appear to be capable of performing this type of operation [140], it would most likely be unreliable and increase the operators' workload significantly.

For the experiments reported in Section 4.3, none of the three rules were explicitly enforced, no dedicated human NODEs were used, and operators were not asked to submit new information only. These measures were not necessary because results from human-robot information exchange were analysed qualitatively only.

4.2 A Shared Environment Representation

This section presents the probabilistic representation of the environment used to perform information fusion. Two independent models are used to describe features in a natural outdoor environment: geometric and visual. The visual model is separated into an appearance and identity level which are statistically correlated. The geometric and visual models are combined to achieve robust data association.

In short, environmental features are represented on three abstraction levels: *geometric, appearance* and *identity.* The representation is shared between all platforms which is required by the DDF algorithms. The shared representation allows information contribution from both robotic sensors and human operators.

4.2.1 Geometric Feature Representation

The most common low-level representation of a point feature is its position in a global coordinate system. Each NODE maintains the geometric state of all features using a probabilistic filter.

Robotic Observation Model for Position In our implementation, robotic platforms observe the bearing to extracted features using cameras. Gaussian Mixture Models (GMMs) are adopted as a representation for the geometric state because they provide a basis for analytical solutions to the general Bayesian filtering problem [164]. Furthermore, this representation can also be used for spatially extended features even though this work only considers point features. A single Gaussian would be sufficient to represent these, however, a GMM or an approximation of it is still required for undelayed point feature initialisation [162]. To guarantee the applicability of the derived DDF algorithms for other applications, the entire GMM representation is maintained for all time steps.

A GMM is defined for a random variable X as

$$p(\mathbf{x}) = \sum_{i=1}^{N} \pi_i \mathcal{N}(\mathbf{x}|\mu_i, \Sigma_i) \tag{4.1}$$

where π_i are weights with the property $\sum_{i=1}^{N} \pi_i = 1$, $\mathcal{N}(\mathbf{x}|\mu_i, \Sigma_i)$ is a Gaussian probability density (also known as a Gaussian mixture component) with mean μ_i and (full) covariance Σ_i, and N is the number of mixture components.

Substitution of GMMs into Bayes Theorem (Equation 2.8) gives

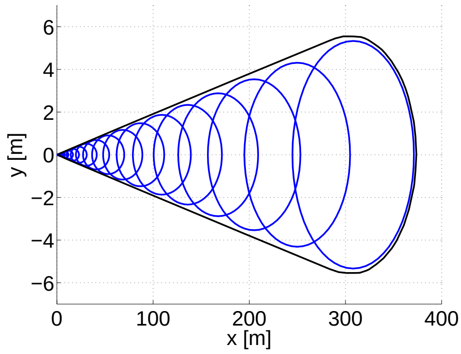
$$p(\mathbf{x}_{k+1}|\mathbf{z}_{0:k+1}) = \alpha \sum_{i=1}^{M} \pi_{zi}\mathcal{N}_{zi} \sum_{j=1}^{N} \pi_{xj}\mathcal{N}_{xj} \tag{4.2}$$

where $\alpha = 1/p(\overline{\mathbf{z}}_{k+1}|\mathbf{z}_{0:k})$ is a normalising constant, the $\mathcal{N}_z$'s represent the likelihood $p(\overline{\mathbf{z}}_{k+1}|\mathbf{x}_{k+1})$ (M mixture components), and the $\mathcal{N}_x$'s represent the prediction $p(\mathbf{x}_{k+1}|\mathbf{z}_{0:k})$.

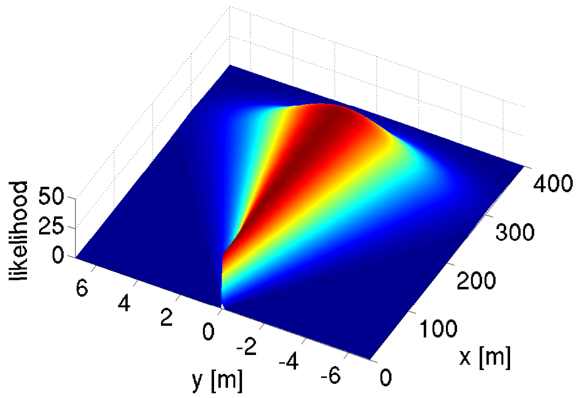
Expanding Equation 4.2 results in $M \times N$ terms, each involving a multiplication of two weighted Gaussians. Thus, the posterior distribution is represented by $M \times N$ weighted Gaussians. GMMs also allow an analytical solution to the Chapman-Kolmogorov equation (Equation 2.9). The multiplicative expansion in parameters requires a component merging technique. Details of the technique and inter-NODE fusion issues are described in [181].

A 2d example of a robotic observation translated into a GMM likelihood is shown in Figure 4.5. A bearing-only observation from position $[0, 0]$ with bearing $0°$ is translated into a GMM with 20 Gaussian components [8]. In our experiments presented in Section 4.3, position is represented by 3-dimensional GMMs (easting, northing, altitude).

Human Observation Model for Position Human operators also observe the position of features even though observation accuracy is lower than that of robotic sensors as shown in Section 3.5. Typically, human operators are considered more valuable in entering higher-



(a) 2d GMM likelihood



(b) 3d GMM likelihood

Figure 4.5: A 20-component GMM likelihood generated via a bearing-only observation of $0°$ from position $[0, 0]$. Individual Gaussian components are shown in (a), along with an equi-likelihood contour encompassing 95% of the probability mass shown in (b).

level information such as the features' class. As a prerequisite, operators also have to enter approximate feature locations to (1) instantiate new features, and (2) distinguish features which are already in the world model (data association).

To enable the integration of human position entry into the filtering scheme presented above, human observations have to yield likelihoods in a GMM-compatible form as formulated in Equation 4.2.

Two input modes for human entry of information were developed as presented in Section 3.2.4. Both were implemented to output single-Gaussian likelihoods and were used simultaneously in our experiments presented in Section 4.3. Their implementation for the deployed human-robot team is described next.

In the likelihood mode, humans operate directly in *state space*. This is implemented by drawing circles on the screen of a GUI which are mapped to a Gaussian likelihood with a 3σ uncertainty in three spatial dimensions (the mean for the elevation is looked up from a digital elevation map).

In raw observation mode, humans operate in *observation space* by specifying a range estimate to a feature relative to their position and an absolute bearing value that is read off a compass. The operator's position is acquired from a handheld GPS receiver. Uncertainty values are added to GPS location according to the GPS's current uncertainty estimate. The bearing value uncertainty is fixed according to the specifications of the compass. The experimentally acquired Human Sensor Model (HSM) presented in Section 3.5 converts the range estimates to a Gaussian likelihood.

4.2.2 Visual Feature Representation

The main objective of building a probabilistic visual feature representation is to improve data association. The model should also allow the classification of features and have the capability of incorporating both robotic and human observations. Another model requirement is the allowance for efficient communication and fusion within the DDF network.

The representation is formulated analytically as part of a Bayesian filtering framework. A filtering approach estimates visual states by keeping a history (see Section 2.3.2) which is likely to improve data association and classification compared to a memoryless method. An analytical derivation of a visual likelihood function avoids online data compression which can be computationally expensive. The likelihood maps a high-dimensional observation into a compressed functional description. As a consequence, computations can be conducted in real-time. A compact representation is also required for communication in a DDF system as bandwidth is often limited, especially when using a broadcast medium [134].

The visual model proposed here is learnt offline from training image patches. Four steps are involved which are described in detail below:

1. Deterministic nonlinear dimensionality reduction of image patches and partial labelling of the patches.
2. Learning a static probabilistic model over the original high-dimensional and resultant low-dimensional spaces.
3. Adding a human visual sensor model.
4. Extension to a dynamic model suitable for filtering.

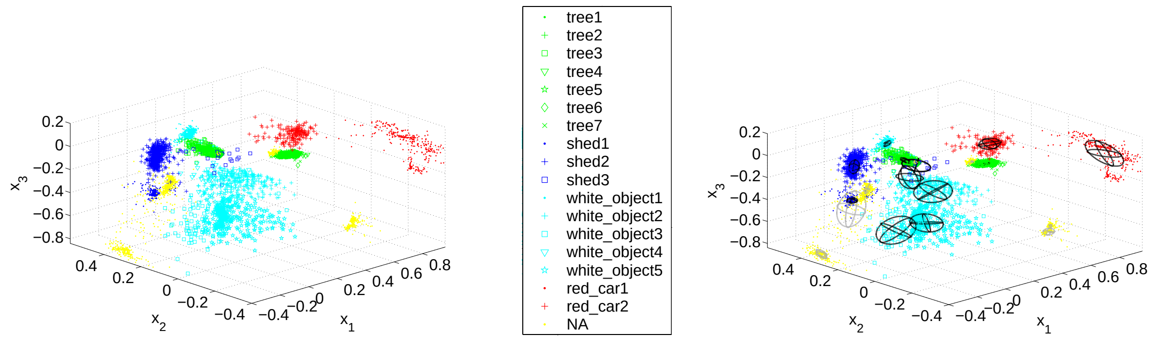
Nonlinear Dimensionality Reduction Most raw visual features exist in a very high-dimensional space that are not readily amenable to simple interpretation and reasoning tasks and therefore require some type of dimensionality reduction. These methods compress the information while keeping most of its content. Note that the objective is different from *classification* which focuses on *discrimination* or *class separability.* To obtain an accurate representation, greatest emphasis is usually placed on features with greatest variability which are not necessarily well separated [46].

The following visual model is independent of any specific feature extraction algorithm. In our experiments the features used were 11×11 patches from colour images. Each of these image patches was represented by a 3D RGB histogram with 9^3 bins resulting in a dimensionality of 729.

Dimensionality reduction of features such as the aforementioned image patches is traditionally performed using methods such as Principal Component Analysis (PCA) or its numerous variants [46]. Although they provide theoretically optimal representations from a data-compression standpoint, they are unable to provide neighbourhood preserving representations that are crucial to data association. This limitation has motivated the development of various nonlinear embedding methodologies such as Kernel PCA [151], Isomap [170], Laplacian Eigenmaps [13] and Locally Linear Embedding [147]. Most non-linear dimensionality reduction techniques presume that the data lies on or in the vicinity of a low-dimensional manifold and attempt to map the high-dimensional data into a single low-dimensional, global coordinate system. The Isomap algorithm is adopted in this work because it can estimate the manifold's intrinsic dimensionality. A quantitative comparison of Isomap to PCA is presented in Appendix A.

Figure 4.6(a) shows the low-dimensional data points generated by Isomap for a data set collected outdoors by ground and air vehicles. Each data point shown in the figure was found by mapping a 729-dimensional image patch to a 3-dimensional space. Using 3 dimensions seems appropriate to represent the data as suggested by Isomap's residual variance plot shown in Figure A.1(a).

Static Probabilistic Model Integration into a Bayesian filtering framework requires the definition of the observation model $p(\mathbf{z}|\mathbf{x})$, describing the measurement uncertainty of a state $\mathbf{x}$, given observations $\mathbf{z}$. For the visual model, $\mathbf{z}$ represent visual observations in the original high-dimensional space which measure abstract states $\mathbf{x}$ in low-dimensional space computed by Isomap. However, the Isomap algorithm and indeed most nonlinear dimensionality reduction algorithms are inherently deterministic algorithms.



(a) State space samples (b) Probabilistic model

Figure 4.6: The low-dimensional visual state space as generated by Isomap: (a) each data point is a deterministic mapping of a high-dimensional image patch into state space; (b) a probabilistic model is learned. Ellipsoids are Gaussians representing clusters of data points whose corresponding image patches have similar appearance with respect to colour. Each ellipsoid corresponds to an identity state whereas a subset of all ellipsoids correspond to a feature class such as "tree". Black/grey ellipsoids represent labelled/unlabelled data points.

Uncertainty can be incorporated by learning a probabilistic model of the joint distribution $p(\mathbf{z}, \mathbf{x})$, given sample sets $\langle \mathbf{x}, \mathbf{z} \rangle$, where $\mathbf{x}$ has been computed by Isomap. Learning a joint probabilistic model over spaces with different dimensionality has previously been shown by Ghahramani *et al.* [61]. The authors propose a model that probabilistically clusters data in the high and low-dimensional spaces simultaneously. This model is known as a *Mixture of Factor Analysers* (MFA) and provides the ability to perform dimensionality reduction while also obtaining a measure of uncertainty in both spaces. The low-dimensional part of this statistical representation conveniently represents highly nonlinear manifolds such as the ones generated by Isomap through the capability to model the local covariance structure of the data in different areas of the manifold.

In this work, a modification of the MFA model is used [141]. Its representation as a Bayesian Network (BN) is displayed in Figure 4.7(a). Round nodes are continuous random variables, square nodes are discrete, and shaded nodes are observed (at run-time). The BN encodes the following joint probability distribution:

$$p(\mathbf{z}, \mathbf{x}, s) = p(\mathbf{z}|\mathbf{x}, s)p(\mathbf{x}|s)p(s) \tag{4.3}$$

The conditional probability distributions are given by:

$$p(\mathbf{z}|\mathbf{x}, s) = \frac{1}{(2\pi)^{\frac{D}{2}}|\Psi_s|^{\frac{1}{2}}} \exp\{-\frac{1}{2}[\mathbf{z} - \Lambda_s\mathbf{x} - \mu_s]^T\Psi_s^{-1}[\mathbf{z} - \Lambda_s\mathbf{x} - \mu_s]\} \quad (4.4)$$

$$p(\mathbf{x}|s) = \frac{1}{(2\pi)^{\frac{d}{2}}|\Sigma_s|^{\frac{1}{2}}} \exp\{-\frac{1}{2}[\mathbf{x} - \nu_s]^T\Sigma_s^{-1}[\mathbf{x} - \nu_s]\} \quad (4.5)$$

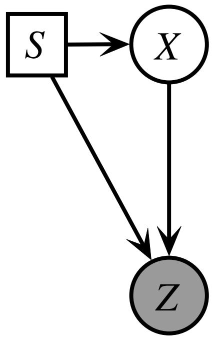
where the terms $\Psi_s, \mu_s, \Lambda_s, \Sigma_s, \nu_s, p(s)$ are the parameters to be learnt, $\Lambda_s\nu_s + \mu_s$ and $\Psi_s + \Lambda_s\Sigma_s^T\Lambda_s^T$ are the means and covariances respectively of the mixture describing the high-dimensional space, ν_s and Σ_s are their counterparts in the low-dimensional space, and Λ_s are known as the regression matrices and model the transformation between the two spaces.

The mixture simultaneously models the data in high and low-dimensional space. The S node is multinomial and determines the number and the weight of the components which are of Gaussian form.

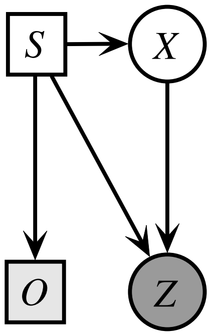
The purpose of dimensionality reduction was to generate a fully supervised data set $\langle\mathbf{x}, \mathbf{z}\rangle$. The data set is extended by manually labelling a subset of the image patches which makes the S node partially observed during the learning phase. Labelling by a human is required to build a human visual sensor model as described in the next section. The S node's dimension N_s is allowed to be larger than the number of specified labels because more clusters may exist in the data than labels are given. The S node is a categorical variable and can be thought of as describing the feature class.

Clusters of unlabelled data points are captured automatically by applying an Expectation Maximisation (EM) learning scheme [141]. If no label for s exists, the standard 2-stage EM algorithm is applied. If a label is available for s, the Expectation step is omitted and only the Maximisation step is executed.

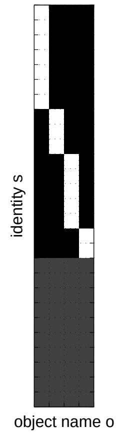
Figure 4.6(b) shows the learned model which was used for the experimental demonstration. Each 729-dimensional image patch in $\mathbf{z}$ space is reduced to a 3-dimensional representation in $\mathbf{x}$ space as shown in Figure 4.6(a). The probabilistic model is visualised as ellipsoids representing the mixture's components with associated weights contained in s. Each Gaussian component represents a cluster containing data points whose corresponding image patches have similar appearance with respect to colour. Different lighting conditions and camera



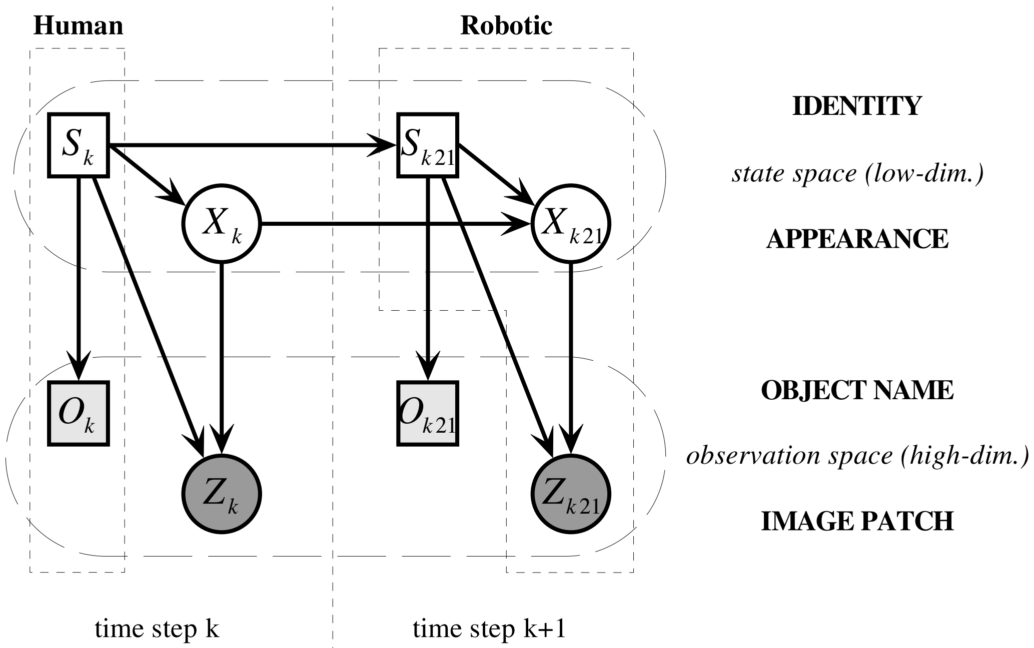
(a) Static model



(b) Human observation node



(c) HSM $p(o|s)$



(d) Dynamic multi-level human-robot representation

Figure 4.7: The visual model as described in the text represented as a BN: (a) the static model: X and Z are realised as low and high-dimensional Gaussians whereas S is realised as a multinomial representing the weights of the Gaussians. (b) Node O is added representing the (human) observation of the feature class. (c) The O node's parameters (human visual sensor model) encoded as a table. Brighter fields symbolise higher values with rows adding up to 1.0. The 4 object names are "tree", "shed", "white_object" and "red_car". The 27 identities correspond to the number of Gaussian components (ellipsoids) of Figure 4.6(b). (d) The filtering process represented as a DBN. Nodes O and Z represent observations by human operators and robotic cameras.

hardware yield different clusters for the same feature class (*e.g.* 7 different clusters for the tree class). It can be recognised that clusters from different classes are well separated.

The dimension of s is chosen to be 27, *i.e.* 27 components are allowed which avoids over-fitting. Only 17 labels are manually assigned which results in 17 clusters represented by black ellipsoids. The remaining unlabelled data points (all yellow) are represented by 10 grey ellipsoids. The number of samples used is 12388 (8428 labelled, 3960 unlabelled). The learning algorithm was implemented in Matlab and typically took 2.5 hours on a 3.2 GHz Intel Xeon machine. EM learning was discontinued when the improvement in the log-likelihood became less than 0.1 or after 10 iterations, whichever came first.

The advantages of this flexible learning scheme are two-fold: (1) it only requires partial labelling, and (2) the number of components in the mixture is not limited to the number of labels.

Human Visual Sensor Model To integrate visual observations from human operators into the probabilistic framework, a human visual sensor model is required. Figure 4.7(b) shows the previous BN extended by a node O representing the (human) observation of the feature class. Its dimension N_o is chosen to be 4: "tree", "shed", "white_object" and "red_car". A human visual observation can represent a subset of all mixture components; *e.g.* a "tree" observation is represented by 7 components (see the legend of Figure 4.6). In loose terms, a human observation o specifies the volume of the state space which corresponds to the feature class: a "tree" observations specifies the volume of the state space enclosed by the 7 "tree" ellipsoids.

The O node's parameter is the conditional probability distribution $p(o|s)$ encoding the human visual sensor model. It is represented by a table of size $N_s \times N_o$ (27×4) as visualised in Figure 4.7(c). Brighter fields indicate higher values and the rows add up to 1.0. The table entries are specified by hand. For each object name, the weights of its corresponding components are distributed equally. The online computation of likelihoods is thus a simple table lookup. The human visual sensor model implies that human operators almost perfectly classify feature classes which is a reasonable assumption for this application. For more complex classification tasks, experiments would be required to find a model. A similar approach to the one described in Section 3.5 could be used.

Dynamic model In order to perform Bayesian filtering, the visual model needs to be extended to a dynamic representation. The BN is extended to a *Dynamic* BN (DBN) as shown in Figure 4.7(d). Additional parameters are the transition models $p(s_{k+1}|s_k)$ and $p(\mathbf{x}_{k+1}|\mathbf{x}_k, s_{k+1})$. They are not required to be learned since the visual state is assumed to be *quasi-static*, *i.e.* its belief is carried across to the next time step without performing a prediction step. Note that this method is different from a memoryless static model which does not keep any history.

Model Interpretation The DBN shown in Figure 4.7(d) is *hierarchical* and can be regarded as an environment representation with multiple abstraction levels. The abstract *appearance* state $\mathbf{x}$ encodes visual properties of a feature in a continuous space. The addition of a human-understandable *identity* state to the representation is useful for two reasons: it (1) offers a higher abstraction level to support analysis and decision making, and (2) permits the incorporation of human observations.

In a probabilistic representation the dependency between identity and appearance results in a statistical correlation, *i.e.* the state beliefs influence each other. This correlation can be used to build up rich complementary feature descriptions with observations originating from robotic platforms and human operators. A robotic sensor observing the appearance of a feature automatically changes the belief of its identity. On the other hand, human operators observing the feature's identity automatically influence the belief of the appearance. The next section shows how the model is used to perform these operations at run-time.

Online Model Usage The software component diagram in Figure 4.8 shows how the visual model is used at run-time. SENSORs extract features via templates or information-theoretic concepts [104] resulting in high-dimensional observations $\mathbf{z}_k$. They are the input for the appearance sensor model which outputs likelihoods $p(\overline{\mathbf{z}}_k|\mathbf{x}_k, s_k)$. This concept decouples feature extraction from representation. The likelihood is a function of $\mathbf{x}$ and s and can be formulated in closed form:

$$p(\overline{\mathbf{z}}_k|\mathbf{x}_k, s_k) = \alpha_s \exp\{-\frac{1}{2}[\mathbf{x}_k - \mathbf{m}_s]^T C_s^{-1}[\mathbf{x}_k - \mathbf{m}_s]\} \tag{4.6}$$

Parameters $C_s, \mathbf{m_s}$ and α_s are calculated as follows:

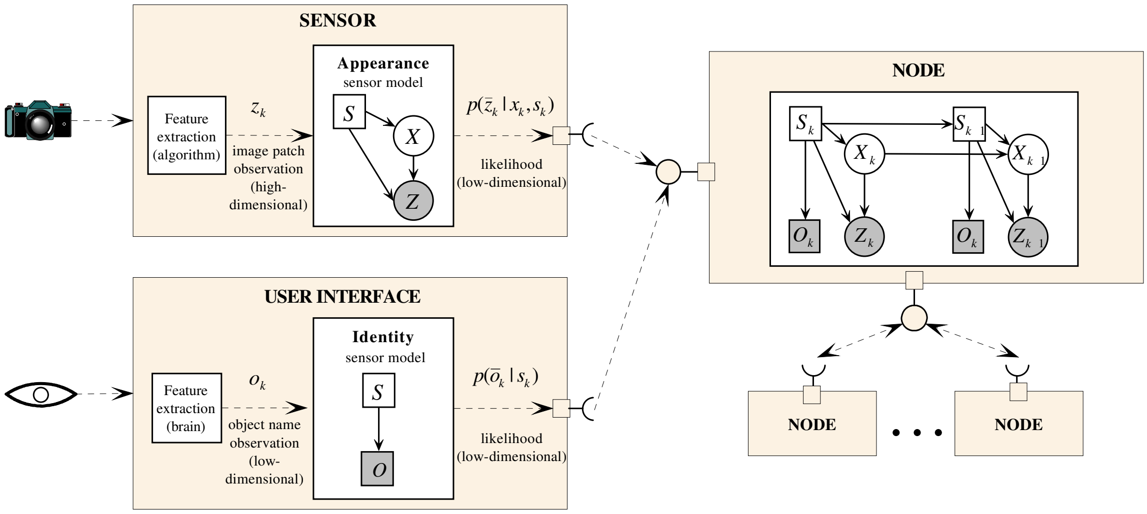


Figure 4.8: Software component diagram for a SENSOR, a USER INTERFACE and a NODE (appearance/identity states only): the SENSOR computes likelihood messages which are communicated to a NODE which in turn computes a posterior belief using the full dynamic model. All NODEs connect into a DDF network.

$$C_s = (\Lambda_s^T \Psi_s^{-1} \Lambda_s)^{-1} \tag{4.7}$$

$$\mathbf{m}_s = C_s \Lambda_s^T \Psi_s^{-1} (\mathbf{z}_k - \mu_s) \tag{4.8}$$

$$\alpha_s = \frac{1}{(2\pi)^{D/2} |\Psi_s|^{1/2}} \exp\{-\frac{1}{2}[-\mathbf{m}_s^T C_s^{-1} \mathbf{m}_s + (\overline{\mathbf{z}}_k - \mu_s)^T \Psi_s^{-1} (\overline{\mathbf{z}}_k - \mu_s)]\} \tag{4.9}$$

The USER INTERFACE, on the other hand, computes likelihoods $p(\bar{o}_k|s)$ using the identity sensor model. Operators specify an object name such as “tree” which is mapped into an unnormalised multinomial distribution of mixture components by a table lookup. Each likelihood is a message which is communicated to a NODE which performs inference on the full dynamic model as previously shown in Figure 4.7(d).

The objective of the NODEs is to maintain a belief of the identity and appearance state given all observations, *i.e.* computing $p(\mathbf{x}_k|\mathbf{z}_{0:k}, o_{0:k})$ and $p(s_k|\mathbf{z}_{0:k}, o_{0:k})$ where $\mathbf{x}_k$ and s_k are appearance/identity state at time step k and $\mathbf{z}_{0:k}, o_{0:k}$ are all robotic and human visual observations up to time step k. Identity state s is used to classify a feature by computing the decision rule $\text{argmax}_s\, p(s_k|\mathbf{z}_{0:k})$.

With the previously mentioned assumption of a static visual environment the recursive computation of the appearance state given all *robotic visual observations* $\mathbf{z}_{0:k}$ is given by:

$$p(\mathbf{x}_k|\mathbf{z}_{0:k}) = \alpha \sum_s \prod_{i=0}^{k} p(\overline{z}_i|\mathbf{x}_i, s_i) p(\mathbf{x}_i|s_i) p(s_i) \tag{4.10}$$

where α is a normalising constant and $p(\overline{\mathbf{z}}_k|\mathbf{x}_k, s_k)$ represents the robotic visual likelihood at time step $i = k$ shown in Figure 4.8 and formulated in Equation 4.6. Expanding Equation 4.10 shows the parallel structure of the update:

$$\begin{aligned} p(\mathbf{x}_k|\mathbf{z}_{0:k}) \quad = \quad & \alpha[p(\overline{\mathbf{z}}_k|\mathbf{x}_k, s_k = 1) \ldots p(\mathbf{x}_0|s_0 = 1) p(s_0 = 1) + \\ & \vdots \\ & p(\overline{\mathbf{z}}_k|\mathbf{x}_k, s_k = N) \ldots p(\mathbf{x}_0|s_0 = N) p(s_0 = N)] \end{aligned} \tag{4.11}$$

This parallel structure suggests that recursive estimation of the visual states can be implemented as a sum of N filters (one line corresponds to one filter). Each filter is initialised with the learnt prior $p(\mathbf{x}|s)p(s)$, which is of Gaussian form. When an observation $\mathbf{z}_k$ is made, the filter number n is multiplied by the likelihood $p(\overline{\mathbf{z}}_k|\mathbf{x}_k, s_k = n)$ which is also of Gaussian form (Equation 4.6). Thus, each filter in the sum only involves Gaussian terms and as a result reduces to a linear Kalman filter.

The identity state represents the weight of each Gaussian in the sum. It is updated as follows:

$$p(s_k|\mathbf{z}_{0:k}) = \alpha p(\overline{\mathbf{z}}_k|s_k) p(s_k|\mathbf{z}_{0:k-1}) \tag{4.12}$$

where $p(\overline{\mathbf{z}}_k|s_k)$ can be obtained in two steps. First, the joint distribution $p(\mathbf{z}, \mathbf{x}|s)$ is computed by multiplying $p(\mathbf{z}|\mathbf{x}, s)$ by $p(\mathbf{x}|s)$ (Equation 4.4-4.5):

$$p\left(\begin{bmatrix} \mathbf{z} \\ \mathbf{x} \end{bmatrix}\right) = \mathcal{N}\left(\begin{bmatrix} \mu_s + \Lambda_s \nu_\mathbf{s} \\ \nu_s \end{bmatrix}, \begin{bmatrix} \Psi_s + \Lambda_s \Sigma_s^T \Lambda_s^T & \Lambda_s \Sigma_s \\ \Sigma_s^T \Lambda_s^T & \Sigma_s \end{bmatrix}\right) p(s). \tag{4.13}$$

Second, the joint $p(\mathbf{z}, \mathbf{x}|s)$ is marginalized over $\mathbf{x}$ to obtain:

$$\begin{aligned} p(\overline{\mathbf{z}}_k|s_k) \quad = \quad & \frac{1}{(2\pi)^{D/2}|\Psi_s + \Lambda_s \Sigma_s^T \Lambda_s^T|^{1/2}} \times \\ & \exp\{-\frac{1}{2}(\overline{\mathbf{z}}_k - \mu_s - \Lambda_s \nu_s)^T (\Psi_s + \Lambda_s \Sigma_s^T \Lambda_s^T)^{-1} (\overline{\mathbf{z}}_k - \mu_s - \Lambda_s \nu_s)\} \end{aligned} \quad (4.14)$$

Under the same assumption of a static visual environment, the recursive computation of the appearance state given all *human visual observations* $o_{0:k}$ is given by:

$$p(\mathbf{x}_k|o_{0:k}) = \alpha \sum_s \prod_{i=0}^{k} p(\overline{o}_i|s_i) p(\mathbf{x}_i|s_i) p(s_i) \quad (4.15)$$

with $p(\overline{o}_k|s_k)$ representing the human visual likelihood at time step k as shown in Figure 4.8.

Equation 4.15 follows the structure of Equation 4.11. Human visual likelihoods can be integrated into the sum of N Kalman filters. For example, a human visual observation at time step k can be fused as follows:

$$\begin{aligned} p(\mathbf{x}_k|\overline{o}_k, \mathbf{z}_{0:k-1}) \quad = \quad & \alpha[p(\overline{o}_k|s_k = 1) p(\overline{\mathbf{z}}_{k-1}|\mathbf{x}_{k-1}, s_k = 1) \ldots p(\mathbf{x}_0|s_0 = 1) p(s_0 = 1) + \\ & \vdots \\ & p(\overline{o}_k|s_k = N) p(\overline{\mathbf{z}}_{k-1}|\mathbf{x}_{k-1}, s_k = N) \ldots p(\mathbf{x}_0|s_0 = N) p(s_0 = N)] \end{aligned} \quad (4.16)$$

The identity state for each of the N filters is updated as follows:

$$p(s_k|o_{0:k}, \mathbf{z}_{0:k-1}) = \alpha p(\overline{o}_k|s_k) p(s_k|o_{0:k-1}, \mathbf{z}_{0:k-1}) \quad (4.17)$$

To summarise, the absence of explicit online data compression through the use of likelihoods, and the update reducing to N Kalman filters, result in a computationally efficient filtering scheme which can be adopted in real-time applications.

4.2.3 Combined Data Association

Data association refers to the problem of assigning observations to existing tracks in the filter. To cull very unlikely observation-to-track associations, *validation gating* can be performed [11]. Formulating a gate that is computable in real-time is still an open problem for non-Gaussian filters [8]. In this system, gating is performed by thresholding.

The visual state space can be used to improve data association which is otherwise based on geometry only. A measure called *evidence* of observations with respect to existing tracks is used to rank data association hypotheses. It is expressed as $p(\overline{\mathbf{z}}_p, \overline{\mathbf{z}}_v|\mathcal{H}_i)$ with $\overline{\mathbf{z}}_p$ and $\overline{\mathbf{z}}_v$ being geometric and visual observations, respectively. $\mathcal{H}_i$ is the hypothesis *"observations $\overline{\mathbf{z}}_p$ and $\overline{\mathbf{z}}_v$ were generated by track i"* [8]. Since geometric and visual state spaces are assumed statistically independent (see introduction to Section 4.2), this expression becomes a product of the position and visual evidence: $p(\overline{\mathbf{z}}_p|\mathcal{H}_i)p(\overline{\mathbf{z}}_v|\mathcal{H}_i)$. Geometric evidence is obtained by summing the weights of the unnormalised GMM resulting from the position update of track i with observation $\overline{\mathbf{z}}_p$ as shown in [8].

Visual evidence at time step k is computed as follows:

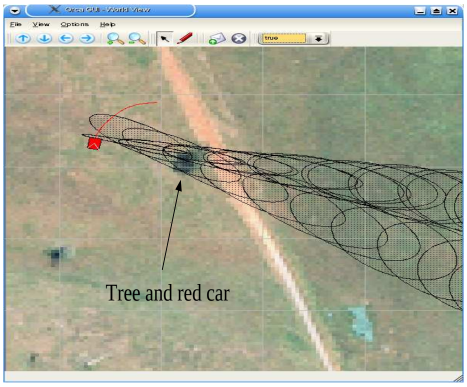
$$p(\overline{\mathbf{z}}_{v,k}|\mathcal{H}_i) = \sum_s p(\overline{\mathbf{z}}_{v,k}|s, \mathcal{H}_i)p(s|\mathcal{H}_i) \tag{4.18}$$

where $p(s|\mathcal{H}_i)$ is given by the weights of track i (Equation 4.12), and $p(\overline{\mathbf{z}}_{v,k}|s, \mathcal{H}_i) = \int p(\overline{\mathbf{z}}_{v,k}|\mathbf{x}_{v,k}, s, \mathcal{H}_i)p(\mathbf{x}_{v,k}|s, \mathcal{H}_i)d\mathbf{x}_{v,k}$.

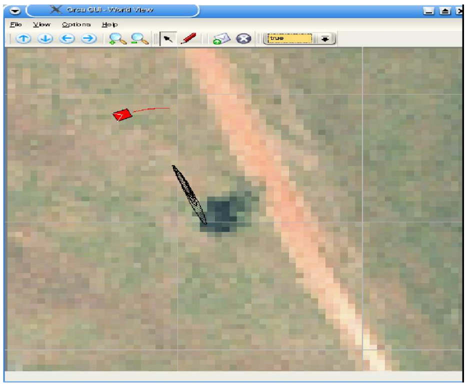
In [182], the probabilistic Bhattacharyya distance is used for data association. This distance evaluates the similarity between an incoming likelihood and existing tracks. Its disadvantage is that the distances scale differently in the position and visual space. As a result, distances computed in the two spaces must be arbitrarily weighted so that they can be combined in a decision rule for data association. The use of evidences does not lead to this problem. The evidences $p(\overline{\mathbf{z}}_p|\mathcal{H}_i)$ and $p(\overline{\mathbf{z}}_v|\mathcal{H}_i)$ are conditional probabilities which naturally scale between zero and one and can be readily compared without resorting to a pre-defined scaling.

Results from the combined data association scheme are presented in Figure 4.9. Figure 4.9(a) shows a raw image of two environment features physically close to each other, a tree and a red car. Figure 4.9(b) visualises two successive bearing-only observations from a ground vehicle resulting in two conically shaped Gaussian mixtures. Data association solely based on position results fuses the two observations into a single track as shown in

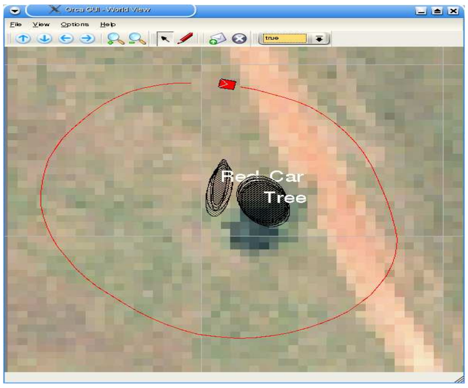
(a) Two features



(b) Two GMM observations



(c) Incorrect fusion



(d) Discrimination

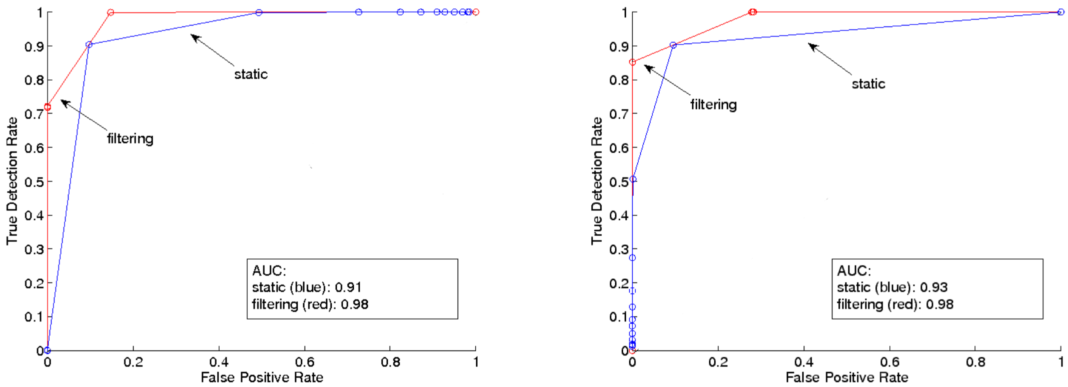
Figure 4.9: Improved data association using a scheme combining position and visual states: (a) two environment features close to each other, a tree and a red car, (b) two successive bearing-only observations from a ground vehicle, (c) incorrect fusion of observations into a single track based on position only, and (d) the combined data association scheme ensures discrimination between the two features.

Figure 4.9(c). Compare this to Figure 4.9(d) where two tracks with corresponding labels are created using the combined data association scheme[2].

4.2.4 Classification using Filtering

Besides improving data association, the visual model is also used for feature classification. It is shown here that incorporation of past information through the filtering process ultimately increases classification accuracy compared to a standard (static) approach which ignores

[2]Note that Figures 4.9(c) & (d) were generated using the same data set but show estimates at different time steps for better visualisation.



(a) Tree vs other classes (b) Red_car vs other classes

Figure 4.10: Filtering improves classification results of static classifier, results are presented as ROC curves: (a) result for "tree" versus all other classes, and (b) result for "red_car" versus all other classes. Information from several time steps is used for filtering.

past information. The effectiveness of filtering over visual states can be quantified by considering classification accuracy over multiple time steps.

The decision rule for static classification is to compute $\operatorname{argmax}_s p(\overline{\mathbf{z}}_k|s)$. Classification with filtering is determined by $\operatorname{argmax}_s p(s|\mathbf{z}_{0:k})$ incorporating all past observations (see Equation 4.12). Results are presented in the form of Receiver Operating Characteristic (ROC) curves as shown in Figure 4.10. Two curves are drawn: static and filtering approach. The object classes "tree" and "red car" were analysed with replayed data obtained from 13 individual runs of the ground vehicle representing 2.8 hours of logging. 350 tracks were observed multiple times with an average of 6 updates.

A measure to evaluate multi-class classifiers is the sum of the areas under the ROC curve (AUC) per class weighted by their prevalence in the data [50]. As shown in the figure, the AUCs for filtering over visual states are higher than the AUCs for the static classifiers.

4.3 Experiments

This section presents experimental results from a deployed human-robot team using the shared geometric and visual representation. First, the architecture of the system is explained. Second, an evaluation of the information fusion results for a single experimental run is given.

4.3.1 System Overview

The human-robot team deployed for the experiments consists of multiple platforms: a ground vehicle, a UAV and two human operators as shown in Figure 4.2. Figure 4.11 shows a georeferenced aerial image of the outdoor environment, the (desired) flight path of the UAV, the area covered by the ground vehicle and the human operators (rectangle) and the UAV platform.

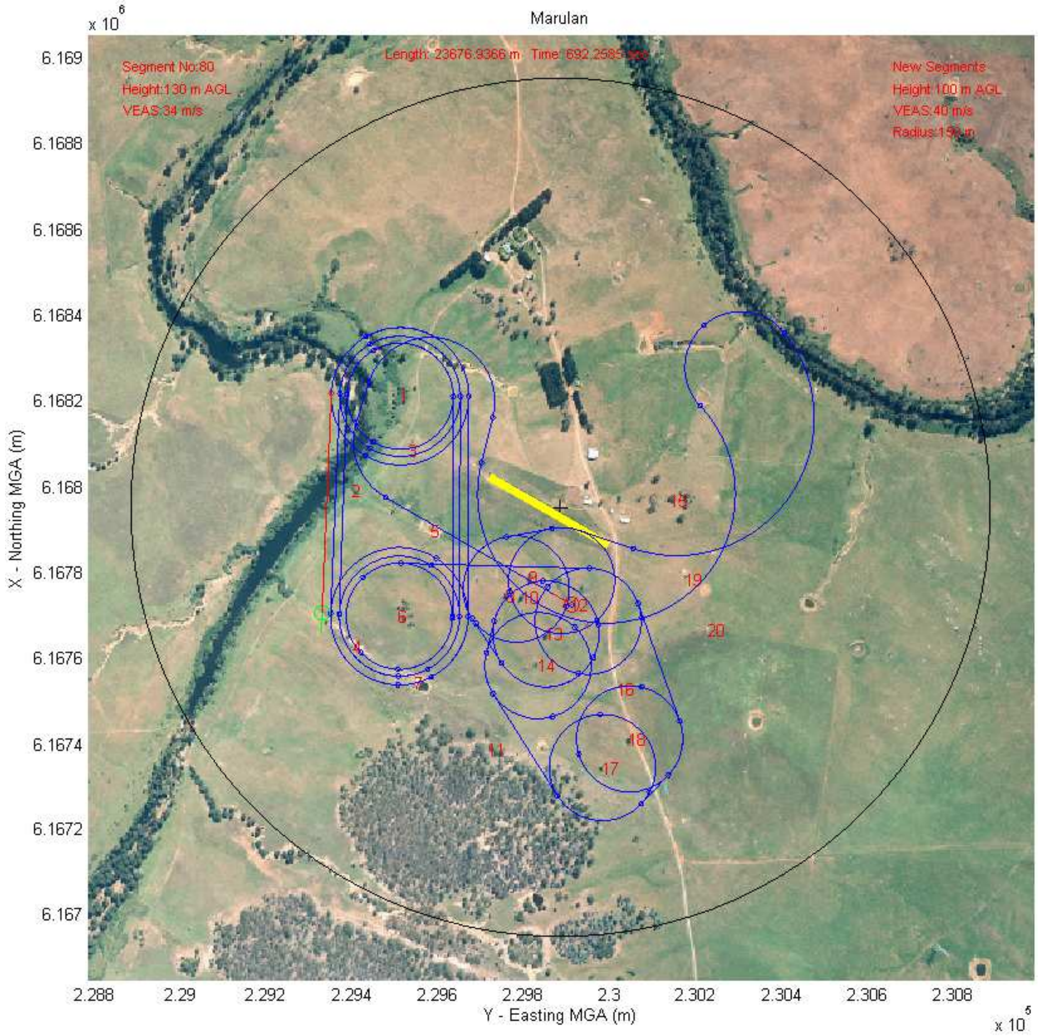
Figure 4.12 shows the software architecture of the system as a UML deployment diagram. All software components expose interfaces to inter-operate with other components. IMAGE-SERVERs and LOCALISERs connect to hardware shown as *artifacts*. Components are run on seven host computers visualised by 3d-boxes. Hosts are part of a physical platform whose boundaries are indicated with dashed lines. The backbone of the system, the DDF network, is highlighted. For clarity, deployed components and interfaces related to monitoring and logging are not shown. In total, up to 20 software components were running at any given time.

The system works as follows: robotic platforms run a LOCALISER which connects to GPS/INS hardware and provides an estimated pose of the platform. The position estimate is sent to the SENSOR which extracts features from captured images [104]. Together with the platform pose information, it can compute a likelihood of the feature position in a global coordinate system. The SENSOR also encodes the visual properties of the features which are related to their class (*e.g.* “tree”). SENSORs communicate their likelihoods to NODEs which fuse them with prior beliefs. They communicate the resulting posterior to other NODEs to maintain a common belief.

Human operators are also part of the system and all roles mentioned in Section 4.1.2 were implemented within the project. Figure 4.12 only shows the roles related to environment information.

4.3.2 Multi-Attribute Feature Map

Figure 4.13-4.18 show results from the final experimental demonstration as graphical USER INTERFACE screenshots. The USER INTERFACE was used by operators to communicate with the system and serves several purposes: it (1) displays the global world belief as a multi-attribute feature map, (2) displays the platforms and their trajectories, and (3) enables the



(a) Aerial image of test facility

(b) Brumby on runway

(c) Brumby in air

Figure 4.11: (a) A georeferenced aerial image of the test facility showing the UAV's (desired) flight path and the area covered by the ground vehicle and human operators (rectangle). (b) The brumby UAV on the runway and (c) in the air.

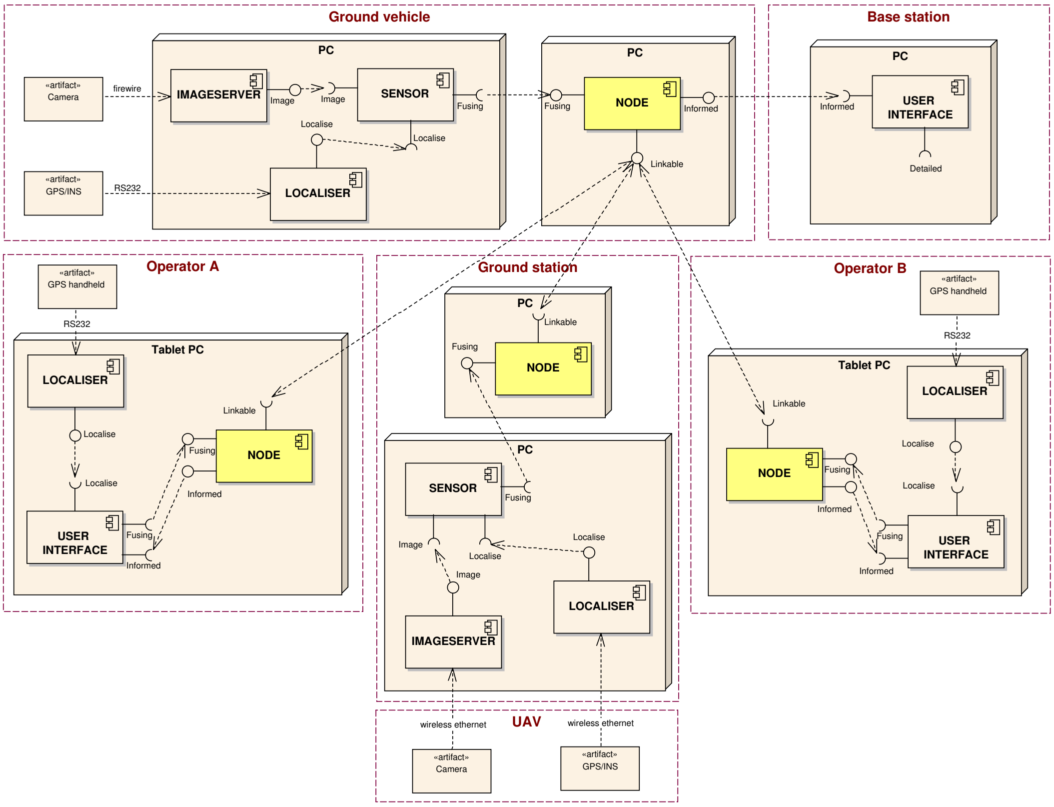


Figure 4.12: The system's architecture as a UML deployment diagram: components run on hosts which are part of a physical platform (robot, human operator, or fixed station). Some components are connected to pieces of hardware shown as artifacts. The DDF network is highlighted.

operator to enter observations. For the latter task, geometric and identity sensor models required to compute likelihoods appropriate for fusion are stored internally.

The USER INTERFACE displays an overlaid aerial map of the environment and a regular grid of cell size $50m \times 50m$. All Gaussians are drawn to scale with 3σ uncertainty ellipses. The fill pattern of the GMM components correspond to their weight – the denser, the more probability mass on the component. Identity states are also shown by summarising the most probable components by a single name ("tree" instead of "tree1") printed on top of the ellipses. Platform representations are iconic, *i.e.* their size is fixed for all zoom configurations.

Figure 4.13 shows the multi-attribute feature map built during the final demonstration of the system. The left image is an aerial photograph of the area covered by the ground vehicle

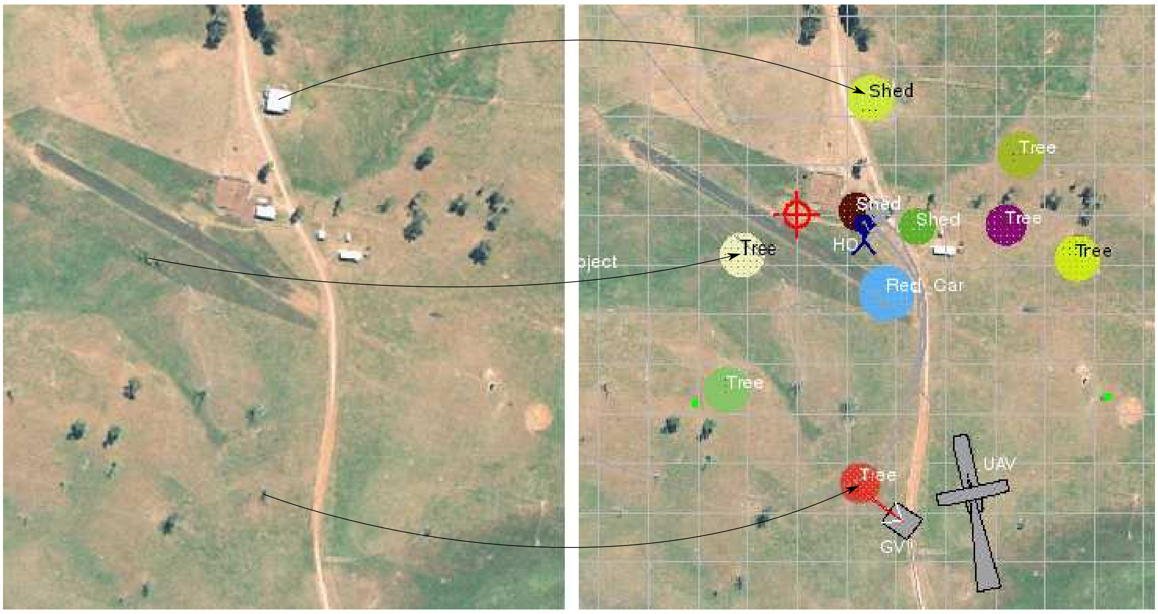
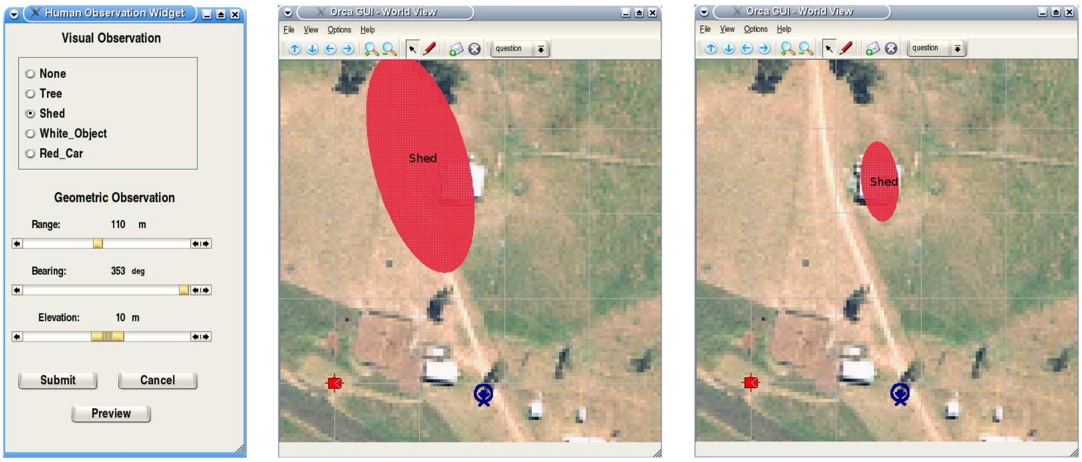


Figure 4.13: Right: A multi-attribute feature map as a result of the experiments. Features are represented by position including uncertainty (coloured ellipses) and their labels (most probable identity). Platforms are shown as icons; UAV = unmanned air vehicle, GV = ground vehicle, HO = human operator. Left: rectangular section of Figure 4.11(a) used as a ground truth reference with arrows indicating the correspondence of real features to the probabilistic representation.

and human operators as previously shown in Fig 4.11(a). It is shown here as a ground truth reference. The image on the right is a USER INTERFACE screenshot visualising the probabilistic feature representation. Note that the GMMs for a single feature mostly lie on top of each other after having been observed several times from different angles. Correspondences of real features in the environment with the feature representation is indicated by arrows from the left to the right image.

4.3.3 Information Exchange Patterns

The integration of human operators into the data fusion system is now illustrated by presenting five information exchange patterns. The scenarios described here demonstrate human-robot peer-to-peer interaction. Unlike in many automated systems where humans act on a higher level of authority, the operator cannot override information acquired by the robots.



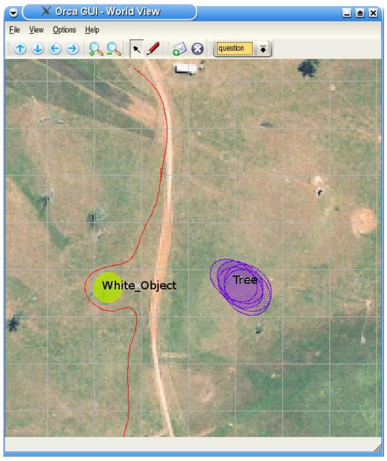
(a) GUI (b) Observation (c) Fusion result

Figure 4.14: Demonstration of the raw observation method through operator-only fusion: (a) a GUI for entering raw geometric observations and labels, (b) a "shed" feature resulting from a raw observation by the operator drawn as a blue stickman (including position uncertainty), and (c) the more informative feature after multiple observations.

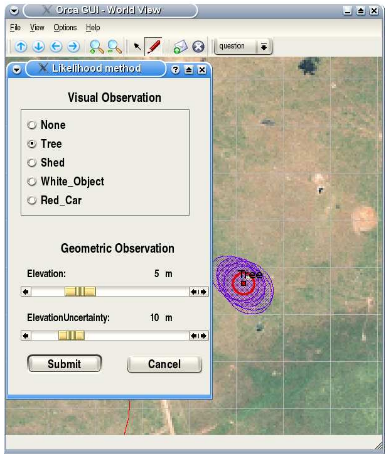
Instead, information from all team members complement each other. All exchange patterns are generated by playing back data from the final experimental demonstration.

Operator-Only Fusion Figure 4.14 illustrates the raw observation method. The operator is displayed as a blue stickman including the person's position uncertainty as a blue ellipse. The operator inserts a raw range/bearing/elevation observation of a shed through the GUI displayed in Figure 4.14(a). After reobserving the feature several times with refined values the final Gaussian uncertainty has significantly decreased with the centre on the true position of the shed (Figure 4.14(c)).

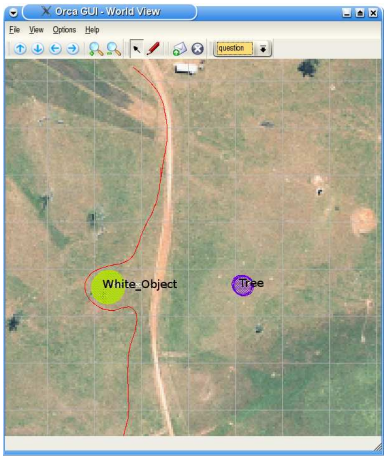
Refinement by Operator Figure 4.15 demonstrates a scenario where the original information is supplied by the ground vehicle and is later refined by the operator. Two features previously observed by the vehicle are shown in Figure 4.15(a) (the red line is the vehicle's trajectory). A human operator decides to refine the position of the tree on the right by drawing an ellipse with smaller uncertainty on top of the GMM (Figure 4.15(b)). The result is a more compact GMM as shown in Figure 4.15(c).



(a) Two features



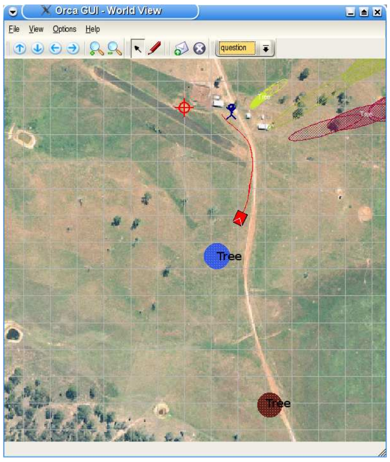
(b) Location refinement



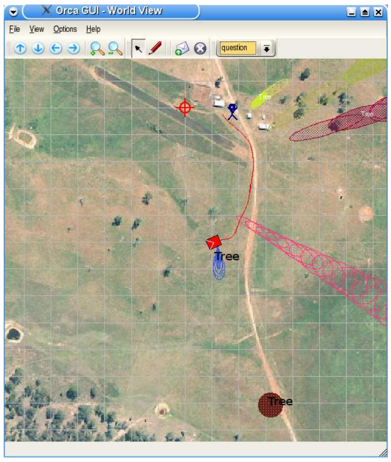
(c) Fusion result

Figure 4.15: Human operator refining a feature entered by a robot: (a) two GMM features previously observed by a passing ground vehicle (trajectory in red), (b) operator refines feature location through likelihood observation, and (c) the resulting fused feature with less uncertainty.

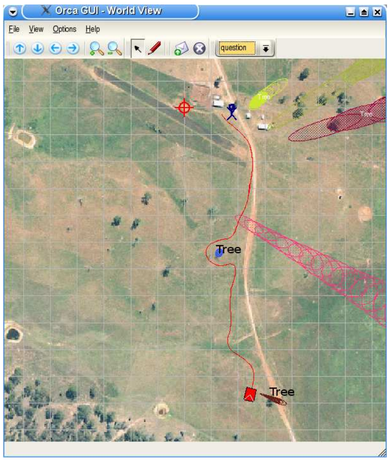
Refinement by Robot This scenario reverses the refinement order of the previous pattern. Figure 4.16 shows how an operator enters two features prior to the mission which are subsequently observed by the ground vehicle. Figure 4.16(a) includes the moving vehicle with its path, a stationary human operator at a control station, a few GMM features in the top right corner and the two "tree" features represented by ellipses. Figure 4.16(b) and (c) illustrate the refinement of the position estimate as the ground vehicle drives past: the estimates are converted into GMM form with smaller uncertainty. The final feature repre-



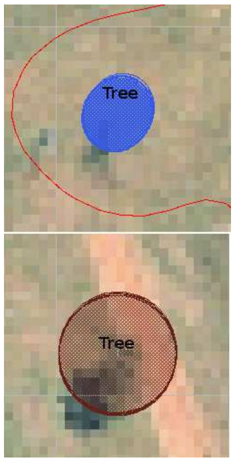
(a) Two human features



(b) Observation feature 1



(c) Observation feature 2

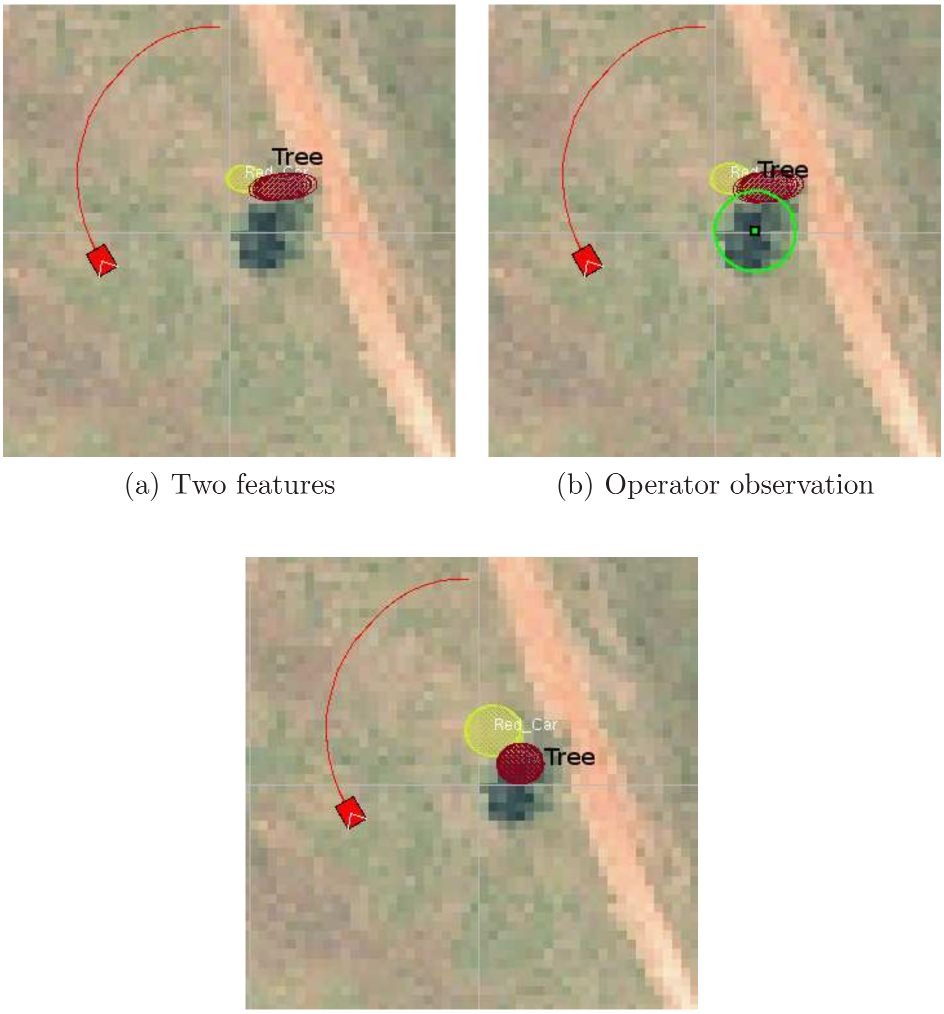


(d) Fusion results

Figure 4.16: Robot refining features initialised by human operators: (a) two "tree" features are initialised prior to the arriving ground vehicle, (b) first feature is shrunk to a GMM based on a bearing-only observation, (c) second feature is shrunk, and (d) the final (compact) feature representation in a zoomed-in view.

sentations have very low entropy and are almost confined to a single Gaussian as displayed in the zoomed-in view of Figure 4.16(d).

Bias Correction by Operator Figure 4.17 shows a scenario where an operator "corrects" a track which is geometrically biased. The green patch from the aerial map indicates the true position of the tree. Due to inaccurate camera calibration, the ground vehicle placed a "tree" track a few metres off the true position. After a human operator has en-



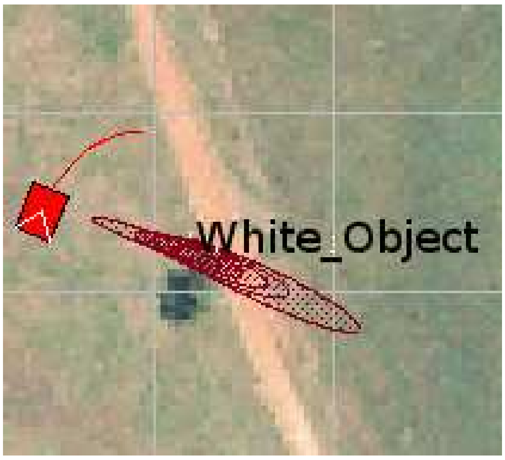
(a) Two features (b) Operator observation

(c) Fusion result

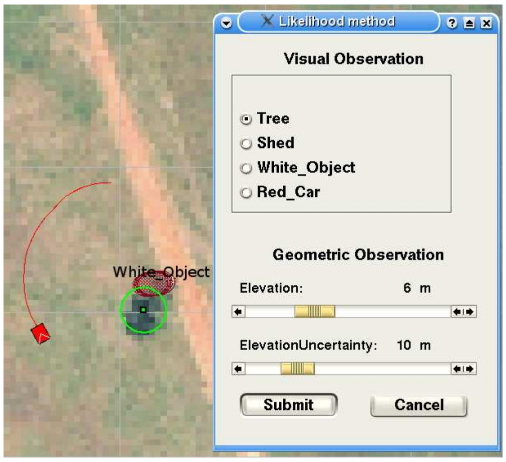
Figure 4.17: Operator "correcting" a bias in position: (a) two features are observed by the ground vehicle, the "tree" feature is off the true position (green patch), (b) operator enters a "tree" observation around the true position, (c) after two observations the "tree" feature is closer to the truth.

tered two "tree" observations closer to the truth (Figure 4.17(b)), the bias becomes smaller as shown in Figure 4.17(c).

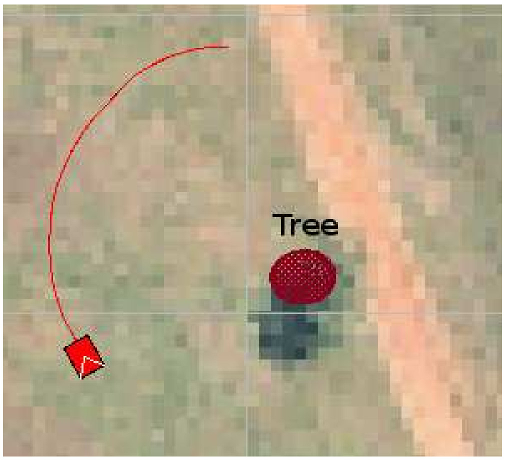
Label Correction by Operator In this scenario, an operator "corrects" a label which was wrongly assigned by the ground vehicle (Figure 4.18(a)). The operator enters a "tree" observation close to the "white_object" track as shown in Figure 4.18(b). Note that it depends on the tuning of data association parameters whether this observation is assigned



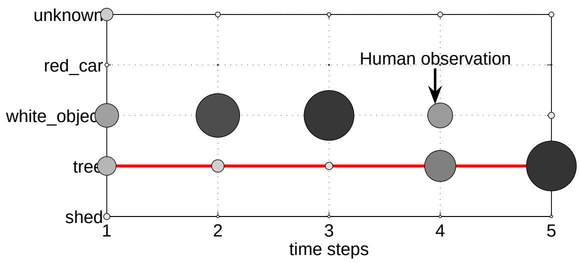
(a) Single feature



(b) Operator observation



(c) Fusion result



(d) Filter behaviour

Figure 4.18: Operator "correcting" the identity of a feature: (a) ground vehicle wrongly assigns a "white object" label to a "tree" feature, (b) operator observes the same feature as a tree (also correcting for position bias, see Figure 4.17), (c) the label becomes "tree", (d) the filtering of the identity state: three robotic and two human observations. Marker size and colour are proportional to the probability mass. The red line indicates the truth.

to the track. In this scenario, it was assigned and fused as displayed in Figure 4.18(c). Figure 4.18(d) shows how the probabilities of the identity state change over time (marker colour and size are proportional to the probability mass). The first three observations were performed by the ground vehicle. After the human observations, the probability mass shifted towards the true identity state.

4.4 Related Work

This section presents related research in scalable human-robot interactions and shared representations for human-robot teams.

4.4.1 Scalable Interactions

The field of adjustable autonomy has recently started to address human interactions with multi-robot systems [32][133]. This problem of interacting with several robots is especially evident for UAV control where the ratio of operators to UAVs is typically higher than one at present [44]. Most of these systems apply supervisory control schemes which are aimed at controlling a small number of robots at the same time but may not scale up indefinitely. In contrast, this work addresses the problem of scalability from the ground up by proposing to let human operators contribute environment information which *indirectly* controls the individual platforms [115].

Scalable human-robot interactions are explicitly addressed in Tews' work [171]. Here, scalability refers to the communication bandwidth between humans and robots which needs to be kept low for larger numbers of entities in the system. It is proposed that the amount of communication increases with decreasing robot autonomy, tighter human-robot coupling, and the number of robots. The authors suggest that a large-scale interaction mechanism must allow for many-to-many, one-to-many, and one-to-one interactions. Additionally, it must allow heterogeneous robot teams and different levels of human-robot coupling. The architecture is server-client based and therefore relies on a centralised server. Information fusion is not addressed in this work.

Our previous work dealt with scalable interactions between humans and a sensor network which laid the foundation of this chapter [94]. Three significant extensions have been presented here: (1) the utilisation of a multi-level representation, (2) the probabilistic modelling of human input, and (3) an extensive experimental demonstration.

4.4.2 Shared Representation for Human-Robot Teams

Researchers in the field of human-robot interaction (HRI) have recently identified the problem of finding a *shared representation* between humans and robots for mapping. Relating geometric maps typically acquired by a robot to the topological view of the environment typically developed by humans [119] has become a popular research area [183][177][103][125][25]. Many of these approaches let the robot build a human-understandable representation but operators are not able to contribute information online. An exception is the work of Topp *et al.* which explicitly acknowledges a human in the loop in a *Human Augmented Mapping* approach [177]. Humans and robots cooperate interactively to build personalised maps which are augmented using clarification dialogs [103]. Humans act as teachers while we regard them as teammates providing uncertain information which can be modeled probabilistically.

A similar effort to build a common substrate or a *collaborative workspace* is presented in [25]. It is argued that a common meaningful representation is required for a human-robot team to achieve a common goal. A mixed team of air/ground vehicles and human operators similar in scope to our system cooperate in a map building mission. The map contains both, lower-level terrain information as well as semantic abstractions provided by a user. A point-and-click interface allows the operator to add objects such as victims in a search-and-rescue scenario. Even though demonstrating a simple scenario of collaborative perception, the focus of their framework is on control and adjustable autonomy [23].

A team of air and ground vehicles were deployed to build a common representation as part of the DARPA MARS project [72][70]. Like in our work, DDF algorithms are used to detect and localise targets in a scalable manner. The advantages of integrating complementary capabilities of air and ground vehicles are emphasised and demonstrated experimentally: while UAVs exhibit better area coverage, ground vehicles can localise features more accurately. This aspect represents a parallel to our work while we add another complementary information source: *human* perceptual capabilities are integrated synergistically with sensor information from air and ground vehicles. In contrast, the human operators' role in the MARS project was limited to task monitoring and supervisory control of the UAVs.

An earlier work addressing the need for a vocabulary common to humans and robots is presented in [99][98]. The authors show how communication between humans and robots can benefit from machine learning by building human-understandable symbols from a robot's perceptions and actions – its low-level internal representation. Human operators act as teachers by providing examples of *concepts* to the robot which builds its own internal representation. Concepts are organised hierarchically on five different abstraction levels [99]. The overall goal is to learn a high-level description of actions, perceptions and the environment from low-level sensing data. The multi-level representation is encoded using inductive Horn logic and thus, is not able to incorporate uncertainties easily. Their results at the lower levels show this weakness. However, concepts at a high-level encoded as Prolog clauses are easy to understand for humans.

4.5 Summary

This chapter showed how a team of humans and robots can interact in a scalable way while cooperating in an information gathering task. Scalability is a property of the DDF algorithms which no member of the team can violate. If human operators mainly act as environment information sources, scalability can be guaranteed. A shared environment representation was presented which is built on multiple abstraction levels in order to represent a complex environment, and to facilitate human information input.

Scalable human-robot information fusion was experimentally validated in an outdoor environment. Two human operators, one UAV and a ground vehicle cooperatively built a multi-attribute map of the environment. Interesting information exchange patterns between operators and the DDF network emerged during the experiment.

This chapter focused on "pure" information fusion, *i.e.* robots did not use the fused information to make autonomous decisions. Also, to guarantee scalability, only environment information was fused. In contrast, the next chapter will add decision-making capabilities to the robots and the representation will contain a mix of environment and platform states. The underlying mechanism of human-robot information fusion, however, will not change: observations from humans and robots will be fused in a similar way but the resulting beliefs will now be used by the robots to make decisions.

Chapter 5

Cooperative Human-Robot Decision Making

While the previous chapter was exclusively concerned with perception, this chapter shows how fused observations from humans and robots can be used to make decisions. A decision-theoretic framework is utilised to make platform decisions which are executed by the robot's actuators as visualised in Figure 5.1. Since the action selection depends on observations from both robotic sensors and human operators, this mechanism is referred to as *cooperative* human-robot decision making.

In this chapter, human resources are treated as limited and thus expensive. It is also assumed that robotic sensors can yield observations at a high rate and low cost[1]. For this scenario, the robot-pull communication pattern as described in Section 1.3.2 is most appropriate since the robot can decide when to make use of the human resource. Letting the robot decide when to query operators can be seen as a mechanism for *Adjustable Autonomy* (AA).

Work presented in this chapter is most closely related to Fong's *Collaborative Control* which also lets the robot pose questions to the operator [53]. Unlike Fong's work which does not specify a mathematical model, this chapter casts collaborative control in a decision-theoretic formulation. The formulation has several advantages: (1) human and robotic observations can be combined in a consistent manner, (2) decisions can be made while

[1]Note that this is only one of many scenarios. Remote planetary exploration is an example where human resources may potentially be cheaper.

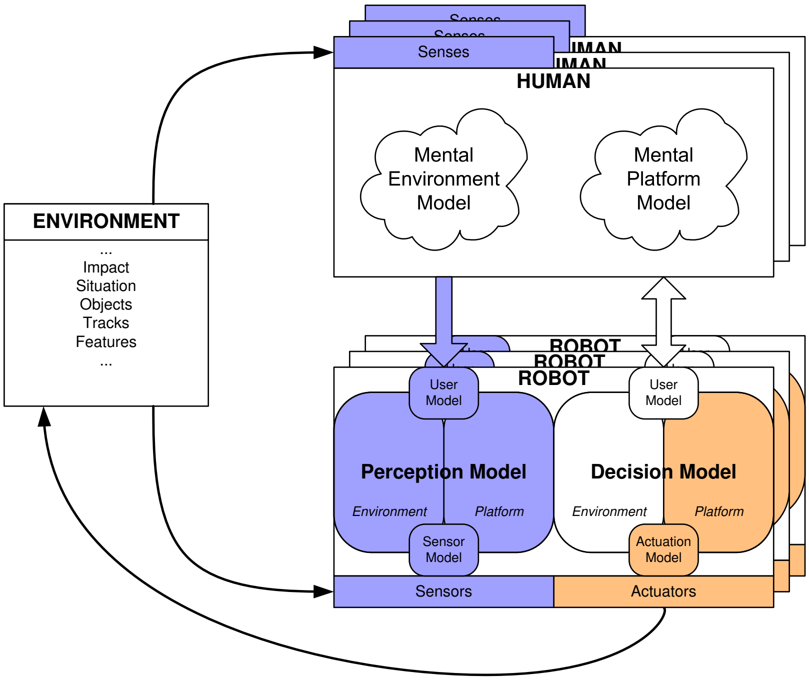


Figure 5.1: This chapter shows how humans and robots can cooperate in making platform decisions by fusing perceptual information. Decisions are executed by the robot using its actuators.

taking the uncertainty of the environment, the platforms, sensors, and human operators into account, and (3) the circumstances under which human operators should be queried can be determined using *Value-Of-Information* (VOI) theory. Unlike in many classical AA systems (see [24]), the robot's autonomy is not switched by the operator in the system presented here. Instead, queries are triggered automatically by the level of uncertainty in the robot's belief. Another novelty is that operators enter perceptual information rather than decisions to control robots. A detailed literature review is presented at the end of the chapter.

The remainder of the chapter is organised as follows. Section 5.1 shows how a decision-theoretic framework can address the questions of what queries to pose and when to pose them – the mechanism the robot uses to relinquish full autonomy. Next, a generic architecture is presented capable of incorporating multiple robots and multiple human operators. Section 5.2 presents an influence diagram representation for a navigation task and demonstrates how the robot-pull communication pattern works for this task. An experimental evaluation of the system in simulation and an extensive user study are presented in Section 5.3. Related work is presented in Section 5.4 before Section 5.5 summarises.

5.1 Robots Query Humans

This section gives reasons why the proposed decision-theoretic framework is suitable for cooperative human-robot decision making with humans acting as a resource to robots.

5.1.1 Query Types and Triggers

The types of queries robots should pose are well defined if humans are regarded as information sources for the probabilistic representation of the environment and the platforms: human operators are queried for observations which are subsequently incorporated into the representation as evidence. The question arises under what circumstances they should be queried. VOI theory as presented in Section 2.4.3 is able to answer this question while satisfying the following requirements: only query for relevant information, pose queries in a reasonable order, and most importantly, take the cost of obtaining evidence into account.

Querying operators can be seen as a robot-initiated shift to lower autonomy at run-time. Whether an autonomy shift is triggered depends on the robot's current probabilistic beliefs and the cost parameter of Equation 2.16. For this chapter, the cost of obtaining evidence from human operators is fixed prior to a mission. If the cost parameter value is high, the robot relies more on its own perceptual capabilities, asks fewer questions, and is more autonomous by definition. Setting the cost parameter can thus be seen as fixing the level of autonomy. We define *autonomy level* as a mission-specific design parameter which determines the degree of autonomy the robot has for a given mission scenario. The parameter should be set according to the constraints of the particular mission. An example of a mission constraint is the communication bandwidth. Section 5.3.1 proposes a methodology for determining the autonomy level systematically by taking mission constraints into account. The terms *autonomy level* and *cost* (of obtaining evidence) are used interchangeably throughout the rest of the chapter.

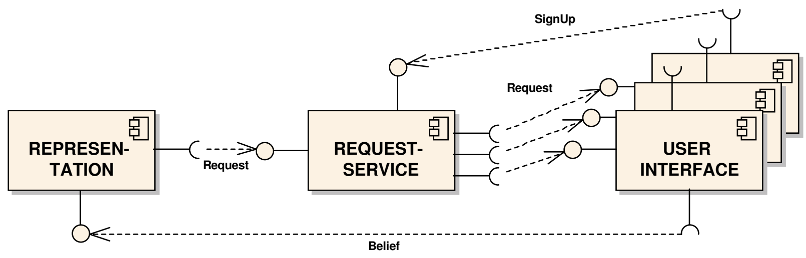


Figure 5.2: Generic architecture for the robot-pull pattern. Rectangles are components with provided and required interfaces shown as circles and semi-circles, respectively. Arrows indicate information flow. The Representation component represents the robots' internal perception and decision models as shown in Figure 5.1.

5.1.2 Robot-Pull Architecture

Figure 5.2 shows a generic[2] human-robot architecture for the robot-pull pattern. The Representation component generates requests based on Equation 2.16 which are sent to the RequestService. The RequestService acts as a query arbiter by relaying requests to suitable human operators who are represented by user interfaces. Operators would have signed up previously by specifying the Quality of Service (QoS) they can provide such as their *expertise* and the sensor *information type* they have access to (*e.g.* laser scan, camera image). When operators receive requests, they can submit observations via the `Belief` interface[3].

The architecture is able to handle many-to-many interactions. Multiple user interfaces can be serviced by the RequestService. While Representation is shown as a single component, it can represent a multi-robot system as mentioned in the following section.

5.1.3 Difficulties with Robot-Pull

Special considerations arise when robots request information from human operators [53]:

1. What if operators are not available or do not answer queries?
2. How are individual differences amongst operators handled?
3. How suitable is robot-pull for many-to-many interactions?

[2] *i.e.* not tied to a particular application/representation

[3] An example of a similar system is the Amazon Mechanical Turk [188] – an online marketplace to distribute tasks humans perform better than computers.

The contribution provided here is to address these questions using a decision-theoretic framework which can provide the following answers:

1. *Autonomy*: a decision-theoretic framework can yield a decision at any time, *i.e.* the system is always capable of operating fully autonomously. Questions are only triggered if it is worthwile obtaining additional observations due to high uncertainty about a relevant internal state. If no answers are received, the system remains capable of acting rationally by selecting the best decision using Equation 2.11.

2. *Uncertainty about operators*: varying levels of user expertise are taken into account because the robot needs to handle answers from experts and novices differently [53]. The level of user expertise is encoded as a probabilistic HSM as presented in previous chapters. Mathematically there is no difference between a robotic and a human sensor model, reinforcing the notion of humans and robots acting as peers.

3. *Scalability*: one concern with using a robot-pull pattern is its perceived inability to scale to multi-robot systems [115]. Chapter 4 argued that truly scalable systems can only be achieved if the representation is concerned solely with the *environment*, not the robotic platforms themselves. This is feasible by decentralising the inference applied to an environment representation. Scalability of cooperative decision making is not experimentally demonstrated in this chapter and left for future work as discussed in Section 6.2.2.

5.2 Representation for Navigation

This section introduces the representation used in the experiments presented in Section 5.3. Unlike the representation presented in Chapter 4 which was solely concerned with the perception of the environment, the model presented here is used to perceive both environment and platform states as well as to make platform decisions. More specifically, it encodes a robot's low-level decision model to choose from a discrete set of actuator commands for a simple navigation application.

Figure 5.3 shows the decision model encoded as an ID. Many well-researched methods in machine learning and knowledge engineering address the problem of how to construct such

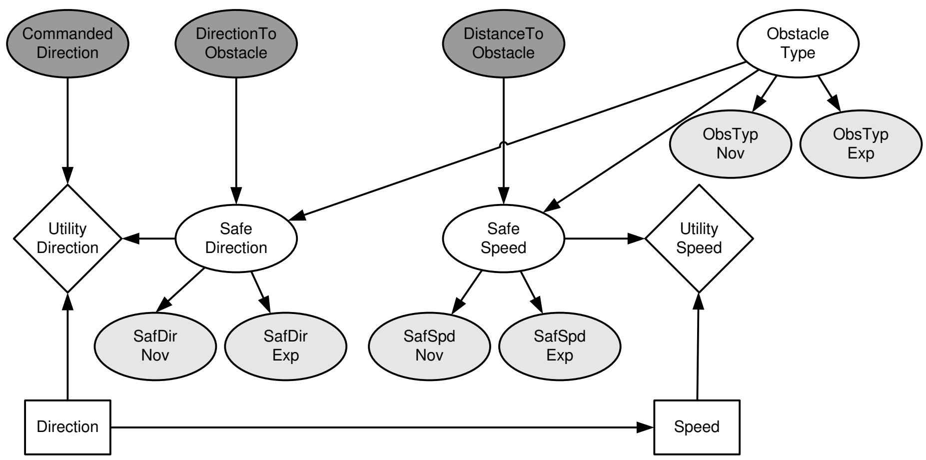


Figure 5.3: Influence Diagram (ID) representation for a navigation task. Ovals are chance nodes, squares are decision nodes, and diamonds are utility nodes. Grey nodes represent information sources: dark grey nodes are observed by the robot, light grey nodes are observed by the human.

Name	**Type**	**States**	**Discretisation**
CommandedDirection	robot-obs.	*left, straight, right*	$[-180,-10]; (-10,10); [10,180]deg$
DirectionToObstacle	robot-obs.	*left, straight, right*	$[-60,-10]; (-10,10); [10,60]deg$
DistanceToObstacle	robot-obs.	*very close, close, far*	$[0.0,0.2]; (0.2,1.5); [1.5,\infty)m$
ObstacleType	latent	*non-trav., trav.*	
ObsTypNov	human-obs.	*non-trav., trav.*	
ObsTypExp	human-obs.	*non-trav., trav.*	
SafeDirection	latent	*left, straight, right*	
SafDirNov	human-obs.	*left, straight, right*	
SafDirExp	human-obs.	*left, straight, right*	
SafeSpeed	latent	*stop, slow, fast*	
SafSpdNov	human-obs.	*stop, slow, fast*	
SafSpdExp	human-obs.	*stop, slow, fast*	
Speed	decision	*stop, slow, fast*	
Direction	decision	*left, straight, right*	
UtilityDirection	utility		
UtilitySpeed	utility		

Table 5.1: Details of the ID representation shown in Figure 5.3: chance nodes are listed first, followed by decision and utility nodes.

a representation as described in Chapter 2. It is not the focus of this work, and therefore both the structure and the parameters are handcrafted here. All nodes are discrete as summarised in Table 5.1 and therefore the probability distributions and utility functions are represented as tables.

At robot run-time, the model is instantiated at each time instance without taking priors into account, *i.e.* no filtering is performed and inference is static. Two decisions need to be made at each time instance in the following order[4]: in which direction to move (decision node *Direction*), and with what speed (decision node *Speed*). The following section discusses how these decisions are made.

5.2.1 Making Decisions

Robot-observable nodes are shown in dark grey in Figure 5.3. The robot is assumed to have two on-board sensors: a laser scanner and a camera. Information provided by the camera is not processed: reliable interpretation of images remains a difficult research problem [37]. Instead, obstacle states are extracted from the current laser scan serving as evidence for *DirectionToObstacle* and *DistanceToObstacle* representing the direction and distance to the closest obstacle (if any). The robot also receives a desired direction (*e.g.* towards a waypoint) which serves as evidence for the *CommandedDirection* node.

After having incorporated all evidence from robotic sensors, the first decision the robot has to make is which direction to move. Alternatively, the robot can query a human operator for more information first. Human-observable nodes are shown in light grey in Figure 5.3 and represent potential candidates for obtaining more information. The robot computes the VOI for all human-observable nodes using Equation 2.15. When a human observation is received, it is incorporated into the representation as additional evidence. Finally, a direction is selected using Equation 2.11. The procedure is then repeated for the *Speed* decision. Table 5.2 shows an example of how the two decisions can be mapped to actuator commands for a differential drive robot.

5.2.2 Human Sensor Models

Human-observable nodes are children of the *latent* (unobserved) variables *ObstacleType*, *SafeDirection*, and *SafeSpeed*. These nodes are considered higher-level in terms of abstraction and are easily understood by humans. *ObstacleType* encodes whether an obstacle is

[4]The order is set arbitrarily here and could be reversed. An ID requires a directed path between decision nodes (see Section 2.4).

Speed	**Turnrate** *Direction* *left*	*straight*	*right*	**LateralVelocity**
stop	$0deg/s$	$0deg/s$	$0deg/s$	$0m/s$
slow	$10deg/s$	$0deg/s$	$-10deg/s$	$0.4m/s$
fast	$20deg/s$	$0deg/s$	$-20deg/s$	$0.8m/s$

Table 5.2: Mapping from actions (*Speed* & *Direction*) to actuator commands (Turnrate & LateralVelocity) for a differential drive robot. The listed values were used in the experiments presented in Section 5.3.

pushable or not which is assumed to be a challenging task using robotic perception. *SafeDirection* and *SafeSpeed* encode the direction and the speed which are currently safest.

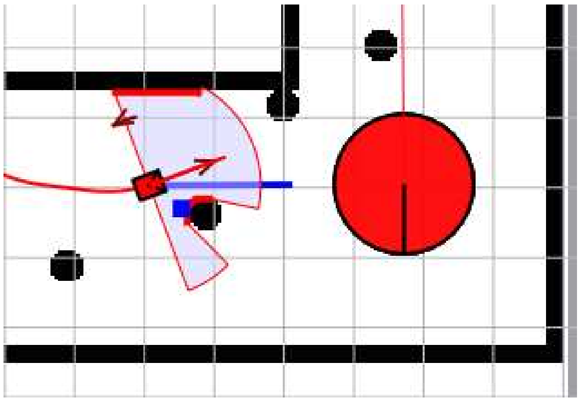
Two types of human-observable nodes (novice/expert) are used to represent the different levels of uncertainty in the operators' answers: in plain terms, how much the answers can be trusted. As in previous chapters, the CPDs of human-observable nodes are called HSMs. HSMs are important for the VOI analysis: the more an answer can be trusted, the more valuable it is.

An example is visualised in Figure 5.4: the robot is in a "critical" situation because there is an obstacle close to its right which is also the direction it is commanded to move (Figure 5.4(a)). The VOI analysis for this situation is pictured in Figure 5.4(b) showing the VOI for all six human-observable nodes. The figure shows that information from an expert is more valuable than from a novice. A more detailed VOI analysis is presented in Appendix B.1.

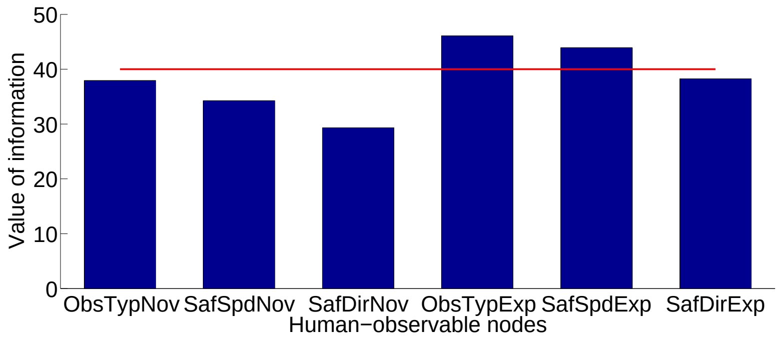
The horizontal line in Figure 5.4(b) shows the (arbitrarily set) cost of obtaining information. In this scenario, two bars exceed the cost: *ObsTypExp* and *SafSpdExp*, *i.e.* Equation 2.16 is fulfilled for these two information sources. Thus, the robot would consult an expert about the type of obstacle which is the most valuable observation it can possibly obtain for this scenario. Section 5.3.1 will present an experimental methodology for finding an appropriate cost parameter.

5.2.3 Sensor Information Type

The previous section discussed how the framework incorporates different levels of operator expertise. Operators specify their expertise when signing up with the RequestService



(a) Robot navigating in a maze



(b) VOI analysis

Figure 5.4: Robot in a "critical" situation: (a) it detected a close obstacle to its right which is also the direction it is supposed to move (blue marker points towards next waypoint, red arrows indicate current speed and turnrate); (b) VOI analysis for this situation showing the VOI for all human-observable nodes. The horizontal line represents the (arbitrarily set) cost of consulting a human operator.

as described previously in Section 5.1.2. The second QoS property specified at signup time is the available sensor information type. In our experiments, humans may have access to the laser scan and/or the camera image which are relayed to the operators using a high-frequency robot-push pattern. If only a laser scan is available (*e.g.* due to bandwidth constraints), the obstacle type (pushable or not) cannot be observed. For the scenario visualised in Figure 5.4, a question about the safe speed (*SafSpdExp* – second highest bar) would be relayed to the operator.

5.3 Experiments

Two experiments are conducted to evaluate cooperative human-robot decision making. The objectives of the first experiment are (a) to measure the overall *effectiveness* of the human-robot team as a function of the autonomy level[5], and (b) to find an appropriate autonomy level for a mission with given constraints, *e.g.* available communication bandwidth.

The results of the first experiment are used as a guide to set the autonomy level for the second experiment which is a user study. The user study has three objectives: (a) to compare the proposed system to conventional teleoperation with respect to performance, operator workload, usability, and users' perception of the robot, (b) to investigate the influence of expertise and available sensor information on performance, and (c) to demonstrate performance variability for teleoperation.

5.3.1 Measuring Team Effectiveness

This section presents a methodology to measure the effectiveness of a human-robot team as a function of the autonomy level. The ultimate objective is to find an appropriate autonomy level prior to the deployment of the team as part of a mission. The autonomy level acts as a design parameter which determines under what circumstances the system adjusts its autonomy online. The methodology is general and may be applicable to other human-robot systems in which robot performance can be measured.

Two metrics are used to measure the effectiveness of the human-robot team as a function of the autonomy level: the performance of the robot, and the number of operator queries required to achieve that performance. It is assumed that robot performance increases with more human input. In order to find an appropriate autonomy level, two competing goals need to be traded off: maximising robot performance while minimising human input.

The relative importance of the two goals is dictated by the priorities and the constraints of the mission. Examples for mission priorities are safety and efficiency, examples for mission constraints are the number of humans and robots available and communication bandwidth. To incorporate mission priorities and constraints, the two metrics mentioned above are weighted accordingly and summed up into a single scalar. The scalar can be interpreted as

[5]The term *autonomy level* was defined in Section 5.1.1

a mission-specific measure of human-robot team effectiveness. The measure can be used to find the appropriate autonomy level for a given mission.

The methodology described above is split into 5 steps which are subsequently applied to the navigation task presented in Section 5.2:

1. Identify robot performance metrics for a given task (here: navigation).
2. Experimentally obtain robot performance values and number of human-robot interactions as a function of the autonomy level.
3. Identify the constraints of the mission such as available resources (*e.g.* number of humans and robots, communication bandwidth), and mission priorities (*e.g.* safety, efficiency).
4. Combine robot performance metrics and number of human-robot interactions into a scalar called *team effectiveness* using a weighted sum. Weights are chosen based on the constraints found in the previous step.
5. Select the autonomy level with the highest team effectiveness value for the mission.

Performance Metrics (Step 1) Two robot performance metrics are used here: the number of successfully completed navigation tasks and the time to complete each (successful) task. The two metrics are proposed in [165] to measure effectiveness and efficiency of a navigation task.

Experiment (Step 2) To obtain a statistically relevant dataset, the following approach is taken. Human answers are simulated by running a software component which generates better informed observations than the robot is able to produce. This is achieved by restricting the robot's laser scan to a field of view of only 40° but allowing the simulated human access to a 120° scan.

We argue that this is a valid approach since humans can often extract more information from the same sensor data than robots by applying their rich perceptual capabilities. Humans can also apply background information not available to today's robots. For the purpose of this experiment, human and robot capabilities are artificially reduced.

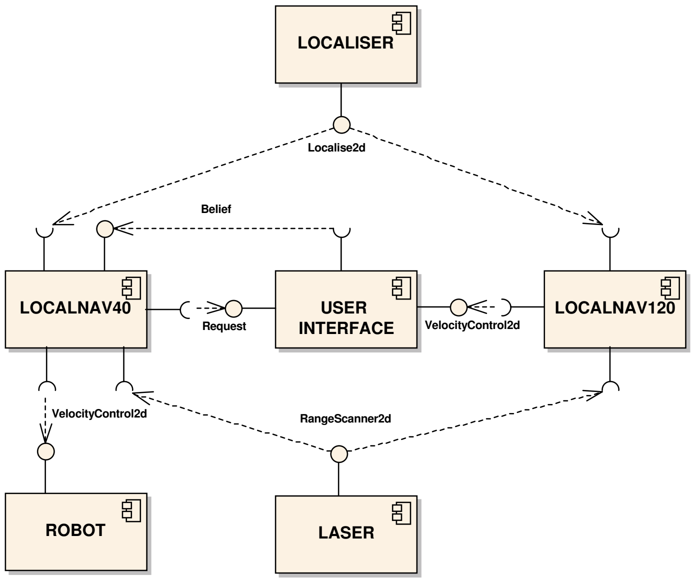


Figure 5.5: Software architecture of the simulated human experiment. Boxes are components with provided (circles) and required (semi-circles) interfaces. LocalNav40 and LocalNav120 are identical algorithmically: they both use the decision model shown in Figure 5.3 but differ in the amount of sensor information used to compute their decisions. The field of view of the laser scan is 40 and 120 degrees, respectively.

Figure 5.5 shows the software architecture used in this experiment. Aside from the different fields of view of the laser, LocalNav120 and LocalNav40 are identical navigation algorithms using the influence diagram representation as described in Section 5.2. LocalNav40 controls the robot based on a 40° laser scan plus the human observations entering through the `Belief` interface.

The simulated human acts as an expert who, whenever requested, is able to produce *SafDirExp* and *SafSpdExp* observations by using information from the LocalNav120 component. These two variables correspond to the 5^{th} and 6^{th} bar in Figure 5.4(b).

The experiment is conducted using 10 environments of size $64m \times 16m$ with randomly generated fixed obstacles as shown in Figure 5.6. The robot is placed near the centre on the far left of the world and a goal waypoint is generated near the centre on the far right (with some randomness). An episode is declared successful if the robot manages to position itself within $1m$ of the goal before a timeout of $120s$ occurs. Timeouts occur if the robot hits an obstacle and as a result gets stuck.

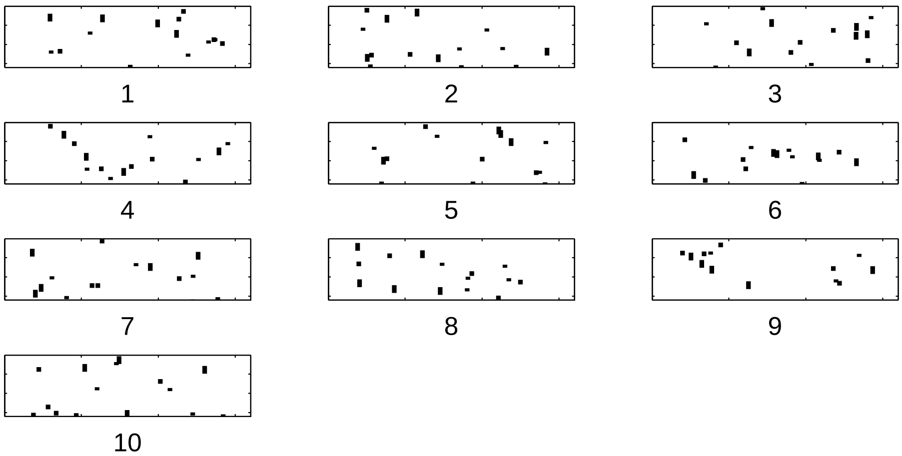


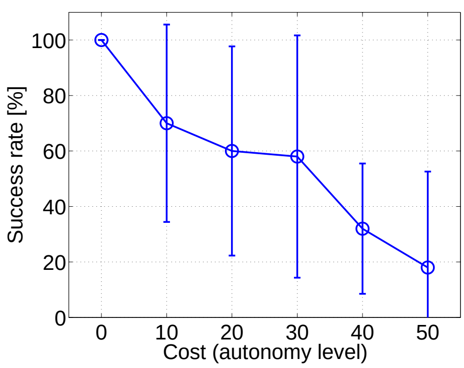
Figure 5.6: Ten environments of size $64m \times 16m$ with randomly generated obstacles.

The evaluation uses 6 cost parameters which are uniformly distributed over the range of possible VOI values. Each of the 10 environments is traversed 5 times resulting in a dataset of size 300. The experiment ran continuously for a total of 8.7 hours. The software components shown in Figure 5.2 were implemented and deployed using the Orca software framework [21].

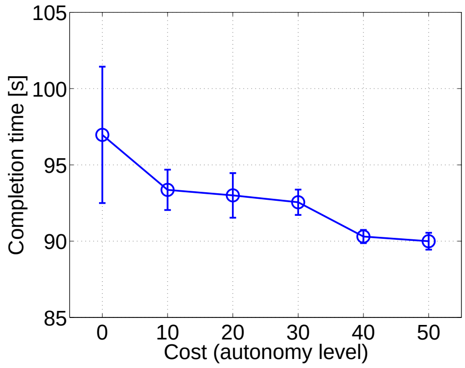
Results of the experiment are presented next. Figure 5.7 shows the success rate and the completion time as a function of the cost parameter. The success rate drops off with increasing autonomy (fewer queries). At cost 0, when the simulated human is queried continuously, all 50 episodes are successful. At cost 50 the robot is autonomous at all times (no queries) and completes the course successfully in only 18% of the cases, on average. To summarise, the plot shows that robot performance increases if input from an expert can be obtained.

The completion time is only measured for successful episodes, *i.e.* when the robot did not hit an obstacle. Completion time is highest for cost 0 when the simulated human is queried continuously. This is expected since the robot carefully avoids all obstacles. The high variance at cost 0 is also expected since more time is needed to navigate in more difficult environments.

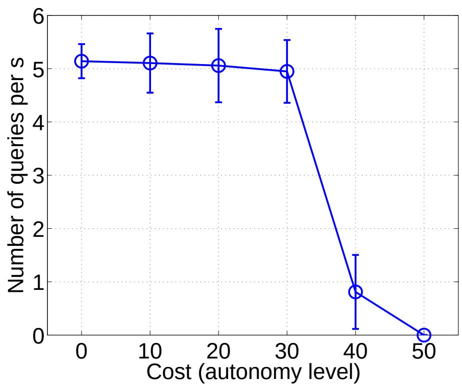
At higher autonomy, the robot takes less time to reach its goal which seems surprising at first. However, the number of successful episodes decreases as shown above, and only environments with few obstacles along the centre are traversed successfully as shown in



(a) Success rate



(b) Completion time



(c) Number of operator queries

Figure 5.7: Team effectiveness metrics ((a),(b) robot performance; (c) number of interactions) as functions of cost, obtained experimentally. Variance in results is due to the varying difficulty levels of the environments. Error bars indicate one standard deviation.

Fig 5.8. The robot is faster if it does not hit an obstacle because it steers "blindly" towards its goal without spending time to avoid obstacles.

Finally, Figure 5.7(c) shows the number of operator queries per second as a function of the cost. The number of queries decreases gently first, then drops off steeply and finally goes to 0 which represents fully autonomous operation. The steep drop-off can be explained by the coarse representation used in this experiment: obstacles are either far, close, or very close. Situations in which obstacles are far away are much more numerous than situations in which they are close. At cost 30, the robot queries the human even in relatively safe situations when obstacles are far resulting in many more queries than at cost 40.

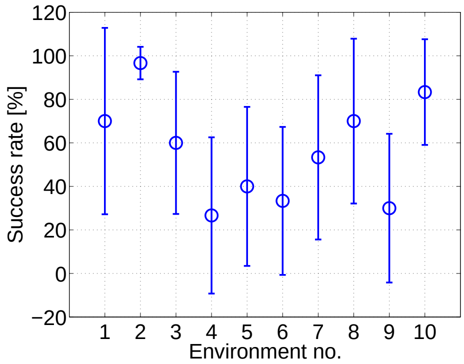


Figure 5.8: Success rates for the different environments. Environment 2 is particularly easy to traverse because there are few obstacles along the centre (compare to Figure 5.6). Variance in results is due to the varying cost parameter. Error bars indicate one standard deviation.

Even at high costs, the number of queries is too high for a practical deployment with real humans. Obstacle tracking and wall recognition algorithms can be used to avoid repeated questions about the same obstacle (see Section 5.3.2). This functionality was turned off to simplify the experiment without having an impact on the conclusions to be drawn. Using obstacle tracking, the shape of the curves are expected to be similar while the number of queries per second would be lower (*i.e.* only the scale of Figure 5.7(c) would change).

Identification of Constraints (Step 3) Three mission scenarios are presented as examples for step 3 of the proposed methodology:

1. *Multiple operators, robot is expensive*: this scenario reflects the remote operation of an expensive robot, *e.g.* in a search and rescue mission. We assume the robot's environment is inaccessible to humans, so operators cannot physically intervene. It is also assumed that many human operators can be contacted, experts as well as non-experts, *e.g.* through a web application[6].

2. *Single operator, multiple robots*: the second scenario assumes a single operator who is in charge of multiple robots at the same time. It is assumed that the operator can only attend to one robot at the time. The operator's capacity is the limiting factor here.

[6] Amazon's Mechanical Turk [188], which is a similar system to the one presented here, was used in September 2007 to search for a disappeared plane [189]. Within three days up to 50,000 people joined in the effort (without success).

3. *Multiple operators, robot is expensive, limited communication bandwidth*: this scenario is similar to scenario 1 with the additional assumption of a limited communication bandwidth. An example application is a remote planetary exploration mission.

Parameter Selection (Steps 4 & 5) Step 4 requires the combination of success rate, completion time, and the number of queries into a single scalar which is called *team effectiveness*. The combination is achieved by a weighted sum which is subsequently scaled to a $[0;1]$ interval. The weights represent the relative importance of each variable and are set according to the mission priorities and constraints identified in step 3.

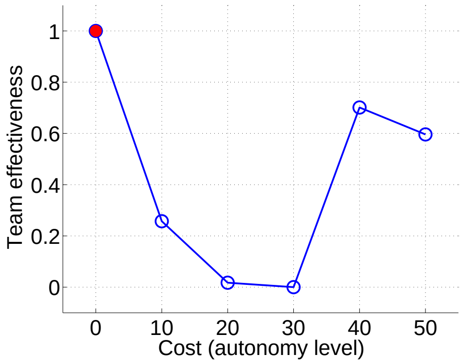
For scenario 1, the completion time is of secondary importance but the robot could be lost if it hits an obstacle. A weight ratio of $3:1:1$ is manually chosen emphasising the importance of success over completion time and number of queries. Figure 5.9(a) shows the team effectiveness for all cost parameters. The highest team effectiveness is achieved by setting the cost parameter to 0.

For scenario 2, the weight ratio is chosen to be $1:1:3$, reflecting the importance of minimising the number of queries and thus the operator workload. The cost parameter with highest team effectiveness is 50 for this case as shown in Figure 5.9(b). Selecting this parameter implies the acceptance of a performance loss.

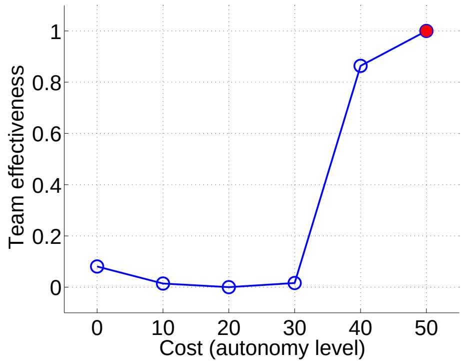
For scenario 3, the weight ratio is $3:1:1.5$ which is similar to scenario 1 but penalises human queries more to avoid excessive communication. Figure 5.9(c) shows the result: the highest team effectiveness is achieved when cost 40 is selected.

Conclusion The experiment demonstrates that robot performance increases with human input, *i.e.* the less autonomy the better the performance. The autonomy level is a design parameter which determines under what circumstances the robot adjusts its autonomy online. In our system, the autonomy level is represented by the cost of obtaining information from the human operator. A methodology has been followed to find an appropriate cost parameter based on quantitative results for robot performance and the number of operator queries.

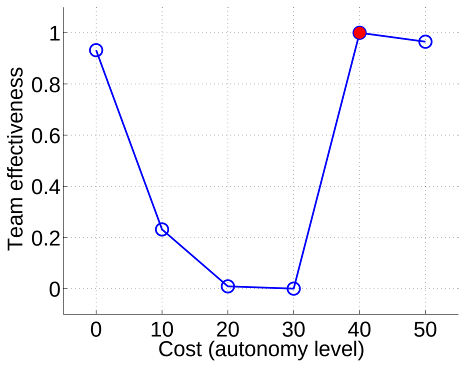
The results from this experiment act as a guide to set the cost parameter for the user study experiment which is presented next. The scenario for the user study is to let a single human



(a) Weight ratio 3 : 1 : 1



(b) Weight ratio 1 : 1 : 3



(c) Weight ratio 3 : 1 : 1.5

Figure 5.9: Team effectiveness graphs generated by weighted sums of 3 variables (success rate, completion time, number of queries). Team effectiveness values are scaled to [0; 1]. The most appropriate autonomy level is found by selecting the cost where team effectiveness is 1 (indicated by the red markers). Three mission scenarios are shown: (a) multiple operators, robot is expensive; (b) single operator, multiple robots; (c) similar to (a) but penalising excessive communication.

interact with a single remote robot. However, the team effectiveness graphs presented above cannot be applied directly because there are two differences in the experimental setup: (1) the robot uses the full 120° scan in the user study, and (2) besides speed and direction, humans are also queried to specify the obstacle type.

5.3.2 User Study

A user study was conducted to compare the robot-pull system to manual teleoperation (simple human-push). Teleoperation is a well-understood research area and especially useful in unstructured environments, *e.g.* in search and rescue scenarios [91]. Well-known limitations of teleoperation are high operator workload and scalability [83]. Other problems of teleoperation are the dependence on the user's skills and the sensor information available to the user. Typically, users need to be trained experts to be capable of teleoperating a robot. In contrast, the robot-pull system poses simple questions that can be answered using common sense.

Based on this discussion, three objectives were identified for this user study: (a) to compare the proposed system to conventional teleoperation with respect to performance, operator workload, usability, and users' perception of the robot, (b) to investigate the influence of expertise and available sensor information on performance, and (c) to demonstrate performance variability for teleoperation.

For the user study, the cost parameter was fixed *a priori* based on the results of Section 5.3.1. Wall and obstacle tracking was turned on to avoid repeated queries related to the same obstacle. To simplify technical terms for participants, the robot-pull system was referred to as "dialog" here.

Experimental Design Two groups of participants, experts and novices, were employed for the experiment: 22 participants (13 male, 9 female) with an average age of 30.7 (oldest 48, youngest 23) were recruited. The expert group consisted of 10 post-graduate or post-doc researchers from our robotics lab. None of them was familiar with the dialog system or had any prior information concerning the purpose of the study.

The novice group consisted of 12 participants who had no prior experience with mobile robots. All but one of the participants in the novice group were holding a university degree while none had an engineering background. Privacy was maintained by using anonymous user names for each participant.

The task was to navigate a robot through the maze shown in Figure 5.10. Participants received written instructions "to get the robot to its goal as quickly as possible while making sure it is safe". The full set of instructions can be found in Appendix B.3.

The experiment followed a within-subjects design: all participants operated the robot using four different modes which represent the experimental conditions. The modes were generated by varying *driving style* and *information type* which are explained next.

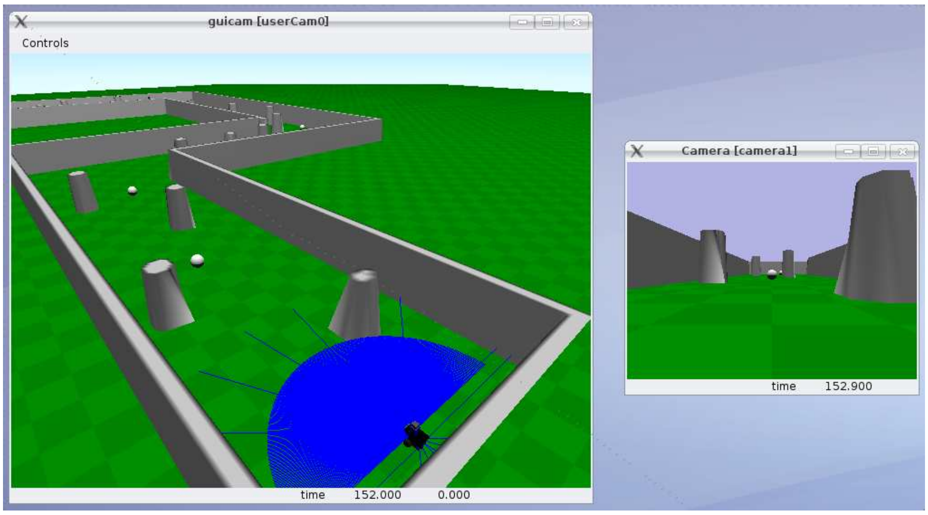


Figure 5.10: Left: global view of the maze in 3d including walls, columns, and spheres. Robot is equipped with an onboard laser (scan is shown in blue) and a camera. Right: robot's camera image.

Available driving styles were teleoperation and dialog. In teleoperation, participants operated the robot manually using a joystick, while the robot acted semi-autonomously in the dialog modes as described in Section 5.2. In the dialog modes, the interactions between operator and robot were limited to an occasional query posed by the robot[7]. If no answer had been received within 30 seconds, the robot made a decision without human input.

Information type relates to the sensor information relayed to the operator: either the laser scan only, or the laser scan and a camera image (participants did not have access to the global view).

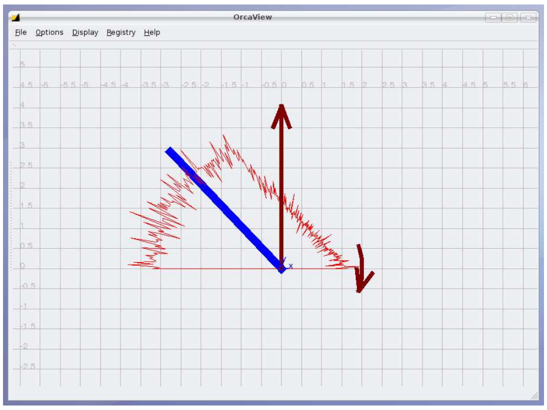
The four modes are summarised in Table 5.3 with the corresponding user interfaces shown in Figure 5.11. The red line is the robot's noisy laser scan, the solid blue bar indicates the direction to the next waypoint in the robot's coordinate frame, and the red arrows give feedback about the current speed and turnrate, respectively. In the dialog modes (second column), the robot poses questions which show up as multiple-choice dialog boxes.

In order to measure workload, a secondary task was introduced which is common practice in human-machine interaction studies [35][30]. Participants were asked to add 2-digit numbers if and only if they had spare capacity to do so. A Graphical User Interface (GUI) was

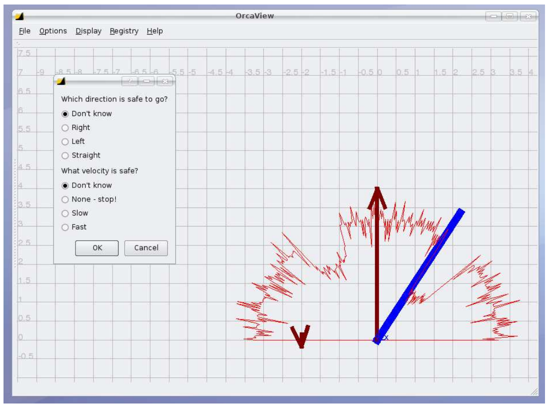
[7]Appendix B.1 presents a VOI analysis for a typical robot path through the maze

Information type	**Driving style**	
	Teleoperation	Dialog
Laser	teleop/laser	dialog/laser
Laser & camera	teleop/camera	dialog/camera

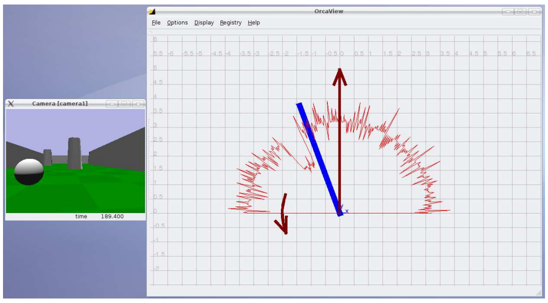
Table 5.3: The four experimental modes.



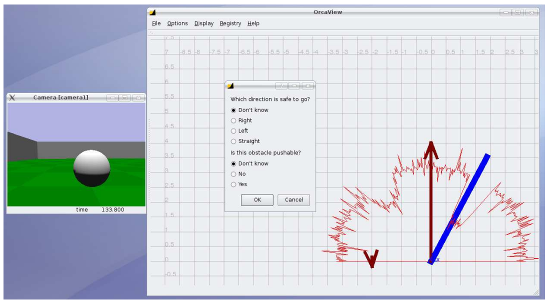
(a) Teleop/laser



(b) Dialog/laser



(c) Teleop/camera



(d) Dialog/camera

Figure 5.11: User interfaces for the four experimental modes.

designed displaying four multiple choice answers as shown in Figure 5.12. Answers could be entered using the buttons on the joystick (teleoperation modes) or by using the mouse (dialog modes).

The experiment was conducted as follows. First, participants completed a pre-experiment questionnaire designed to gather demographic data and assess the participants' driving skills. After reading a set of instructions, teleoperation of the robot in the maze was practised for about 1 minute. After that, the experiment started with one of the four modes. To account for learning effects, the order of testing was counterbalanced.

Figure 5.12: GUI to measure workload (secondary task).

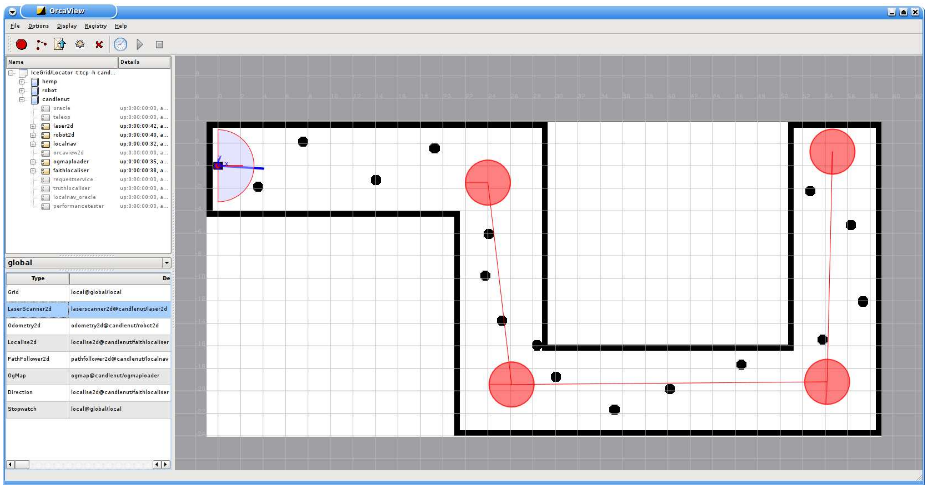


Figure 5.13: Global view of the maze in 2d: occupancy grid map of the columns and walls, robot with its laser scan, and waypoints (red circles).

The experimenter, who had access to the global view on a separate computer screen (Figure 5.13), monitored the progress of the experiment and noted the number of collisions. Figure 5.14 shows the physical setup of the user study.

After each task was completed, participants were asked to fill in a post-task questionnaire consisting of 7-point Likert scale test items (13 for teleoperation, 17 for dialog). After all four tasks were completed, a post-experiment questionnaire was filled in, containing six multiple choice questions and three free text sections. The overall procedure took the average participant about 45 minutes.

All three questionnaires can be found in Appendix B.3.

Figure 5.14: The physical setup of the user study: participants controlled the robot in simulation using one of the interfaces shown in Figure 5.11, while the experimenter had access to the global view shown in Figure 5.13.

Measures Measures used in this study were performance, workload, usability, and the users' perception of the robot. Measures of secondary importance were related to the dialog modes only and are reported in Appendix B.2.

Performance was measured using two variables: number of collisions and time to complete the task. For teleoperation, a collision was registered whenever the robot ran into an object (in teleop/camera spheres did not count because they had been declared pushable). For the dialog modes, collisions were situations in which the robot could not keep driving by itself and needed to be freed by the experimenter. These situations occurred if participants answered inappropriately. In dialog/laser, collisions with spheres were also counted. Completion time was the total time it took to reach the last waypoint.

Workload was measured by two means: (1) the number of maths problems correctly solved, and (2) through introspection using the post-task questionnaire. Four statements were used for the latter derived from the NASA Task Load IndeX (TLX) method [75]. Usability was measured using five statements in the post-task questionnaire which were adapted to a navigation task from [19].

Remaining measures were related to the users' perception of their relationship to the robot in the different modes, namely safety, trust, intelligence, and partnership. They were assessed using the questionnaires.

Hypotheses Based on the objectives listed in the introduction to Sec. 5.3.2, the following experimental hypotheses were tested:

1. Performance-related hypotheses:
   (a) Dialog modes will be superior to teleoperation for both performance measures (number of collision, completion time).
   (b) Having access to camera images will result in better performance than having access to a laser scan only.
   (c) Experts will perform better than novices.
   (d) Performance will be more variable in teleoperation than in dialog and will depend on the available information type (laser/camera) and expertise (novice/expert).
   (e) There is a positive correlation between teleoperation performance and users' self-reported driving skills.
2. Workload will be higher in teleoperation than in dialog.
3. Dialog modes will be more usable than teleoperation.
4. The robot will be perceived as safer, more trusted, more intelligent, and more peer-like using dialog modes.

Results The experiment used a $2x2x2$ factorial design with two within-subjects variables (driving style, information type) and one between-subjects variable (expertise) [82]. The data was analysed by applying a three-way repeated-measures ANOVA which enables testing for the overall main effects of the three variables and their interactions. The level of statistical significance was set at $p = .05$. Three conventional levels are used to report results: $\{.05, .01, .001\}$. If measures had no physical unit, they were scaled to a range of 0.0 to 1.0.

Measure	**Main effects** **ds**	**it**	**ex**	**Interactions** **ds × it**	**ds × ex**
collisions	●	●	○	○	○
completion time	○	●	●	●	●
workload I	●				
workload II	●				
usability	●				
perception of robot	●				

Table 5.4: Summary of the statistical analysis showing all effects of interest. They were tested for significance using a three-way repeated-measures ANOVA. A filled circle represents a statistically significant effect, an empty circle means no significance was found. Independent variables were ds: driving style, it: information type, ex: expertise.

Measure	**Main effects**						**Interactions**			
	ds		**it**		**ex**		**ds × it**		**ds × ex**	
	F	p	F	p	F	p	F	p	F	p
collisions	11.7	$**$	6.9	$*$	1.1	$n.s.$	2.8	$n.s.$	3.0	$n.s.$
completion time	0.3	$n.s.$	11.3	$**$	13.1	$**$	18.1	$***$	7.3	$*$
workload I	63.8	$***$								
workload II	8.2	$**$								
usability	5.5	$*$								
perception of robot	20.3	$***$								

Table 5.5: Details of the statistical analysis (see Table 5.4): all statistically significant results are shown using $F[1, 20]$. Three conventional significance levels are applied here: $p < \{.05, .01, .001\}$ represented by $\{*, **, ***\}$ ($n.s.$ means not significant). Independent variables were ds: driving style, it: information type, ex: expertise.

Results of the statistical analysis relevant to the hypotheses listed above are summarised in Tables 5.4 & 5.5. A mapping from the statistical tests to the hypotheses and figures is given in Tables 5.6 & 5.7. A discussion of the results is presented below.

Results from the performance measures are presented first. Figure 5.15(a) shows the absolute numbers of collisions for all participants in the four different modes. Most collisions occurred in teleoperation, particularly teleop/laser which accumulated twice as many as teleop/camera. As shown in Table 5.5 and visualised in Figure 5.15(b), dialog was a significantly safer mode than teleoperation supporting hypothesis 1(a). Having a camera available also resulted in significantly less collisions when compared to having access to a laser scan only supporting hypothesis 1(b). No significant overall difference in the number of collisions between experts and novice could be found. As shown in Figure 5.15(d), performance

Measure	**Main effects**			**Interactions**	
	ds	**it**	**ex**	**ds × it**	**ds × ex**
collisions	1(a)	1(b)	1(c)	1(d)	1(d)
completion time	1(a)	1(b)	1(c)	1(d)	1(d)
workload I	2				
workload II	2				
usability	3				
perception of robot	4				

Table 5.6: Mapping of hypotheses to statistical analysis. Independent variables were ds: driving style, it: information type, ex: expertise.

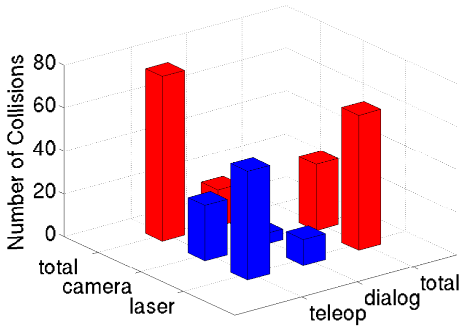
Measure	**Main effects**			**Interactions**	
	ds	**it**	**ex**	**ds × it**	**ds × ex**
collisions	5.15(a)&(b)	5.15(a)&(b)	5.15(a)&(b)	5.15(d)	5.15(e)
completion time	5.16(a)	5.16(a)	5.16(a)	5.16(c)	5.16(d)
workload I	5.17(a)				
workload II	5.17(b)				
usability	5.17(c)				
perception of robot	5.18(a)–(d)				

Table 5.7: Mapping of plots to statistical analysis. Independent variables were ds: driving style, it: information type, ex: expertise.

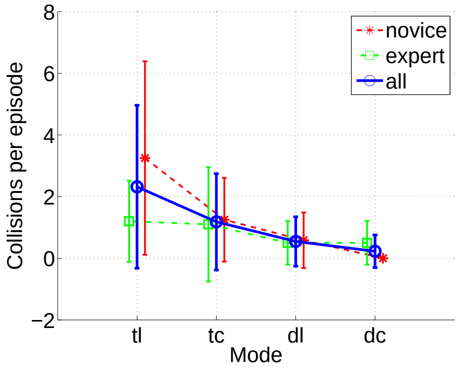
differences between the two information types are greater for teleoperation than for dialog. The same is true for the two expertise levels as shown in in Figure 5.15(e). However, these interaction effects were not statistically significant.

Completion time results are shown in Figure 5.16. Overall, completion times for teleoperation and dialog were nearly identical contradicting hypothesis 1(a). Having access to a camera yielded significantly faster completion times than having access to a laser scan only supporting hypothesis 1(b). Experts completed the task significantly faster than novices. As shown in Figures 5.16(c) & (d), completion time differed more in teleoperation than in dialog modes. Significant interactions were found for both expertise and information type, *i.e.* completion time is dependent on both variables. This finding supports hypothesis 1(d).

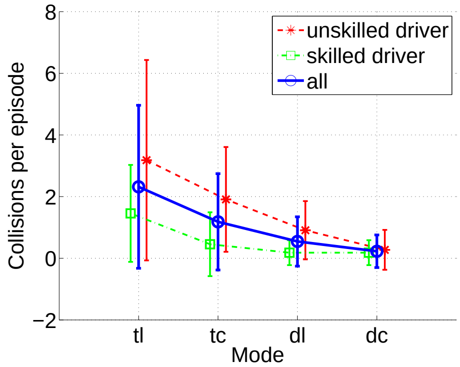
Hypothesis 1(e) is investigated next. As presented above, experts were significantly faster than novices in completing the task but no significant difference between experts and novices was found for the number of collisions. For teleoperation, performance may depend on factors such as hand-eye coordination and driving experience as predicted by hypothesis 1(e). To investigate the relationship between users' driving skills and their teleoperation



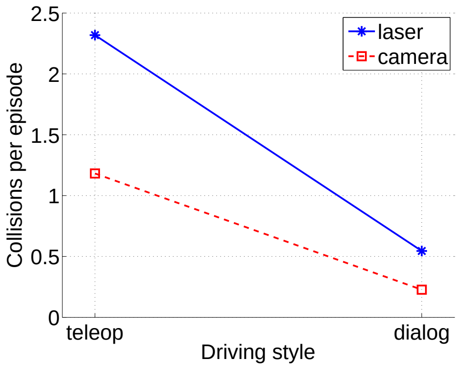
(a) Total number of collisions



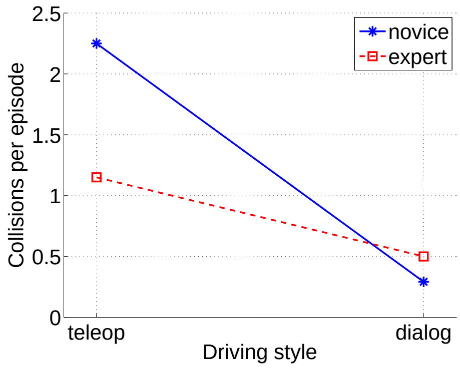
(b) Novice vs. expert



(c) Skill-based separation



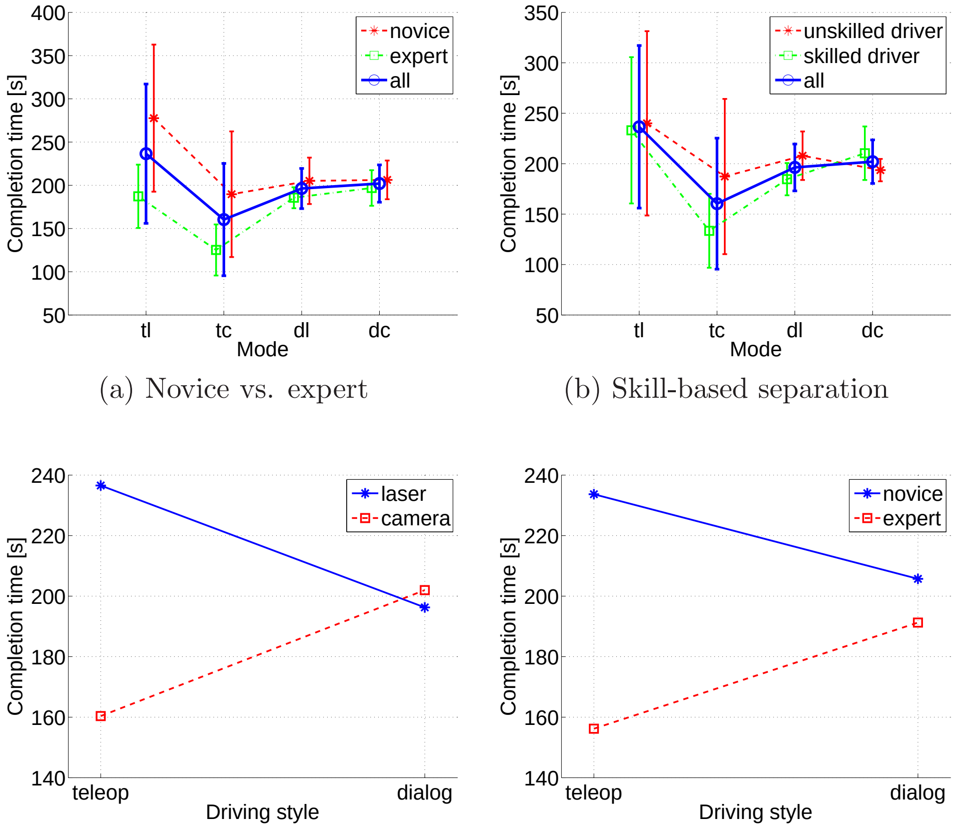
(d) Interaction: ds × it



(e) Interaction: ds × ex

Figure 5.15: User study results for first performance measure (number of collisions). Modes are tl: teleop/laser, tc: teleop/camera, dl: dialog/laser, dc: dialog/camera. Independent variables were ds: driving style, it: information type, ex: expertise.

performance, a correlational analysis was conducted. Driving skills were assessed using data from the pre-experiment questionnaire: experience in car driving, computer use, video games, and remote driving as well as hand-eye coordination and navigation skills (see Ap-



(a) Novice vs. expert (b) Skill-based separation

(c) Interaction: ds × it (d) Interaction: ds × ex

Figure 5.16: User study results for second performance measure (completion time). Modes were tl: teleop/laser, tc: teleop/camera, dl: dialog/laser, dc: dialog/camera. Independent variables were ds: driving style, it: information type, ex: expertise.

pendix B.3). The data was scaled and a weighted sum was computed with a weight ratio of 1 : 1 : 2 : 2 : 10 : 10 with respect to the driving skill factors yielding a single score. A similar method was used to obtain a single score for teleoperation performance: the number of collisions and completion times were scaled and summed. A Spearman correlation was computed yielding a statistically significant positive correlation between driving skills and teleoperation performance confirming hypothesis 1(e) ($r = +.58; n = 22; p < .01$). As a comparison, the correlation between expertise and teleoperation was also significant but slightly smaller ($r = +.53; n = 22; p < .01$).

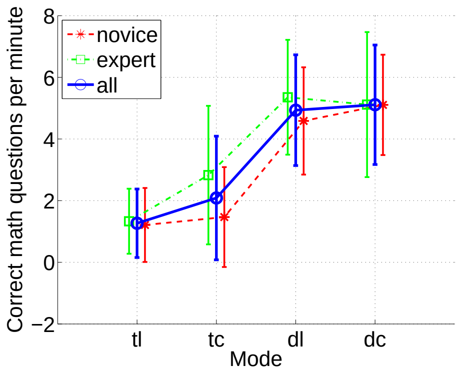
Figures 5.15(c) & 5.16(b) show the performance plots when participants were grouped into unskilled and skilled drivers rather than novices and experts. The median of the driving skill score was used to divide participants into two equal-sized groups. A greater difference between participants across all modes is evident in Figure 5.15(c) when compared to Figure 5.15(b) where the groups were separated by expertise. The same is not true for completion times where driving skill level did little to differentiate performance (Figure 5.16(b)) compared to the novice/expert division (Figure 5.16(a)).

Results for the workload measures are presented next. Figure 5.17(a) shows the number of correctly solved maths questions per minute which is a measure of spare capacity, and thus workload. Workload was significantly higher for teleoperation modes than for dialog modes as shown in Table 5.5. The difference between teleop/laser and teleop/camera was small for the novice group. This seems surprising given that the camera view provides a larger sensor range than the laser. However, as indicated by seven participants in the free text section of the post-experiment questionnaire, having the camera and the laser led to information overload. Experts seemed to cope better with the overload than novices. The slight decrease in number of correct answers from experts in dialog/camera can be explained as follows: after answering the robot's question, experts would often watch the robot's actions as a result of their answers rather than going back to solving math problems (observed subjectively by the experimenter).

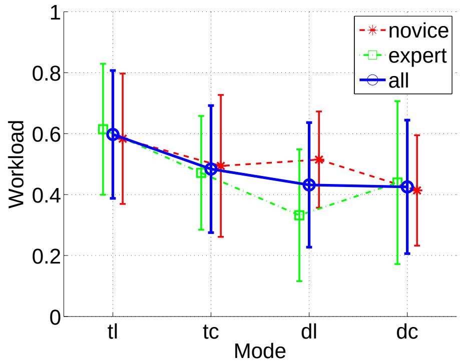
An alternative workload measure was obtained through introspection using four Likert-scale statements which were summarised into a single score (Figure 5.17(b)). The results agreed with above: workload was perceived higher for teleoperation than for dialog as shown in Table 5.5. Based on these results, hypothesis 2 was validated.

Similar to workload, usability was measured by averaging results of five Likert-scale statements. Dialog modes were perceived as more usable than teleoperation modes as shown in Figure 5.17(c) and Table 5.5. Hypothesis 3 was therefore also validated.

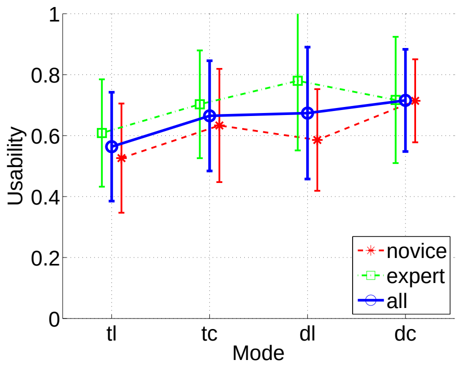
The users' perception of the robot with respect to safety, trust, intelligence, and partnership were measured using single Likert statements in the post-task questionnaire (see Figure 5.18). As before, the four measures were summarised into a single score for the statistical analysis. The robot was perceived significantly safer, more intelligent, more trustworthy and peer-like in the dialog modes compared to the teleoperation modes. Three out of the four measures were also evaluated using the post-experiment questionnaire (multiple choice, nominal data). Figure 5.19 shows the relevant plots while Table 5.8 summarises the statistical results using a Sign test to compare teleoperation to dialog. Based on these results, hypothesis 4 was validated.



(a) Spare capacity: secondary task



(b) Workload: self-reported



(c) Usability: self-reported

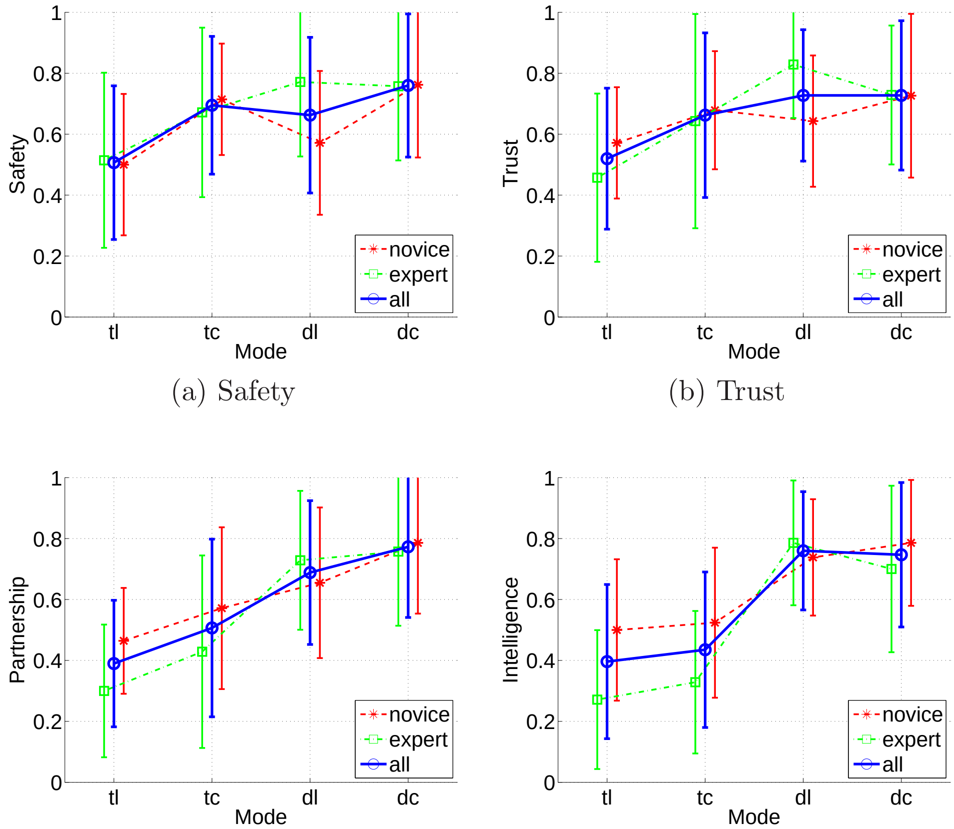
Figure 5.17: User study results for workload and usability. Modes were tl: teleop/laser, tc: teleop/camera, dl: dialog/laser, dc: dialog/camera.

Measure	**Post-experiment (Sign test)**
Safety	$N = 22; x = 6; p < .05$
Trust	$N = 22; x = 6; p < .05$
Intelligence	$N = 22; x = 1; p < .001$

Table 5.8: Users' perception of robot evaluated using data from the post-experiment questionnaire. A one-tailed Sign test was applied.

Discussion The experimental results show that performance using classical teleoperation is highly dependent on the operators' expertise, their driving skills, and the available information type. Similar results are reported in [117]. In contrast, performance results are more consistent when using the dialog system to operate the robot.

More collisions occurred in teleoperation than in dialog modes. However, the experiment also showed that completion time varies with expertise and the information type available



(a) Safety (b) Trust

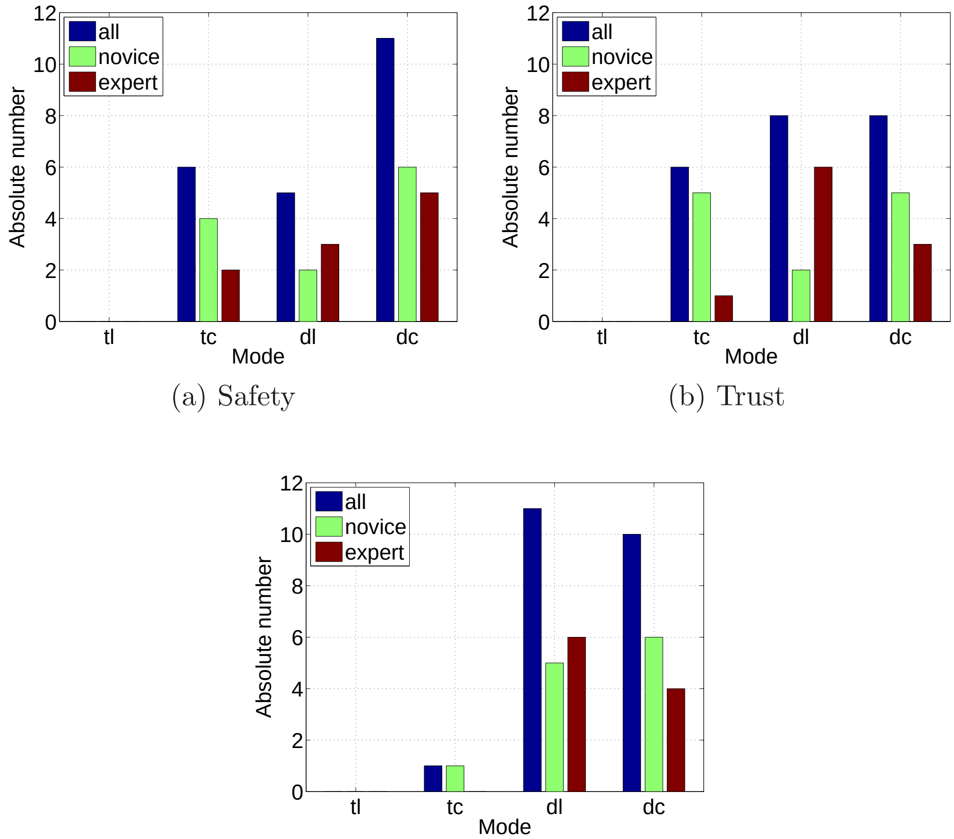
(c) Partnership (d) Intelligence

Figure 5.18: User study results for users' perception of the robot. Modes were tl: teleop/laser, tc: teleop/camera, dl: dialog/laser, dc: dialog/camera.

to the operator. For example, the average expert completed the course 35s faster using teleoperation than using dialog. Thus, a tradeoff exists between safety and timeliness when an appropriate driving mode needs to be selected. A procedure to find an appropriate driving mode is presented next.

The two performance metrics (safety and timeliness) can be combined into a scalar utility using a weighted sum. Figure 5.20 shows plots of the utility function for different weight ratios which reflect the relative importance of timeliness and safety. Utility is scaled from 0 to 1. The most appropriate driving mode can be found by selecting the mode where the utility is 1.

If safety and timeliness are equally important, the preferred driving mode is dialog/camera for novices and teleop/camera for experts as shown in Figure 5.20(a). If safety is twice as



(a) Safety

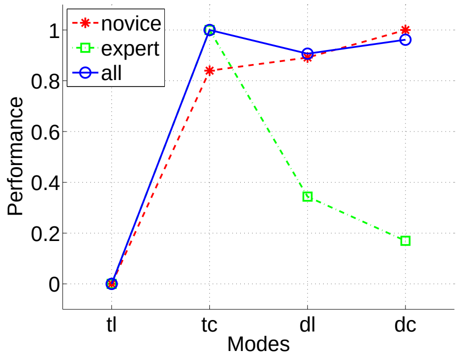
(b) Trust

(c) Intelligence

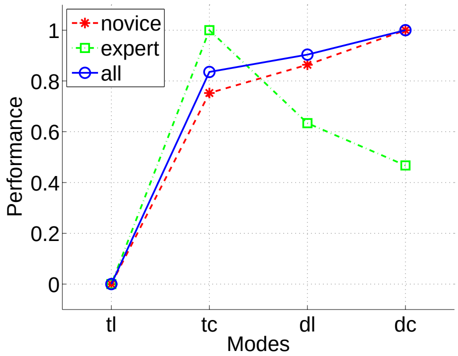
Figure 5.19: Results from the post-experiment questionnaire. Modes were tl: teleop/laser, tc: teleop/camera, dl: dialog/laser, dc: dialog/camera.

important as timeliness, the results are the same (Figure 5.20(b)). However, if safety is 100 times more important, an expert should switch to dialog/laser as shown in Figure 5.20(c). If timeliness is twice as important as safety, novices should use dialog/camera whereas experts should use teleop/camera as shown in Figure 5.20(d). If collisions do not matter (Figure 5.20(e)), even the novice should use teleop/camera.

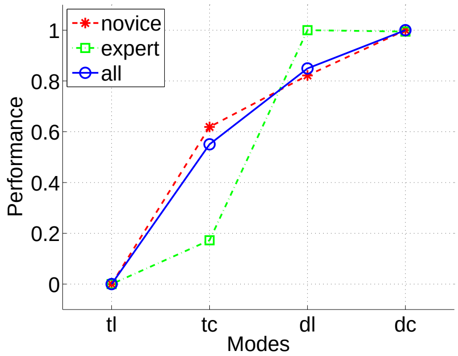
Workload results are discussed next. Operator workload was significantly lower for dialog modes than for teleoperation. This result has consequences for the scalability of a system: one human can operate multiple robots simultaneously. The allocation between robots and a human does not have to be fixed using dialog. It is possible to shift responsibilities dynamically in teams involving multiple robots and multiple humans. An experimental



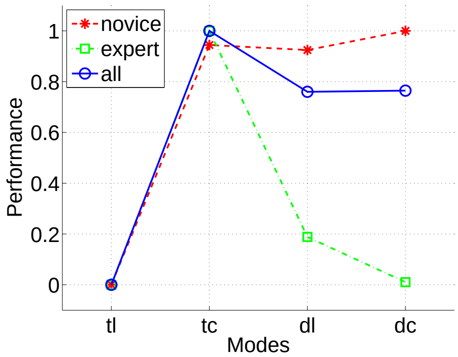
(a) Weight ratio 1 : 1



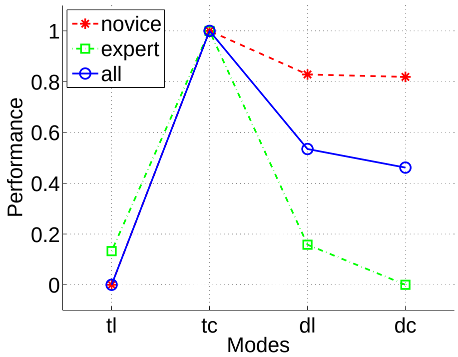
(b) Weight ratio 1 : 2



(c) Weight ratio 1 : 100



(d) Weight ratio 2 : 1



(e) Weight ratio 100 : 1

Figure 5.20: Utility functions to choose an appropriate driving mode. Weights reflect the relative importance of time and safety: (a) equally important; (b)–(c) safety more important; (d)–(e) time more important. The driving modes are tl: teleop/laser, tc: teleop/camera, dl: dialog/laser, dc: dialog/camera.

demonstration of many-to-many interactions is part of future work as discussed in Section 6.2.2.

Finally, usability and the users' perception of the robot are discussed. Using dialog instead of teleoperation to drive a robot was perceived to be a more usable interaction mode. Users attributed properties such as safety, trust, and intelligence to the robot while also acknowledging the robot as a peer. These factors may contribute to more effective human-robot cooperation in human-robot team settings.

Conclusion Based on the discussion above, the following conclusions are reached. The proposed dialog system has four advantages over teleoperation for a navigation task:

- Task performance does not depend as much on the background and driving skills of the human operator.
- Task performance does not depend as much on the type of sensor information available to the operator.
- Low workload leads to the potential of scaling up to large human-robot teams.
- Dialog modes are more usable and more appropriate for peer-to-peer human-robot interaction.

Despite these advantages, teleoperation may still be a preferable driving mode for a given mission. The dependency between mission priorities such as safety and timeliness and the appropriate driving mode was demonstrated above.

5.4 Related Work

5.4.1 Control Fusion

The material presented in this chapter can be seen as a particular solution to the general *control fusion* problem [64]: how to combine input from multiple sources to control a common resource. For the case where the common resource is a mobile robot, several *behaviour-based* architectures have been proposed such as Subsumption [22], Motor Schemas [4], and

DAMN [145]. The integration of human input into these systems can be achieved by letting operators add or alter behaviours for instance [3]. Other methods include averaging over motion vectors [64], and employing voting methods [65][63].

In contrast to the behaviour-based methods mentioned above, this work employs a purely deliberative architecture. Rather than arbitrating multiple commands, perceptual input from humans and robots is fused using a probabilistic model, and decisions are made according to the model's beliefs and utility functions.

5.4.2 Comparison to Collaborative Control

Work presented in this chapter is closely related to Fong's *Collaborative Control* which, like our approach, treats humans as a resource to robots [53]. Bidirectional communication in the form of human-robot dialog is used to exchange information of different types such as commands, queries and responses. He classifies his system as a form of teleoperation which compensates for the limitations of the conventional approaches by integrating human expertise, typically into the control loop. However, it is also anticipated that operators can close perception loops, cognition loops or combinations of the above. The systems has been experimentally evaluated with user studies for both single robot [55] and multi-robot scenarios [54].

There are many parallels to our view of cooperative decision making: robots are treated as peers, and perception loops are closed by posing queries. The main difference is the chosen approach: while Fong uses no specific underlying mathematical method, the problem of collaborative control is cast in a decision-theoretic formulation here. The probabilistic representation lies at the heart of the approach from which many of the issues of human-robot communication can be derived. This approach is contrasted with Fong's below.

Dialog management The type of queries, how to trigger them, who and in what order to ask, are well-defined using a decision-theoretic representation. Fong, in contrast, uses a number of hand-crafted questions. While both approaches require a designer, it can be argued that a decision-theoretic method is more general. The problem of finding a suitable representation remains; however, existing robotics representations can be reused by extending them with human-observable states.

Autonomy A decision-theoretic framework yields a decision at any given time, *i.e.* the system is always capable to operate fully autonomously by choosing the best action. This *self-reliance* is a prerequisite for building collaborative control systems as Fong points out in [53]: if no answer is available the robot should proceed regardless.

Authority Relationship Since everything about the world and the platforms is encoded in the underlying representation, there is no explicit distinction between humans and robots in our framework. This is an interesting outcome of the formulation which emphasises the concept of humans and robots acting as *peers*. This criteria does not apply to Fong's work since no underlying mathematical model is used.

Operator Uncertainty Similar to us, Fong employs stereotype user profiles, namely novice, scientist, and expert. Each profile has a fixed level of three user attributes: response accuracy, expertise level, and query interval. The approach of this work is to use probabilistic HSMs to encode accuracy and expertise, and thus treat human input the same way as robotic sensor readings. The benefits of this approach were presented in detail in Section 3.2.

Human Attention and Delays One of the results of Fong's user studies is that "*human assistance is a limited resource that must be carefully managed*" [55]. His query arbitration mechanism ranks queries with respect to priority and expiration. Expiration times are included in the transmitted messages which the query manager uses to sort the queue. In the approach presented here, the cost for human attention is handled explicitly by VOI theory. While time-outs are handled in a similar way to Fong, expected time delays are not taken into consideration yet. Time delays can be incorporated into the VOI calculation as discussed in Section 6.2.2.

Human-Pull Fong also considers questions originating from operators and addressed to the robot. He hand-crafted a number of questions which operators did not use often during the user studies [55][54]. As is the case for robot-pull, the type of questions to ask is well-defined in our system because all relevant information is encoded in the probabilistic representation. Operators can query the robot's belief at any time for monitoring purposes. Beliefs include uncertainty and relaying this information to operators can contribute significantly to enhance Situation Awareness (SA) [49].

System Evaluation Fong evaluates his system from an engineering point of view focussing on qualitative measures to improve system design and usability [55]. The quantitative evaluation method presented in Section 5.3.1 aimed at measuring team effectiveness to find an appropriate autonomy level. The proposed method is only applicable if a mathematical representation of the autonomy level exists – the cost of obtaining evidence in our case.

5.4.3 Adjustable Autonomy

The fields of Adjustable Autonomy (AA) and Mixed Initiative Control aim at bridging the gap between full human control and full autonomy [67][23][117][150]. The fundamental questions in these fields are how to decide when to relinquish control and on what criteria to base that decision [149]. In many AA systems, the human operators are in charge of switching between a set of predefined discrete modes which imposes a significant responsibility on the operator as pointed out in [24]. In contrast, the approach advocated here is to let the robot decide when to query operators for input based on the uncertainty in the robot's beliefs. This can be seen as a robot-initiated shift to lower autonomy at run-time. How often this shift occurs depends on the previously set autonomy level which acts as a design parameter.

Similar to us, Gunderson *et al.* believe that the degree of autonomy should be based on the amount of *uncertainty* which includes sensor inaccuracies, action failures and exogenous events [74]. They show in simulation that these three types of uncertainties have functional correlations to plan success. If a plan cannot be completed successfully, the agent needs to use external resources (*e.g.* humans) which can be accessed by adjusting its autonomy. They do not, however, address the question of how to model uncertainty or how a request for more information would be triggered.

Another approach addressing the problem of when to request help from human operators is presented in [155]. Like in our system, requesting help is one of several possible communication modes and is motivated by both the scarceness of the human resource and the requirement to integrate multiple robots. The decision of when to request help is computed using a decision tree taking expected time delays and the likelihood of success into account. The latter requires a model of the user which is another parallel to the work presented here [154]. Unlike in the applications anticipated as part of this work, operators can actively participate in the task. The application they present is the large-scale assembly of structures in a simulated space environment. In contrast, this work regards robots as the sole decision-makers and humans are treated as a resource to robots.

A non-robotics AA system which uses probabilistic methods to decide when to switch control is presented by Scerri *et al.* [149][150]. A set of Markov Decision Processes (MDPs) are utilised to make autonomous decisions on behalf of users and query their input if necessary. The application is a distributed software scheduler able to schedule meetings and order meals. Scerri's work specifically addresses the problems of making decisions and switching autonomy when multiple agents have a common goal. Multiple robots having a common objective has not been considered as part of this work. Scerri's approach may be applied in future work. For robotics applications, however, more uncertainties need to be taken into account than MDPs can handle which suggests the use of Partially Observable Markov Decision Processes (POMDPs) [20].

A fundamental difference between all classical AA (including Scerri's work) and the approach presented in this chapter is the type of human input. Whereas classical AA systems transfer *decision-making* control to humans [149], the system presented here queries humans for *observations* and the decision making remains with the agent/robot. This exhibits true peer-to-peer interaction since humans are not able to override the robots' decisions.

5.4.4 Applications Using VOI

Using VOI theory to decide what information source to query is not new and has been applied to a wide range of applications. A few examples are listed below. To the best of the author's knowledge, this work is the first to apply VOI theory to human-robot communication.

An early work using VOI for planning a robot's order of sensing actions is presented in [102]. The cost in this case was the time spent on the sensory action. In the sensor network field, a trade-off between certainty of the estimated states and power consumption exists [101].

Each sensor only has a limited power budget in this work and the goal is to find the most informative subset of observations in the network. The research area of *distributed sensor management* uses information value theory for problems such as sensor-target assignments, sensor cueing or mode management [116][131]. Another problem in distributed systems is to decide when to communicate information. This problem can be approached similarly using the value of communication as a measure [39][12].

5.4.5 Human-Robot Interaction Metrics

A methodology for measuring the effectiveness of a human-robot team was presented in Section 5.3.1. Others have addressed the problem of finding metrics to evaluate human-robot systems more formally [32][165][133][143][5]. One of the goals of measuring the effectiveness of a human-robot team is to enable the prediction of an appropriate number of robots a user can effectively operate for a given task ("Fan-Out") [133][32]. Another goal is to compare different human-robot system configurations for a given task. An example is a planetary exploration task where different options may exist to combine human and robotic resources [143]. In contrast, effectiveness was measured here to determine an appropriate autonomy level for an adjustable autonomy system.

Crandall *et al.* propose a set of general *metric classes* applicable to all parts of a human-robot system: operators, individual robots, and the overall team [32]. The methodology proposed in Section 5.3.1 falls in the metric class of *Interaction Efficiency* measuring, among other things, how human input affects robot performance.

The simulation presented in Section 5.3.1 can be seen as a proof-of-concept method which needs to be complemented by experiments employing "real" human subjects in the future. For the anticipated experiments, additional metrics could be incorporated to measure human-robot team effectiveness. They include the *Mean Time Between Interventions* (MTBI) and the *Mean Time To Intervene* (MTTI) as defined by Arnold [5], and metrics related to the operators' SA [195][153][43].

5.5 Summary

This chapter demonstrated how human operators can contribute information to improve robotic decision making. A probabilistic representation in the form of an influence diagram lies at the heart of the approach. The representation is used to fuse observations from sensors with observations from human operators. Operators are treated as an expensive resource in this chapter which motivates the use of the robot-pull communication pattern: robots decide when it is necessary to query operators, and thus, when to relinquish full autonomy.

The questions of what queries to pose and when to pose them were addressed in Section 5.1. It was argued that answers to these questions can be derived from the probabilistic representation. Furthermore, it was shown how the representation can be used to overcome difficulties specific to requesting information from operators.

An influence diagram for a mobile robot navigation task was introduced in Section 5.2. The model was used to explain how cooperative human-robot decision making works in practice. An experimental evaluation using the model was presented in Section 5.3. Two experiments were conducted: (1) simulated human, and (2) a user study.

The simulated human experiment demonstrated that robot performance increases with more human input. Better performance comes at the cost of a larger number of operator queries. These two metrics were used to measure the overall effectiveness of the human-robot team as a function of the autonomy level. Both metrics were summed and weighted into a single scalar called team effectiveness. The weights were chosen manually according to the constraints and priorities of a mission. Using three example scenarios, it was shown how the team effectiveness measure can be used to find an appropriate autonomy level.

Results from the simulated human experiment were used as a guide to set the autonomy level for the user study. The user study was conducted to compare the developed system to a benchmark method of robot navigation – conventional teleoperation. Metrics were performance, operator workload, usability, and users' perception of the robot which were measured both objectively and by self-report using questionnaires. The results showed that the proposed robot-pull system has four advantages over teleoperation: (1) less performance variability among operators, (2) less dependency on available sensor information, (3) lower workload, (4) higher usability and potential for peer-to-peer interactions.

Chapter 6

Conclusions

This chapter starts by summarising the contents of this thesis in Section 6.1. Section 6.2 proceeds with discussing avenues for future research based on this work, and presents anticipated real-world applications. Finally, Section 6.3 concludes.

6.1 Summary

Chapter 1 motivated the objective of this thesis: to investigate methods of combining the perceptual abilities of humans and robots to cooperate effectively. It was argued that communication is necessary if humans and robots are to solve a task together. The approach to realise communication was outlined in general terms, and the focus of the thesis was introduced, namely human-robot information fusion. Four relationship taxonomies based on communication were introduced. Complemented by standard taxonomies from the literature, the full set of taxonomies was used to classify related human-robot systems. The same set was applied to classify the contents of each chapter in this thesis.

Chapter 2 contained the background of probabilistic representations which was required to present the rest of the document. Probabilistic robotics representations from the literature were reviewed using the categories of perception, decision making, and planning. Arguments were presented in favour of using graphical model representations such as Bayesian Networks (BNs) and Influence Diagrams (IDs) for human-robot communication. An overview of these representations and techniques for inference and learning was presented briefly. The chapter finished with a human-robot system graph showing an ID representation for human-robot communication.

Chapter 3 proposed probabilistic data types such as likelihoods and posteriors for bidirectional human-robot communication. From that point on, the main data flow was restricted to human-to-robot which led to the concept of regarding humans as information sources or high-level sensors. Arguments in favour for this concept were presented and submitting information was compared to providing actions. The differences between robotic and human sensors were highlighted. It was shown how these differences can be incorporated using the type of probabilistic representations introduced in Chapter 2. The approach to user modelling taken in this thesis, namely Human Sensor Models (HSMs), was introduced next. Limitations of the overall approach were also discussed. The experimental section of the chapter demonstrated both the benefits and limitations of human-robot information fusion in simulation, and presented a HSM for range observations built from real user data.

Chapter 4 presented an application of human-robot information fusion in the context of scalable, fully decentralised information gathering in an outdoor environment. First, algorithms for decentralised fusion were presented which impose constraints on the integration of human operators into a decentralised system and hence the roles they can play. A shared environment representation for geometric and visual feature properties was introduced. The use of the shared representation for human-robot information fusion was particularly emphasised. The experimental section presented results from the deployment of a human-robot team comprised of a UAV, a ground vehicle, and two human operators. Operators submitted observations using the human-push communication pattern in the experiment. The main results were presented as five information exchange patterns demonstrating human-robot information fusion in a qualitative manner.

Chapter 5 added platform decision making to the representation. The same information fusion mechanism as in previous chapters was used to combine human and robotics evidence into probabilistic beliefs. In addition, the robot utilised its beliefs to make decisions. Human operators were regarded as a sparse resource available to the robot in this chapter. This motivated the usage of the robot-pull communication pattern to query operators for information only when the expected value of their observation exceeded the cost of obtaining it. The system can be viewed as an adjustable autonomy system whereby the robot adjusts its autonomy at run-time based on the uncertainty of its beliefs. A navigation task was used to demonstrate the adjustable autonomy system. Two experiments were presented using the navigation task: (1) a quantitative evaluation of human-robot team effectiveness to determine an appropriate autonomy level, and (2) an extensive user study.

6.2 Future Work

Broadly, there are two areas which have a large potential for future research. The first one is concerned with probabilistic modelling, the second one with human-robot dialog.

6.2.1 Probabilistic Modelling Issues

The potential of probabilistic representations for human-robot communication has not been fully exploited yet in this thesis. Firstly, models were simplified which was required to keep the work focussed on human-robot information fusion. Secondly, the main usage of the models was online inference which assumed a fixed model structure and parameters. Both points offer a potential for future work.

Advanced Models The visual model used in Chapter 4 consisted of two levels only. It is desirable to demonstrate the usage of representations containing many levels of abstraction and different time scales. An example of a good application for a multi-level model is given in [109]. A person's daily movement through the community is modelled and inferred using GPS sensor measurements. An extension of that work could be to add observations from human operators at higher levels, *e.g.* current transportation mode, to infer other states in the model more accurately.

A different type of simplification was applied to the navigation model used in Chapter 5: the robot's decision model was static and only used the current state to make navigation decisions. For higher-level decision making such as path planning, it is necessary to look ahead, *i.e.* have the capability to plan into the future. Partially Observable Markov Decision Processes (POMDPs) [20] are a suitable representation which may be utilised in future work for human-robot information fusion.

Learning from Human Observations The work in this thesis relied on building models of the world by either offline learning (Chapter 4) or expert design (Chapter 5). Human and robotic observations are fused online given the fixed model. As pointed out in Chapter 3, this approach is limited: the model may have to adapt online and human input can help to achieve this.

Supervised machine learning techniques are often applied offline with operators supplying data labels. Using Bayesian techniques, it is also possible to adapt models at run-time using incremental learning. Every human observation can be used to adjust the model's parameters. Operators may also be queried by the system to label data: the field of Active Learning investigates what training data should be selected for labelling [27].

An application of this idea to a robotic navigation task is presented in [112]. A service robot uses a POMDP model to plan and navigate through a populated indoor space in this work. To achieve portability to new locations, the transition and observation model parameters are adapted using a human-guided exploration phase. During this active learning phase, operators are occasionally queried to confirm the robot's most likely belief state.

The results of that work are encouraging to achieve the objective of *lifelong learning* [93] where robots use all interactions with humans to constantly update their model. One research question in this context is how to minimise the number of required operator queries.

Learning from Human Decisions The previous section anticipated online model adaptation using human *perceptual* input. This is in line with the rest of this work's philosophical stance of regarding humans as information sources. However, humans are also excellent decision makers which is a capability more commonly exploited for human-robot collaboration, *e.g.* in many adjustable autonomy systems. If the robot uses a decision-theoretic model to make decisions, human input in form of actions can be used to improve the model.

Similar to active learning as presented in the previous section, operators can be queried for *actions* while the system incrementally adjusts the decision model. Reinforcement learning techniques have successfully been applied by some researchers, *e.g.* to update personalised user models [149].

Another option is to sense the operators' decisions passively, *i.e.* in a natural and non-intrusive manner. An example application area is intelligent buildings where occupants switch lights manually [124]. Using this information, the system can continuously update the preferred light switching strategies and become gradually smarter.

Generating Model Structure Using Human Input So far, online learning of model parameters has been discussed as a direction for future research. Besides parameter values, the structure of the model also needs to be specified. Typically, as done in this thesis, a knowledge engineer designs the structure by incorporation of domain knowledge. As intelligent systems gradually enter everyday life, it is desirable to be able to learn from non-experts such as ordinary computer users. An example of integrating information from multiple "noisy experts" to generate the structure of a BN is presented in [142].

The idea of exploiting knowledge from non-experts to design the model structure could be applied to distributed robotic systems. Research is required to investigate how to translate from a user-understandable abstraction layer to a graphical model structure. An example application is a smart building where lights are switched according to people's typical motion paths. Users could specify typical routes of building occupants by drawing on a GUI which displays the floor plan. This "meta-structure" could be combined with additional background information and get translated into a BN structure.

6.2.2 Human-Robot Dialog

The human-robot information exchange investigated in this thesis has been limited to four simple communication patterns as described in Section 1.3.2. The number of interactions per exchange is either one (push patterns) or two (pull patterns: query & response). The scope of this thesis was further restricted to human-to-robot data flow and the usage of perceptual information to demonstrate human-robot information fusion.

Several extensions to the established communication framework are possible: usage of other probabilistic data types, investigation of robot-to-human data flow, and ultimately the establishment of a human-like *dialog* between humans and robots. The different directions are discussed below.

Human as a Decision Maker This thesis regarded human operators as information sources where *information* was defined as observations or likelihoods. Chapter 3 argued that information can be superior to actions, mainly for scalability reasons. However, other probabilistic data types such as utility functions and policies may be utilised for robot control as visualised in Figure 6.1. One avenue of research is to investigate how scalable these methods are, *i.e.* how well a single user can operate multiple robots. Another research question is how to combine human and robotic decisions on different control layers ("shared control") as addressed in [137].

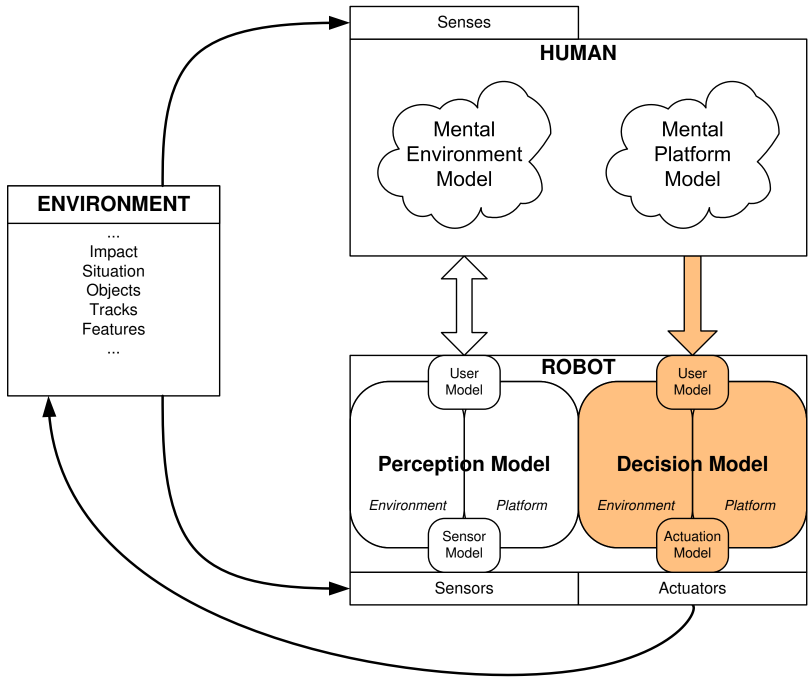


Figure 6.1: Human operator acting as a decision maker by submitting decision data such as actions, utility functions, and policies. A research question is how many robots can be operated simultaneously using these methods.

Robot-to-Human Data Flow The main data flow direction investigated in this thesis has been human-to-robot. The reverse direction as indicated in Figure 6.2 affords opportunities for future research. Operators often require data transmitted by robots to comprehend the current state of the world and the robotic platforms, *e.g.* in situations where operators are remotely located and are responsible to make decisions. The research area of Situation Awareness (SA) deals with the problem of generating accurate mental models [49]. In robotics, better mental models may be generated if humans have access to adequate abstractions of the data stored in the robot's representation. Translating probabilistic beliefs into human-understandable forms is an interesting approach which needs further investigation.

Another objective of human-to-robot data flow is the enhancement of humans' trust in the autonomous system. Trust in automation is a well-researched topic in human-machine interaction [48]. A novel contribution could be to let the robot explain its reasoning and decision making to humans, *e.g.* through a request (human-pull). The research hypothesis predicts a more accurate mental model which in turn, enhances humans' trust in robots' actions. One research question is how to translate between the mechanisms of probabilistic inference and decision making and a human-understandable output. Different modalities could be used to implement the output, *e.g.* graphical representations, speech, and text.

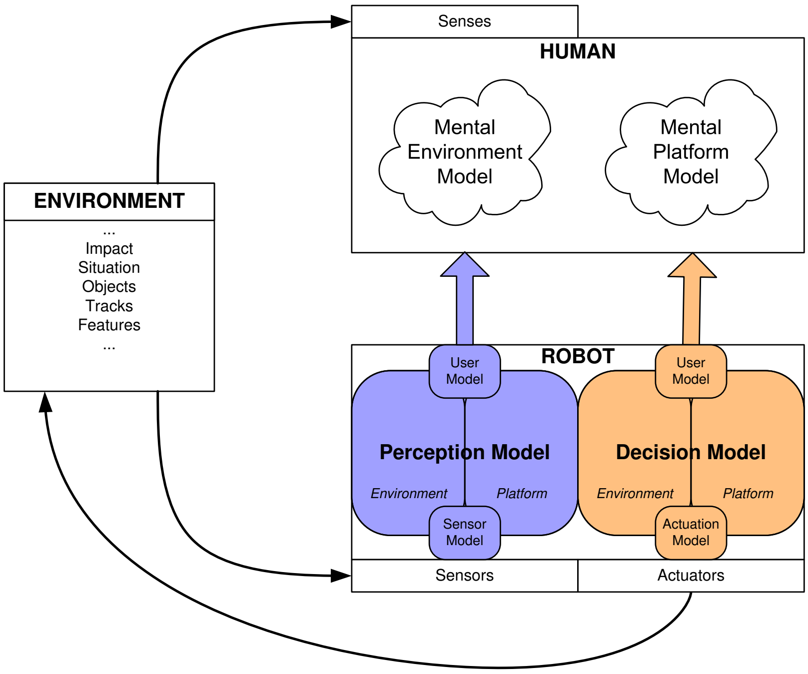


Figure 6.2: Robot-to-human data flow to increase SA and humans' trust in automation. Beliefs and decisions can be visualised and explained to generate more accurate mental models.

Advanced Dialog The ultimate objective for human-robot communication is to establish a more sophisticated human-like dialog between humans and robots. The following difficulties need to be overcome when managing human-like dialog [108]: asynchronicity (dialog events at overlapping time periods), mixed task initiative (robot or human can initiate dialog), open-endedness (no clear start and end points for dialog), resource-binding (actions must be produced in time to be effective dialog contributions), and simultaneousness (actions may be produced and received at the same time). Future work will investigate if probabilistic robotics representations can be used to develop sophisticated human-robot dialog with the properties listed above.

User Models Simple probabilistic user models named *Human Sensor Models* (HSMs) have been used throughout this thesis. As discussed in Section 3.6.3, more sophisticated models are required if (1) the operators' mental states become important, or (2) the individuality of operators needs to be modelled more accurately. The first condition becomes true for more advanced dialog systems as presented above. Robots need to know what the operators' goals and intentions are (context) to establish a meaningful dialog.

The second condition becomes true if better resource allocation in a mixed human-robot team needs to be achieved. For instance, the REQUESTSERVICE in Chapter 5 could also take current operator workload into account when querying operators for their observations. The cost for obtaining observations from operators can be adapted online based on the current belief of the users' workload. A similar idea is presented in [80] where computer users are alerted based on the belief of their current activity. The workload state could also be used to predict the operators' response time delays. Time delays are straight-forward to integrate into the cost as part of VOI theory as shown in [179] and [196].

Adapting the cost online by considering operator workload and expected time delays would result in a more intelligent adjustable autonomy system. The anticipated system would adjust its autonomy online based on both the environment's and the human operators' states.

Scalability Chapter 4 argued that truly scalable systems can only be achieved if the representation is concerned solely with the *environment*, not the robotic platforms themselves. The feasibility of scalable human-robot interactions was demonstrated using decentralised inference which was applied to an information gathering task.

Scalability has not been experimentally demonstrated for cooperative human-robot decision making. However, the system presented in Chapter 5 was designed with scalability in mind.

Two research directions are envisioned to realise scalability: firstly, apply decentralised inference to tasks such as path planning and exploration where the shared representation is concerned with the environment only. Secondly, for tasks where the representation contains platform states, investigate to what extend other probabilistic data types such as utility functions and policies are scalable. Experimental demonstrations involving multiple humans and robots will be conducted to measure scalability, *e.g.* through workload and task performance.

End-to-End AA System The adjustable autonomy system presented in Chapter 5 was concerned with a simple navigation task. It is desirable to build a more practical end-to-end system whose architecture includes higher-level control layers, *e.g.* for path planning. Operator queries should still be triggered by the robot to avoid the burden of having the operator switch between a set of discrete autonomy modes [24]. In future work, a probabilistic multi-layer decision model will be built (see [38]) with human-observable nodes at every level – a straight-forward extension of the material presented in Chapter 5.

6.2.3 Real-World Applications

This section provides a brief discussion about potential real-world applications of the work provided in this thesis and its extensions proposed above.

Reconnaissance & Crisis Management These two applications can be implemented based on the material presented in Chapter 4. Information gathering may be carried out semi-autonomously by UAVs, ground vehicles, sensor networks, and human operators who either work in the field or remotely. Often, operators are decision makers as part of these missions, so gaining SA is also an important issue.

Search And Rescue (SAR) & Fire Fighting SAR missions may be carried out by UAVs and rescue staff, *e.g.* to search for lost targets at sea [17]. Rescue staff may contribute information which is fused with sensor data to change the shape of the probability distribution of the target's location. The probability distribution is used to optimise the search strategy the UAVs employ.

To be able to predict bush fires, various mathematical models of fire spread have been developed [169]. Among them are probabilistic approaches which may be utilised or extended for human-robot information fusion. Fire fighters could enter higher-level information based on their experience which propagates to lower levels to predict routes the fire is likely to take.

Information entered by either SAR or fire fighting personnell could also be used to update the parameters and structure of the probabilistic models. This would be particularly useful in these domains since lost targets and fire spread are hard to model.

Mining Automating mining operations has recently gained attention in field robotics [1][33]. One of the objectives is to compile an accurate picture of the entire mine site including its geometry and geology, the location of all mining equipment, vehicles, and people. This task offers an opportunity for human-robot information fusion. One example is a model of the mine face geology which is hard to fully automate and laborious for humans. Geologist could cooperate with a semi-autonomous sensing system: sensors measure the geometry of different layers and provide an estimate of the geology while the geologist enters information for estimates which are too uncertain.

Another mining application is the automation of haul trucks where a single operator would be responsible for a fleet of autonomous trucks. The end-to-end AA system described above could be applied to keep operator workload low.

Planetary Exploration Robots have successfully been deployed for remote planetary exploration tasks on Mars [190]. These robots have been teleoperated from Earth ("Sojourner", 1997) or ran in a semi-autonomous mode ("Spirit" & "Opportunity", 2004) to minimise communication[1]. The robot-pull method as presented in Chapter 5 could be applied to this application where robots query operators only when necessary.

Intelligent Vehicles The ultimate goal of the intelligent vehicles domain is to develop a reliable and robust driverless car [192]. Besides the technical, legal and political problems which need to be overcome, a social issue is to get people to trust autonomous vehicles. It

[1]Dependent on the relative positions of Earth, Mars and the Sun, the communication delay between Earth and Mars ranges from 6.5 to 44 minutes [191].

may be possible to build up trust gradually if the vehicle is able to explain its (uncertain) reasoning and decision making when queried by the passenger (see Section 6.2.2).

Intelligent Buildings The vision of intelligent buildings with pervasive and embedded sensing, actuation, and computation has been around for at least two decades [29]. The interaction of everyday users with an autonomous system provides new challenges to human-system interaction research. For this application, a potential for incremental and lifelong learning as described in Section 6.2.1 may exist.

6.3 Conclusion

This thesis has made a contribution towards realising effective human-robot cooperation. It was shown how the perceptual abilities of humans and robots can be combined using probabilistic human-robot information fusion. Two application domains were successfully addressed with this method: scalable information gathering and cooperative decision making. A general human-robot communication framework was also proposed as part of this thesis. While the focus was on human-to-robot information flow, extensions are possible to reach the ultimate goal of establishing sophisticated human-like dialog between humans and robots.

Appendix A

Visual Model Evaluation

The objective for the visual model presented in Section 4.2.2 is to build a *compressed representation* of feature appearance properties that can be efficiently communicated and fused in the DDF network. Even though the model is also used for classification, the primary objective is to preserve the information content of the high-dimensional space. In this appendix, the superiority of Isomap for this task compared to PCA is empirically demonstrated. PCA is chosen as a comparison because it is a standard *unsupervised* method for dimensionality reduction. An unsupervised scheme is required since only some of our data is labelled for the purpose of building a human visual model as described in Section 4.2.2.

As a second step, model performance with respect to classification is investigated. For this purpose, a supervised method for dimensionality reduction is added to the discussion, namely the *Fisher Linear Discriminant Analysis* (FDA). Discrimination performance of all three methods (Isomap/PCA/FDA) is compared by (1) using a benchmark classification method, *k-nearest neighbour (k-NN)*, and (2) fitting Gaussians to demonstrate class separability.

A.1 Dimensionality Reduction

Dimensionality reduction using PCA PCA projects high-dimensional data onto a lower-dimensional subspace reducing the reconstruction error in a linear manner [46]. It first calculates the scatter matrix of the data set. Then, *eigenvectors* are computed and sorted according to decreasing *eigenvalues* which represent the variance of the data along the principal components. The eigenvectors with the largest eigenvalues are chosen to represent the high-dimensional data.

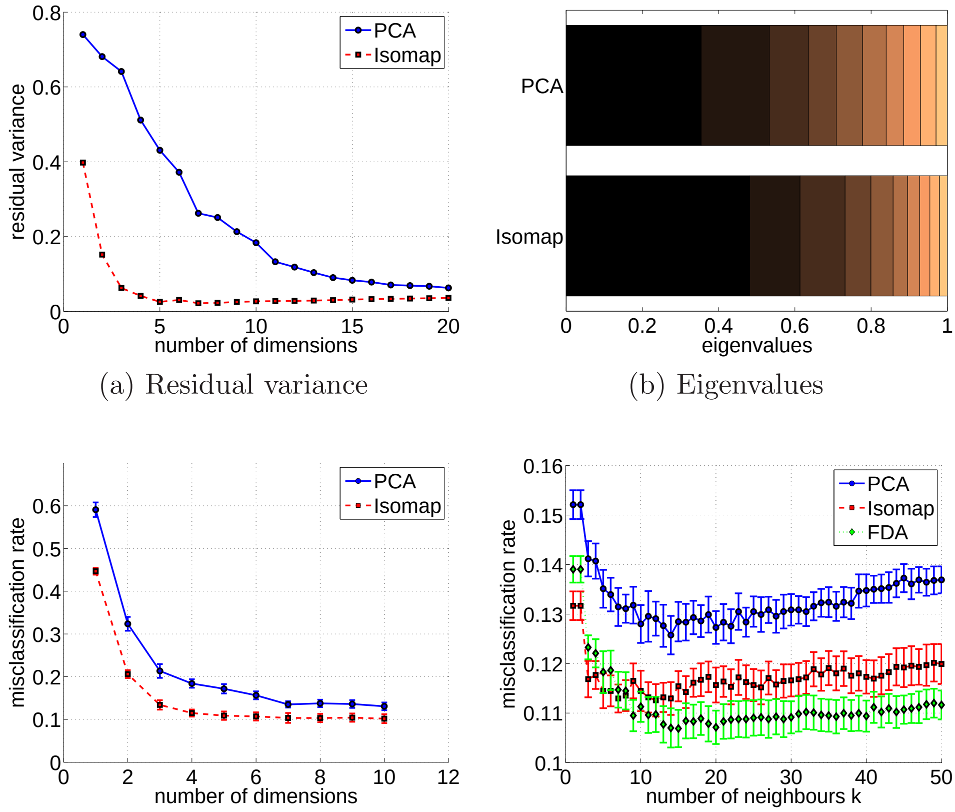
Figure A.1(a) shows a comparison of PCA's and Isomap's estimate of the manifold's dimensionality by computing the residual variances as described in [170]. The results are similar to theirs: even though the true dimensionality of the manifold is unknown here, PCA tends to overestimate the dimensionality and the curve does not cease to decrease significantly with added dimensions. Isomap's graph, in contrast, shows an "elbow" from which the intrinsic dimensionality can be estimated. A similar behaviour is reported for a hyperspectral data set in [185].

Figure A.1(b) shows a bar graph with the 10 largest eigenvalues as computed by Isomap and PCA. It can be concluded that Isomap is able to represent the high-dimensional manifold with less dimensions than PCA.

Dimensionality reduction using FDA While PCA finds components which are useful for representing data, *discriminant analysis* is designed to seek components that are efficient for discrimination. A natural generalisation of FDA for c classes was implemented which involves $c-1$ discriminant functions [46]. Unlike PCA or Isomap which give an estimate of the dimensionality of the reduced space, FDA fixes it to $c-1$ which is 16 for the data set under consideration.

A.2 Classification Performance

Classification using *k*-NN *k-NN* is a simple non-parametric method to estimate probability densities or perform classification [46]. A training sample in multi-dimensional space is classified by a majority vote of its closest k neighbours. Euclidian distance is used to compute distances between points. Figure A.1(c) shows the misclassification rate as a function of the dimensionality of the state space generated by Isomap and PCA (FDA is not shown because its dimensionality is fixed to 16). Three neighbours were used to classify a data set which was divided into training and test sets using 10-fold cross-validation. Error bars indicate one standard deviation. Isomap performs better for all dimensions which is also reported for a face images classification task in [106].



(a) Residual variance (b) Eigenvalues

(c) *k-NN*: dimensions (d) *k-NN*: neighbours

Figure A.1: Comparison of the Isomap model to other standard methods: (a) shows the residual variance as a function of dimensionality computed for PCA and Isomap, Isomap shows an "elbow" in the curve where PCA does not; (b) shows the first 10 eigenvalues for PCA and Isomap as a bar graph, Isomap needs less dimensions than PCA to represent the data; (c) shows the *k-NN* misclassification rate for Isomap/PCA as a function of dimensionality, standard error bars are included for 10-fold cross-validation, number of neighbours $k = 3$; (d) shows the *k-NN* misclassification rate for Isomap/PCA/FDA as a function of neighbourhood size, number of dimensions $d = 16$.

Figure A.1(d) shows the misclassification rate as a function of the neighbourhood size (number of k's) for the three methods. For a valid comparison, 16 dimensions have to be used which is given by FDA. Comparing unsupervised methods, Isomap performs better than PCA in separating classes with the same label. Lower test errors are expected from FDA since it is a supervised method and optimised for discrimination. For the data set under consideration, FDA only performs better than Isomap if more than 8 neighbours are used.

Note that using *k-NN* is not an option for our task which is finding a probabilistic visual representation rather than pure classification. *k-NN* is a non-probabilistic method which

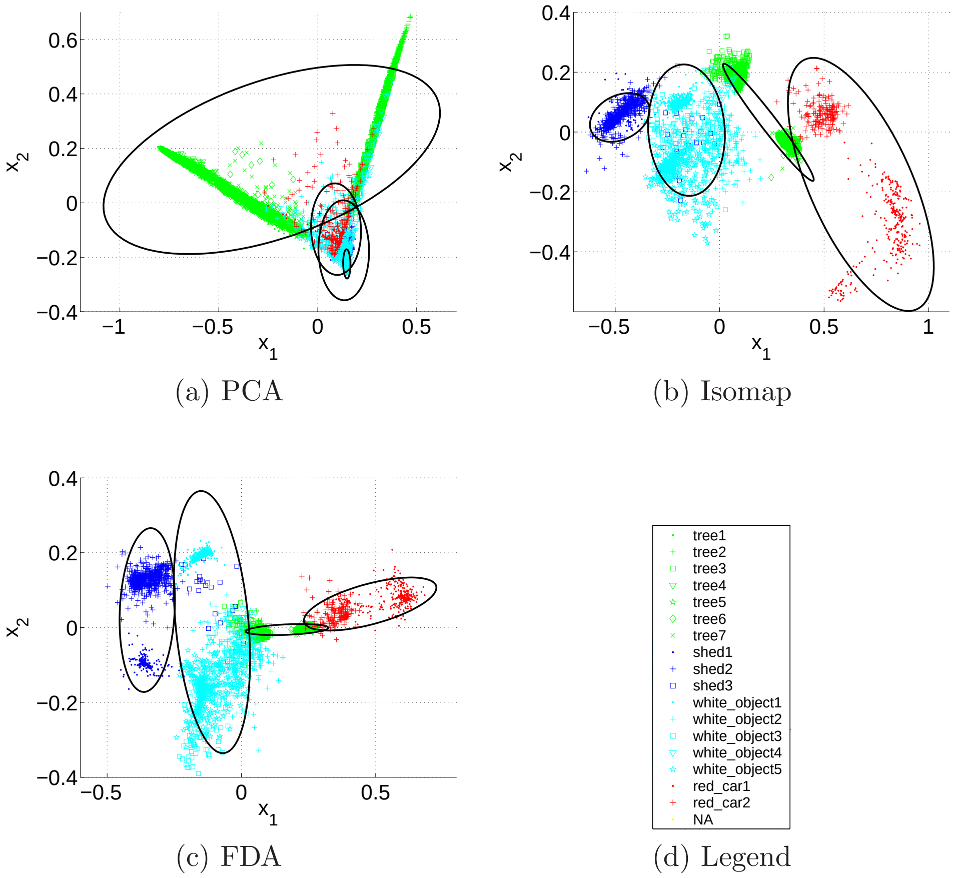


Figure A.2: State spaces with fitted Gaussians for (a) PCA, (b) Isomap, and (c) FDA. Only 2 dimensions are visualised with 4 Gaussian components. Isomap and FDA separate classes better than PCA.

cannot be integrated into a filtering scheme. It also requires all training data to be stored in memory for online classification while a closed-form probabilistic representation only stores a small number of parameters.

Class separability To visualise how the three methods under consideration separate classes, 2-dimensional Gaussians are fitted to data points in low-dimensional space as shown in Figure A.2. As expected, PCA performed worst in clustering data points with the same label whereas Isomap and FDA achieve comparable results.

Appendix B

User Study

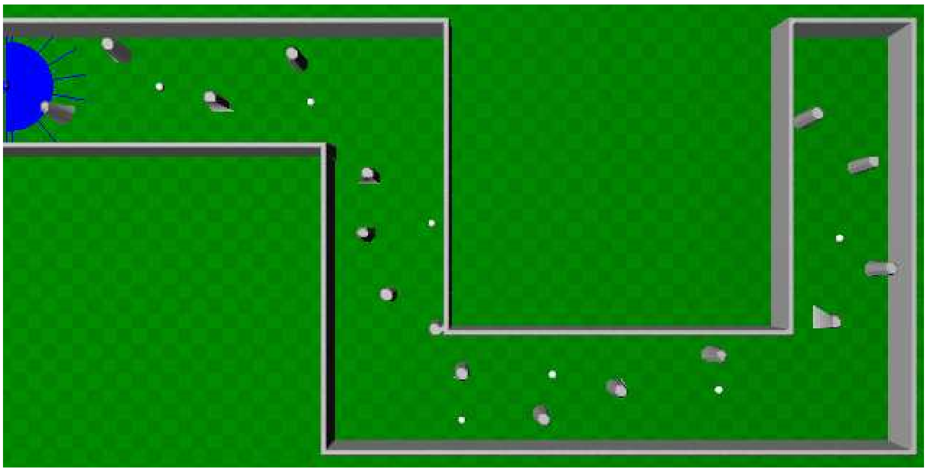
This appendix contains three things related to the user study presented in Chapter 5: (1) a VOI analysis for the navigation task is presented in Section B.1, (2) user study results of secondary importance are discussed in Section B.2, and (3) documents given to participants of the user study are shown in Section B.3.

B.1 VOI Analysis

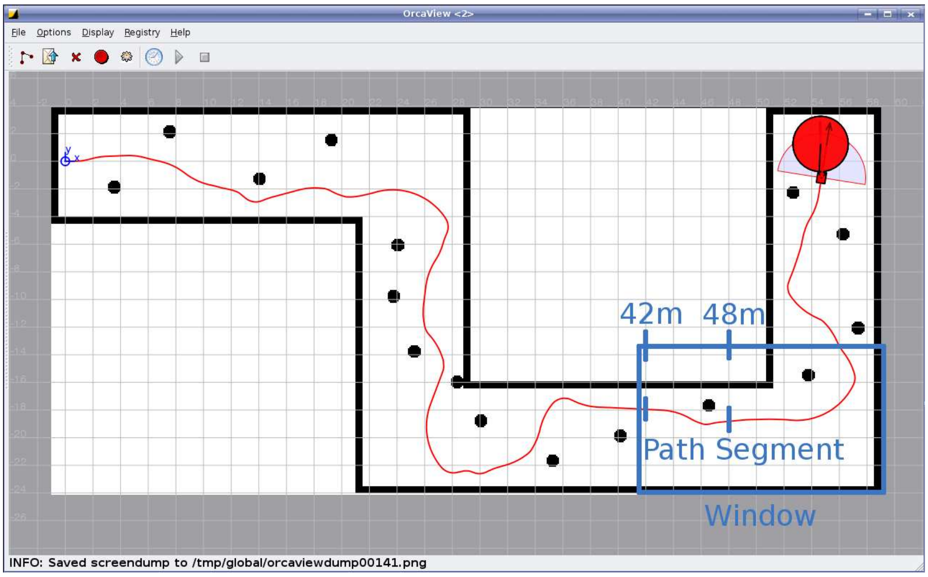
This section presents a VOI analysis of a typical robot through the simulated maze employed in the user study. Figure B.1 shows the maze from a global point of view. Figure B.1(a) shows the extruded world including columns and pushable spheres while Figure B.1(b) shows an example of the robot's history just before it reaches the last waypoint.

The part of the world enclosed in a window on the bottom right of the figure is used to show VOI results. It also serves to show the changes in the robot's state when encountering an obstacle as summarised in Table B.1 (see Section 5.2 for a complete description of the navigation representation).

Figure B.2 shows the VOI analysis for the section of the path highlighted within the window of Figure B.1(b). It is evident that information from experts is worth more than information from novices. The red asterisks correspond to the three states listed in Table B.1.



(a) World view

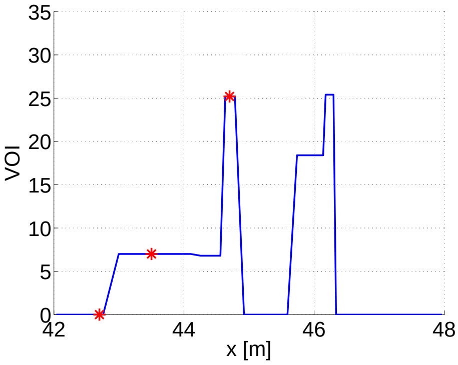


(b) Typical path

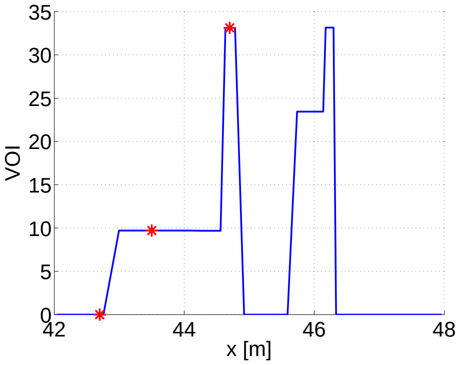
Figure B.1: Representative robot path through the maze: (a) the simulated extruded world including solid columns and pushable spheres, (b) the robot's path history before reaching the last waypoint. The part of the world encapsulated by a window on the bottom right corner serves to show detailed VOI results.

In state 1, the robot has not encountered any obstacles and there is no information to be gained from asking for *ObsTypNov,Exp* ($VOI \approx 0$). There is a low expected gain from querying for *SafSpdNov,Exp* and *SafDirNov,Exp*.

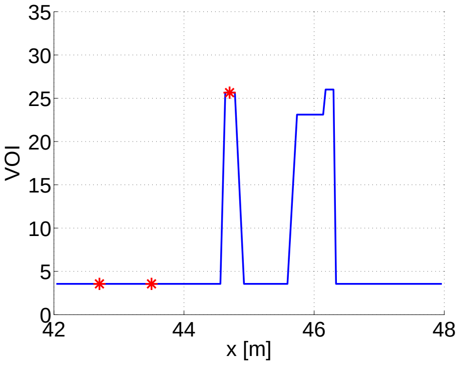
In state 2, the robot has found an obstacle and knowing *ObsTypNov,Exp* has now become valuable. However, there is no change for *SafSpdNov,Exp* because the obstacle is still far



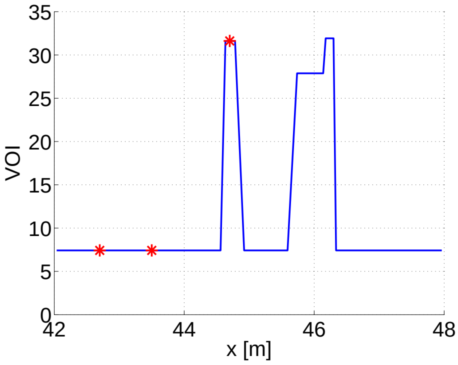
(a) *ObsTypNov*



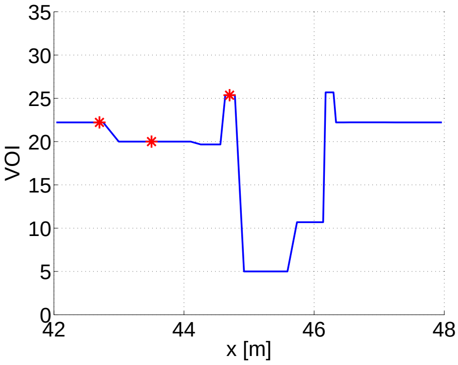
(b) *ObsTypExp*



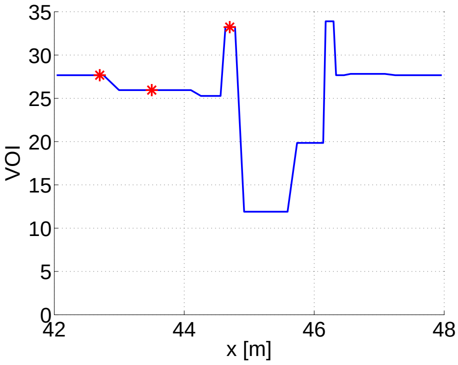
(c) *SafSpdNov*



(d) *SafSpdExp*



(e) *SafDirNov*



(f) *SafDirExp*

Figure B.2: VOI analysis for path section highlighted in Figure B.1(b). The x-axis is the x-coordinate of the robot's pose. The three states shown in Table B.1 correspond to the red asterisks.

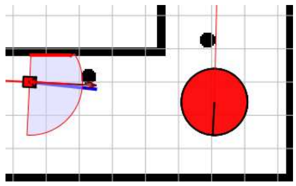
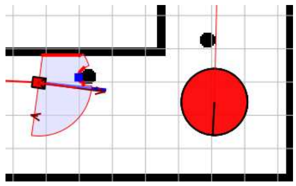
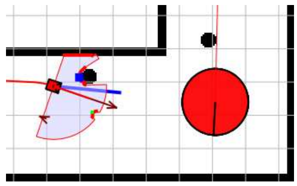
			
CommandedDirection	*straight*	*straight*	*left*
DirectionToObstacle	*none*	*left*	*left*
DistanceToObstacle	*far*	*far*	*close*

Table B.1: Three consecutive state changes of the robot when encountering an obstacle. The figures represent the part of the world enclosed in a window shown in Fig. B.1(b). Solid blue bars point towards the next waypoint while red arrows denote current speed and turnrate.

away. The small reduction in VOI for *SafDirNov,Exp* compared to state 1 can be explained as follows: in state 1, *DirectionToObstacle* was unspecified (*none*) while in state 2 it is *left* which does not conflict with *CommandedDirection* (*straight*).

In state 3, the robot has moved close to the obstacle. VOI for all three variables are highest for this "critical" state. *CommandedDirection* (*left*) now conflicts with the *DirectionToObstacle* (also *left*) which increases the VOI for *SafDirNov,Exp*.

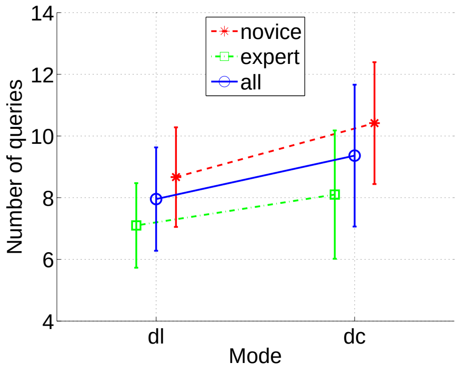
B.2 Dialog Measures

This section presents results from the user study which are not directly related to the hypotheses listed in Section 5.3.2. More specifically, they are concerned with the dialog system only (not teleoperation) and serve as feedback to improve the system design including the chosen representation and human sensor models.

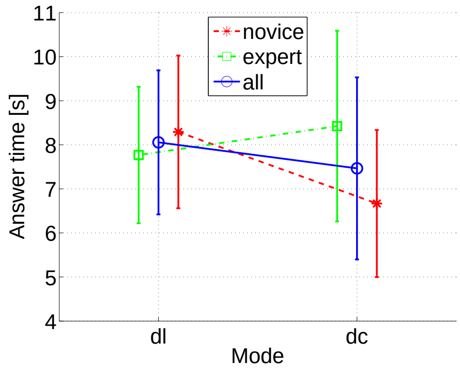
B.2.1 Suitability of Dialog

This section presents measures which were used to evaluate the dialog system with respect to (1) the number of operator queries, (2) answer time[1], (3) appropriateness of the questions, and (4) ease of answering different types of questions. The last two were evaluated using

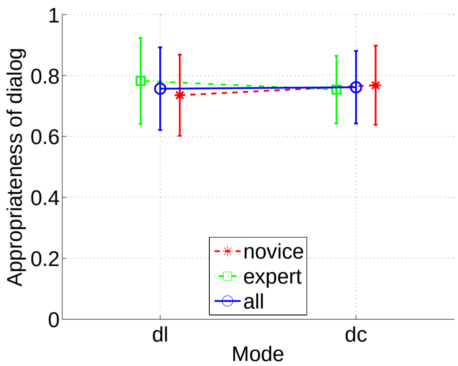
[1] Answer time is also referred to as *Interaction Time* (IT) in the literature [133][32].



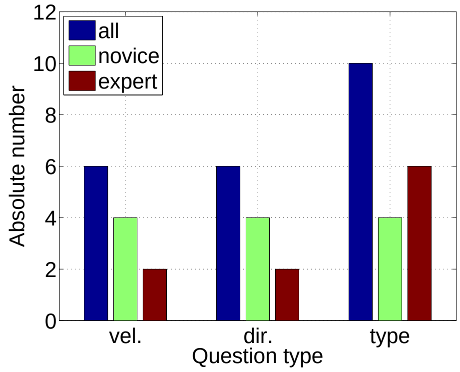
(a) Number of queries



(b) Answer time



(c) Appropriateness of dialog (scaled)



(d) Question type

Figure B.3: Evaluation of dialog modes (dl: dialog/laser, dc: dialog/camera, vel.: velocity, dir.: direction, type: obstacle type (pushable or not)).

four Likert statements in the the post-task questionnaire, and one multiple choice question in the post-experiment questionnaire. Figure B.3 shows the results for all four measures.

The number of queries posed by the robot was different for experts and novices due to the different sensor models used. Figure B.3(a) shows the number of queries for each mode. More queries were posed in dialog/camera than in dialog/laser. On average, the robot queried the operator 8.7 times per episode.

The average answer times per query were 8.1 and 7.5 seconds for the two modes. For experts, answer time was larger when a camera was available (Figure B.3(b). This can be explained by the information overload mentioned in Section 5.3.2: experts tended to interpret the scene more thoroughly whereas novices answered quickly. This was observed subjectively by the experimenter and indicated by four participants of the experts group in the free text section of the post-experiment questionnaire.

Appropriateness of questions was ranked high with no significant differences between the two modes and the novice/expert groups (Figure B.3(c)). No statistically significant differences were found between the ease of answering the 3 different questions as visualised in Figure B.3(d).

B.2.2 Human Sensor Model

The dialog system used different *a priori* human sensor models (HSMs) for expert and novice groups. HSMs were required to compute the VOI which the robot used to decide whether to query a human operator. In this section, the objectives are to investigate how accurate the participants' answers were, how much they differ between experts and novices, and ultimately, to refine the *a priori* HSMs.

In this study, the answers participants gave to the robot's queries were logged, together with the robot's belief of the *SafeDirection* and *SafeSpeed* states. Tables B.2 & B.3 show the distribution of the answers with the most likely state (the mode of the discrete distribution) as columns and the participants observations as rows. If the distribution's mode represented the truth, the tables could be used as the HSMs, or alternatively, fused with the *a priori* HSMs.

It is debatable whether the mode can be considered the truth for all entries. If it were the case, most answers would be expected to lie on the main diagonal of the table "matrix". When asked for the safe direction, participants often answered *straight* as can be seen from the high number in the *straight* rows of Table B.2. The most likely state, however, was either *right* or *left* which is simply more conservative. The participants answers, therefore, were not "wrong" in this case (they did not lead to collisions).

The same applies to Table B.3: participants often observed the safe speed to be *fast* whereas the most likely state was *slow*. In most cases, however, no collisions occurred, so the answers were not "wrong". Interestingly, experts were more confident to specify *fast* than novices (71.4% vs. 59.8%).

The distribution's mode can be considered the truth for some off-diagonal elements which are underlined in Table B.2. They represent more severe disagreements on the safe direction than discussed above and occasionally lead to collisions. Experts made less mistakes than novices here, but only slightly (15.7% vs. 15.0% and 6.9% vs. 3.6%).

In summary, it is difficult to judge the accuracy of participants' answers by using the most likely state as the "truth". A better "truth" could be obtained by letting the experimenter note the most appropriate answer subjectively whenever participants respond to queries. This method will be applied in future experiments and the data will then be used to refine the *a priori* HSMs.

A second result is that answers to dialog questions did not differ much between novices and experts in this experiment. This is not unexpected when such a coarsely grained state space is used. Larger differences are expected when, instead of using few choices (slow, fast, right, left), a more fine-grained or continuous value was requested.

Participants' observations	**Most likely state** *right*		*left*		*straight*
right	13	(25.5%)	2	(6.9%)	0
left	8	(15.7%)	9	(31.0%)	0
straight	30	(58.8%)	18	(62.1%)	0
Σ	51	(100.0%)	29	(100.0%)	0

(a) Novice

Participants' observations	**Most likely state** *right*		*left*		*straight*
right	9	(22.5%)	1	(3.6%)	0
left	6	(15.0%)	6	(21.4%)	0
straight	25	(62.5%)	21	(75.0%)	0
Σ	40	(100.0%)	28	(100.0%)	0

(b) Expert

Table B.2: HSMs for *SafeDirection*. Rows represent the participants' observations, columns the most likely state (mode of the discrete distribution). The *straight* columns are filled with zeros because *straight* never became the most likely state.

Participants' observations	**Most likely state** *stop*	*slow*		*fast*	
stop	0	0		0	
slow	0	39	(40.2%)	0	
fast	0	58	(59.8%)	4	(100.0%)
Σ	0	97	(100.0%)	4	(100.0%)

(a) Novice

Participants' observations	**Most likely state** *stop*	*slow*		*fast*	
stop	0	0		0	
slow	0	18	(28.6%)	1	(100.0%)
fast	0	45	(71.4%)	0	
Σ	0	63	(100.0%)	1	(100.0%)

(b) Expert

Table B.3: HSMs for *SafeSpeed*.Rows represent the participants' observations, columns the most likely state (mode of the discrete distribution). The *stop* columns are filled with zeros because *stop* never became the most likely state. The *stop* rows are filled with zeros because participants never chose to stop the robot.

B.3 User Study Documents

This section contains documents which were given to participants of the user study, namely the written instructions and the questionnaires.

B.3.1 Written Instructions

Instructions for Participants

The goal of this user study is to compare four different modes for driving a robot in a simulated environment. You will operate all four of them and will be asked to fill in a questionnaire after each task.

It is important to understand that we are not testing you but our system. Honest answers to the questions are critical for meaningful results. No names or other personal information will be recorded and all results are stored using a randomly assigned user name.

For each mode, the primary task is to get the robot to its goal as quickly as possible while making sure it is safe. A secondary task is to answer simple maths questions (adding 2-digit numbers). You should only do the secondary task if you have spare capacity to do so. Please do not stop the robot just to answer the math question. Only give an answer if you are pretty sure you got it right.

The experiment has several phases. Each one of them is described below:

1. **User name:** You will receive a random user name by drawing from a pool of user names.

2. **Pre-experiment questionnaire:** The purpose is to get an impression of how experienced you are in driving vehicles.

3. **Math practice:** This is to practice answering the math questions. Again, it's irrelevant how you personally score here.

4. **Robot driving:** You will drive the robot in four different modes. Each mode will be explained to you beforehand and you'll get as much time as you like to practice and become familiar before the actual experiment starts. The experiments can be paused if you feel like something is unclear. After each mode, a post-task questionnaire is to be filled in.

5. **Post-experiment questionnaire:** One more questionnaire is to be filled in at the end to give an idea of the overall user experience.

Please feel free to ask any questions you may have. Thank you very much for participating. Enjoy the ride!

World

Fig. 1 shows the kind of environment you will be driving in. It's a maze which contains two types of obstacles: (1) solid columns, and (2) spheres which the robot can push, i.e. you can drive into the spheres without penalty. The robot has a laser which scans the area in front of it (180 degrees, 3 meters) and a camera. You can see the laser scan and the camera image in the figure.

Figure 1: Example world, laser scan (blue), camera view.

Modes

This section will describe the 4 different modes. You can use this sheet to remind you of the different modes when filling in the questionnaires.

Teleop/Laser You will use a joystick to drive the robot manually in this mode. Fig. 2 shows what you will see on the screen. The red line is the robot's noisy laser scan (there's probably a wall in front of it). The blue bar indicates where the next waypoint is from the robot's point of view (you should turn left in the example). The bar will turn into a circle when a waypoint is reached. The dark red arrows show the current speed and turnrate: the larger the arrows the faster the robot drives and turns. In this case the robot drives fast while turning a little to the right.

Teleop/Camera The same as above but you also get the view of the camera which is mounted on top of the robot as shown in Fig. 3. You can see the sphere showing up in the laser scan.

Dialog/Laser The robot will drive itself in this mode. Every now and then, it will ask you a couple of questions as shown in Fig. 4. You can choose whether you want to answer one, two or none. You will base your answer on the laser scan you can see in the figure. If you don't answer within 30 seconds, the robot will decide what to do without your help.

Dialog/Camera The same as above but you also get the view of the camera on top of the robot as shown in Fig. 5. Note that the camera will pan a little towards the obstacle, so you can see it better. This means the camera image does not show the direction the robot points at. You can see that when you compare the camera image to the laser scan. The sphere is actually a little to the left, not directly in front of the robot.

Figure 2: Teleop/laser

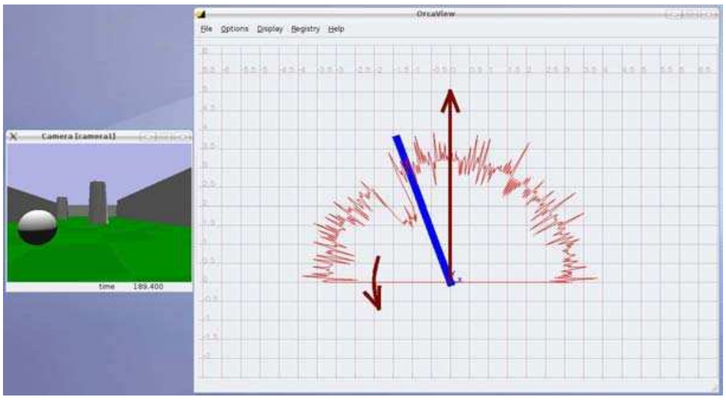


Figure 3: Teleop/camera

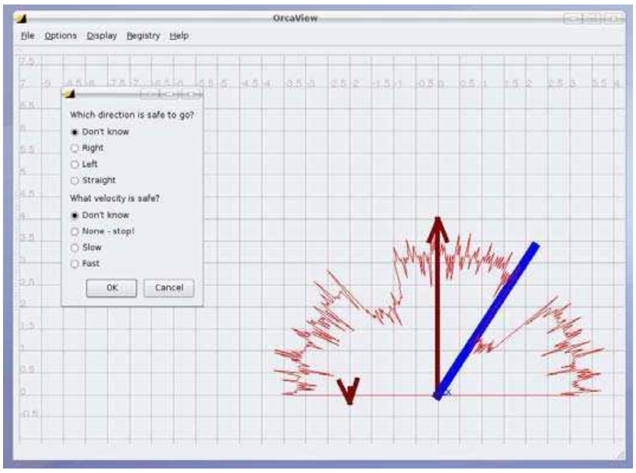


Figure 4: Dialog/laser

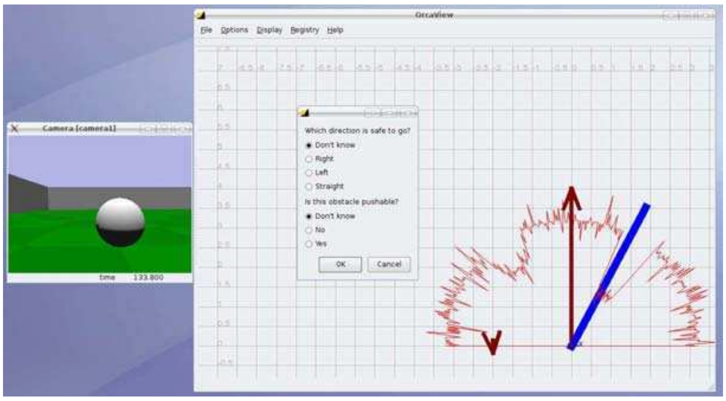


Figure 5: Dialog/camera

B.3.2 Pre-experiment Questionnaire

Pre-experiment questionnaire

General

User name: ____________ Age: _____ Gender: ☐ Male ☐ Female

Education

☐ High School ☐ Bachelor ☐ Master ☐ PhD

Discipline: ______________________________

Driving experience

Do you have a driving licence? ☐ Yes ☐ No
If yes, how often do you drive:
☐ daily ☐ weekly ☐ monthly ☐ rarely ☐ never

Computer experience

How often do you use a computer?
☐ daily ☐ weekly ☐ monthly ☐ rarely ☐ never
What type of user do you consider yourself?
☐ novice ☐ experienced ☐ developer

Video game experience

How often do you play video games?
☐ daily ☐ weekly ☐ monthly ☐ rarely ☐ never
What type of video games do you play?
☐ action ☐ strategy ☐ adventure ☐ other

Remote driving experience

What vehicles have you ever driven remotely?
☐ ground robot ☐ aerial robot ☐ toy car ☐ other ☐ none
If any, how often?
☐ Once ☐ A few times ☐ Many times

Other skills

On a scale from 1 to 10 (10 being best): how do you rate your hand-eye coordination?

On a scale from 1 to 10 (10 being best): how do you rate your navigation skills?

B.3.3 Post-Task Questionnaire

Post-task questionnaire

User name: ____________________

Mode:
☐ teleop/laser ☐ teleop/camera ☐ dialog/laser ☐ dialog/camera

Scale

This questionnaire consists of a number of statements. You are asked to express how much you agree with the statements using a 7 point scale. Please tick the appropriate boxes. They mean the following (from left to right):

1. very strongly agree
2. strongly agree
3. agree
4. undecided
5. disagree
6. strongly disagree
7. very strongly disagree

Important note: all statements refer to the primary task – getting the robot safely and fast to goal. Do no take answering the maths questions into consideration when you express your agreement/disagreement.

Getting the robot to its goal was mentally demanding in this mode.
agree ☐ ☐ ☐ ☐ ☐ ☐ ☐ disagree

I felt rushed doing the task in this mode.
agree ☐ ☐ ☐ ☐ ☐ ☐ ☐ disagree

It felt like hard work to accomplish the task.
agree ☐ ☐ ☐ ☐ ☐ ☐ ☐ disagree

I felt insecure, discouraged, irritated, stressed and annoyed.
agree ☐ ☐ ☐ ☐ ☐ ☐ ☐ disagree

This mode was easy to use and operate.
agree ☐ ☐ ☐ ☐ ☐ ☐ ☐ disagree

I would imagine most people would learn to operate this mode very quickly.
agree ☐ ☐ ☐ ☐ ☐ ☐ ☐ disagree

I found this mode very cumbersome to use.
agree ☐ ☐ ☐ ☐ ☐ ☐ ☐ disagree

I felt very confident using this mode.
agree ☐ ☐ ☐ ☐ ☐ ☐ ☐ disagree

I needed to practice quite a bit until I became familiar with this mode.
agree ☐ ☐ ☐ ☐ ☐ ☐ ☐ disagree

I think the robot was safe in this mode.
agree ☐ ☐ ☐ ☐ ☐ ☐ ☐ disagree

I trusted the robot in this mode.
agree ☐ ☐ ☐ ☐ ☐ ☐ ☐ disagree

The robot felt more like a partner rather than a tool in this mode.
agree ☐ ☐ ☐ ☐ ☐ ☐ ☐ disagree

The robot seemed to be smart in this mode.
agree ☐ ☐ ☐ ☐ ☐ ☐ ☐ disagree

Dialog

The following questions are only applicable to dialog modes.

I found it easy to answer the robot's questions.
agree ☐ ☐ ☐ ☐ ☐ ☐ ☐ disagree

I think the robot asked reasonable questions.
agree ☐ ☐ ☐ ☐ ☐ ☐ ☐ disagree

The robot asked me too many questions.
agree ☐ ☐ ☐ ☐ ☐ ☐ ☐ disagree

I got annoyed by answering these questions.
agree ☐ ☐ ☐ ☐ ☐ ☐ ☐ disagree

B.3.4 Post-Experiment Questionnaire

Post-experiment questionnaire

User name: ___________________

It was easiest to achieve the task in the following mode:
☐ teleop/laser ☐ teleop/camera ☐ dialog/laser ☐ dialog/camera

I felt most stressed in the following mode:
☐ teleop/laser ☐ teleop/camera ☐ dialog/laser ☐ dialog/camera

The robot was safest in the following mode:
☐ teleop/laser ☐ teleop/camera ☐ dialog/laser ☐ dialog/camera

It trusted the robot most in the following mode:
☐ teleop/laser ☐ teleop/camera ☐ dialog/laser ☐ dialog/camera

The robot appeared most “intelligent” in the following mode:
☐ teleop/laser ☐ teleop/camera ☐ dialog/laser ☐ dialog/camera

I found it easiest to answer questions about:
☐ velocity ☐ direction ☐ type of obstacle (pushable or not)

Describe what you liked (specify relevant modes):

Describe what annoyed you (specify relevant modes):

Anything else you’d like to say?

Bibliography

[1] ACFR, "Rio Tinto Centre for Mine Automation (CMA)," 2007. [Online]. Available: http://www.usyd.edu.au/CMA

[2] K. Ali, "Multiagent telerobotics: Matching systems to tasks," Ph.D. dissertation, Georgia Tech College of Computing, 1999.

[3] K. Ali and R. C. Arkin, "Multiagent teleautonomous behavioral control," *Machine Intelligence and Robotic Control*, 2000.

[4] R. Arkin, "Motor schema-based mobile robot navigation," *International Journal of Robotics Research*, vol. 8, no. 4, pp. 92–112, 1989.

[5] J. Arnold, "Towards a framework for architecting heterogeneous teams of humans and robots for space exploration," Master's thesis, Dept. of Aeronautics and Astronautics, Massachusetts Institute of Technology (MIT), 2006.

[6] N. Ashish, D. Kalashnikov, S. Mehrotra, N. Venkatasubramanian, R. Eguchi, R. Hegde, and P. Smyth, "Situational awareness technologies for disaster response," in *Terrorism Informatics: Knowledge Management and Data Mining for Homeland Security*, H. Chen, E. Reid, J. Sinai, A. Silke, and B. Ganoz, Eds. Springer, 2007.

[7] T. Bailey and H. Durrant-Whyte, "Simultaneous localisation and mapping (SLAM), part II: State of the art," *Robotics and Automation Magazine*, 2006.

[8] T. Bailey, B. Upcroft, and H. Durrant-Whyte, "Validation gating for non-linear non-Gaussian target tracking," in *9th International Conference on Information Fusion (Fusion'06)*, Florence, Italy, 2006, pp. 1–6.

[9] J. C. Baird and W. R. Biersdorf, "Quantitative functions for size and distance judgments," *Perception and Psychophysics*, vol. 2, pp. 161–166, 1967.

[10] Y. Bar-Shalom, *Multitarget multisensor tracking: advanced applications.* Norwood, MA: Artech House, 1990.

[11] Y. Bar-Shalom and T. Fortmann, *Tracking and Data Association.* Academic Press, 1988.

[12] R. Becker, V. Lesser, and S. Zilberstein, "Analyzing myopic approaches for multi-agent communication," in *Intelligent Agent Technology (IAT)*, Compaigne, France, 2005.

[13] M. Belkin and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," University of Chicago, Department of Computer Science and Statistics, Tech. Rep., 2002.

[14] J. Berger, *Statistical Decision Theory and Bayesian Analysis*, ser. Springer Series in Statistics. Springer, 1985.

[15] E. Blasch and S. Plano, "JDL level 5 fusion model: User refinement issues and applications in group tracking," *Aerosense*, vol. 4729, pp. 270–279, 2002.

[16] U. Boker and N. Dershowitz, "How to compare the power of computational models," in *Computability in Europe 2005: New Computational Paradigms*, L. T. S. Barry Cooper, Benedikt Loewe, Ed. Amsterdam: Springer-Verlag, Berlin, Germany, 2005, vol. 3526.

[17] F. Bourgault, H. Durrant-Whyte, and T. Furukawa, "Optimal search for a lost target in a Bayesian world," in *4th International Conference on Field and Service Robotics (FSR'03)*, 2003.

[18] J. Bradshaw, M. Sierhuis, A. Aquisti, P. Feltovich, R. Hoffman, R. Jeffers, D. Prescott, N. Suri, A. Uszok, and R. van Hoof, *Adjustable Autonomy and Human-Agent Teamwork in Practice: An Interim Report on Space Applications.* Kluwer, 2003, pp. 243–280.

[19] J. Brooke, "SUS: a "quick and dirty" usability scale." in *Usability Evaluation in Industry*, P. Jordan, B. Thomas, and A. McClelland, Eds. London: Taylor and Francis, 1996.

[20] A. Brooks, "Parametric POMDPs for planning in continuous state spaces," Ph.D. dissertation, The Australian Centre For Field Robotics (ACFR), The University of Sydney, 2007.

[21] A. Brooks, T. Kaupp, A. Makarenko, S. Williams, and A. Orebaeck, "Orca: A component model and repository," in *Software Engineering for Experimental Robotics*, ser. STAR, D. Brugali, Ed. Springer, 2007, vol. 30, pp. 231–251.

[22] R. Brooks, "A robust layered control system for a mobile robot," *IEEE Transactions on Robotics and Automation*, vol. 2, no. 1, pp. 14–23, 1986.

[23] D. Bruemmer, D. Few, R. Boring, J. Marble, C. Walton, and C. Nielsen, "Shared understanding for collaborative control," *IEEE Transactions On Systems, Man, and Cybernetics – Part A: Systems and Humans*, vol. 35, no. 4, pp. 494–504, 2005.

[24] D. Bruemmer, D. Few, C. Nielsen, and M. Walton, "Mixed-initiative control for collaborative countermine operations," *Submitted to IEEE Transactions on Robotics*, 2007.

[25] D. Bruemmer and M. Walton, "Collaborative tools for mixed teams of humans and robots," in *NRL Workshop on Multi-Robot Systems*, Washington, DC, USA, 2003, pp. 219–229.

[26] J. Burke, R. Murphy, E. Rogers, V. Lumelsky, and J. Scholtz, "Final report for the darpa/nsf interdisciplinary study on human-robot interaction," *IEEE Systems, Man and Cybernetics Part A*, vol. 34, no. 2, pp. 103–112, 2004.

[27] D. A. Cohn, Z. Gharamani, and M. Jordan, "Active learning with statistical models," *AI Research*, vol. 4, pp. 129–145, 1996.

[28] C. Conati, A. Gertner, and K. D. M. VanLehn, "On-line student modeling for coached problem solving using bayesian networks," in *Sixth International Conference on User Modelling (UM'97)*, 1997, pp. 231–242.

[29] D. Cook and S. Das, Eds., *Smart Environments: Technology, Protocols, and Applcations*. John Wiley & Sons, 2005.

[30] J. Crandall and M. Goodrich, "Characterizing efficiency of human robot interaction: a case study of shared-control teleoperation," in *IEEE/RSJ International Conference on Intelligent Robots and System (IROS'02)*, vol. 2, 2002, pp. 1290–1295.

[31] J. Crandall, M. Goodrich, D. Olsen, and C. Nielsen, "Validating human-robot interaction schemes in multi-tasking environments," *IEEE Transactions on Systems, Man, and Cybernetics – Part A: Systems and Humans*, vol. 35, no. 4, pp. 438–449, 2005.

[32] W. Crandall and M. Cummings, "Developing performance metrics for the supervisory control of multiple robots," in *2nd Annual Conference on Human-Robot Interaction (HRI'07)*, Washington, D.C., 2007.

[33] CSIRO, "Mining ICT & automation," 2007. [Online]. Available: http://www.dem.csiro.au/mining_automation

[34] M. L. Cummings and S. Bruni, "Collaborative human-computer decision making in network centric warfare," *International Journal of Aviation Psychology*, 2007.

[35] M. Cummings and S. Guerlain, "Using a chat interface as an embedded secondary tasking tool," in *2nd Annual Human Performance, Situation Awareness, and Automation conference*, 2004.

[36] J. Da Silva and S. Fukusima, "Stability of individual psychophysical functions for perceived distance in natural indoor and outdoor settings," *Perceptual and Motor Skills*, vol. 63, pp. 891–902, 1986.

[37] R. Davies, *Machine Vision: Theory, Algorithms, Practicalities*, 3rd ed. Elsevier, 2005.

[38] T. Dean and M. Wellman, *Planning and Control*. San Francisco, California: Morgan Kaufmann, 1991.

[39] R. Deaves, P. Greenway, and D. Bull, "Communications management in decentralised data fusion systems," in *IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI'96)*, 1996, pp. 796–805.

[40] J. Diard, P. Bessire, and E. Mazer, “Hierarchies of probabilistic models of navigation: the Bayesian map and the abstraction operator,” in *IEEE International Conference on Robotics and Automation (ICRA’04)*, New Orleans, LA, USA, 2004, pp. 3837–3842.

[41] A. Dielmann and S. Renals, “Multistream Dynamic Bayesian Network for meeting segmentation,” in *Multimodal Interaction and Related Machine Learning Algorithms Workshop (MLMI-04)*, S. B. Bourlard and H., Eds. Springer, 2004, pp. 76–86.

[42] S. Dittmer and F. Jensen, “Myopic value of information for influence diagrams,” in *13th Conference on Uncertainty in Artificial Intelligence (UAI’97)*, 1997.

[43] J. Drury, H. Keyes, and H. Yanco, “LASSOing HRI: analyzing situation awareness in map-centric and video-centric interfaces,” in *ACM/IEEE International Conference on Human-Robot Interaction (HRI’07)*, 2007, pp. 279–286.

[44] J. Drury, L. Riek, and N. Rackliffe, “A decomposition of UAV-related situation awareness,” in *ACM International Conference on Human-Robot Interaction (HRI’06)*, 2006, pp. 88–94.

[45] M. Druzdzel and L. van der Gaag, “Building probabilistic networks: ”where do the numbers come from?” guest editors’ introduction,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 12, no. 4, 2000.

[46] R. Duda, P. Hart, and D. Stork, *Pattern Classification*. New York: John Wiley & Sons, Inc., 2001.

[47] H. Durrant-Whyte and T. Bailey, “Simultaneous localisation and mapping (SLAM), part I: The essential algorithms,” *Robotics and Automation Magazine*, 2006.

[48] M. Dzindolet, S. Peterson, A. Pomranky, L. Pierce, and H. Beck, “The role of trust in automation reliance,” *International Journal of Human-Computer Studies*, vol. 58, pp. 697–718, 2003.

[49] M. Endsley, B. Bolte, and D. Jones, *Designing for Situation Awareness: An Approach to User-Centerd Design*. Taylor & Francis, 2003.

[50] T. Fawcett, “ROC graphs: Notes and practical considerations for data mining researchers,” HP Intelligent Enterprise Technologies Laboratory, Tech. Rep. HPL-2003-4, 2003.

[51] T. Fong, M. Bualat, L. Edwards, L. Flueckiger, C. Kunz, S. Lee, E. Park, V. To, H. Utz, N. Ackner, N. Armstrong-Crews, and J. Gannon, “Human-robot site survey and sampling for space exploration,” in *AIAA Space 2006*, September 2006.

[52] T. Fong, I. Nourbakhsh, R. Ambrose, R. Simmons, A. Schultz, and J. Scholtz, “The peer-to-peer human-robot interaction project,” in *AIAA Space 2005*, 2005.

[53] T. Fong, C. Thorpe, and C. Baur, “Collaboration, dialogue and human-robot interaction,” in *10th International Symposium of Robotics Research*, Lorne, Victoria, Australia, 2001.

[54] T. Fong, C. Thorpe, and C. Baur, "Multi-robot remote driving with collaborative control," *IEEE Transactions on Industrial Electronics*, vol. 50, no. 4, pp. 699–704, 2003.

[55] T. Fong, C. Thorpe, and C. Baur, "Robot, asker of questions," *Robotics and Autonomous Systems*, vol. 42, no. 3-4, pp. 235–243, 2003.

[56] T. Fong, C. Thorpe, and B. Glass, "Robot as partner: Vehicle teleoperation with collaborative control," in *NRL Workshop on Multi-Robot Systems*, Washington, DC, USA, 2002, pp. 195–202.

[57] T. Fong, C. Thorpe, and C. Baur, "Collaborative control: A robot-centric model for vehicle teleoperation," in *AAAI Spring Symposium: Agents with Adjustable Autonomy*, 1999.

[58] J. Forbes, T. Huang, K. Kanazawa, and S. Russell, "The BATmobile: Towards a bayesian automated taxi," in *International Joint Conference On Artificial Intelligence (IJCAI'95)*, 1995, pp. 1878–1885.

[59] K. Galotti, *Cognitive psychology in and out of the laboratory*, 3rd ed. Thomson/Wadsworth, 2004.

[60] D. Gentner and A. Stevens, Eds., *Mental Models*. Hillsdale NJ: Lawrence Erlbaum Associates, 1983.

[61] Z. Ghahramani and G. E. Hinton, "The EM algorithm for mixtures of factor analyzers," University of Toronto, Tech. Rep., 1996.

[62] J. Glover, S. Thrun, and J. Matthews, "Learning user models of mobility-related activities through instrumented walking aids," in *IEEE International Conference on Robotics and Automation (ICRA'04)*, vol. 4, 2004, pp. 3306–3312.

[63] K. Goldberg and B. Chen, "Collaborative control of robot motion: Robustness to error," in *IEEE/RSJ International Conference on Robots and Systems (IROS'01)*, vol. 2, 2001, pp. 655–660.

[64] K. Goldberg, B. Chen, R. Solomon, S. Bui, B. Farzin, J. Heitler, D. Poon, and G. Smith, "Collaborative teleoperation via the internet," in *IEEE International Conference on Robotics and Automation (ICRA'00)*, vol. 2, 2000, pp. 2019–2024.

[65] K. Goldberg, S. Dezhen, and A. Levandowski, "Collaborative teleoperation using networked spatial dynamic voting," *Proceedings of the IEEE*, vol. 91, no. 3, pp. 430–439, 2003.

[66] P. Gonsalves, G. Rinkus, K. Das, and N. Ton, "A hybrid artificial intelligence architecture for battlefield information fusion," in *Second International Conference on Information Fusion (Fusion'99)*, 1999.

[67] M. Goodrich, D. Olsen, J. Crandall, and T. Palmer, "Experiments in adjustable autonomy," in *IJCAI Workshop on Autonomy, Delegation and Control*, 2001.

[68] M. Goodrich and A. Schultz, "Human-robot interaction: A survey," *Foundations and Trends in Human-Computer Interaction*, vol. 1, no. 3, pp. 203–275, 2007.

[69] S. Grime and H. Durrant-Whyte, "Data fusion in decentralized sensor networks," *Control Eng. Practice*, vol. 2, no. 5, pp. 849–863, 1994.

[70] B. Grocholsky, J. Keller, V. Kumar, and G. Pappas, "Cooperative air and ground surveillance," *IEEE Robotics & Automation Magazine*, vol. 13, no. 3, pp. 16–25, 2006.

[71] B. Grocholsky, A. Makarenko, T. Kaupp, and H. Durrant-Whyte, "Scalable control of decentralised platforms," in *Information Processing in Sensor Networks*, F. Z. (eds.) and L. Guibas, Eds. Springer Verlag, 2003, pp. 96–112.

[72] B. Grocholsky, R. Swaminathan, J. Keller, V. Kumar, and G. Pappas, "Information driven coordinated air-ground proactive sensing," in *IEEE Conference on Robotics and Automation (ICRA'05)*, 2005, pp. 2211–2216.

[73] M. Guhe, W. Liao, Z. Zhu, Q. Ji, W. Gray, and M. Schoelles, "Non-intrusive measurement of workload in real-time," in *49th Annual Conference of the Human Factors and Ergonomics Society (HFES'05)*, 2005.

[74] J. Gunderson and W. Martin, "Effects of uncertainty on variable autonomy in maintainance robots," in *Workshop on Autonomy Control Software (Agents'99)*, 1999, pp. 26–34.

[75] S. Hart and L. Staveland, "Development of NASA-TLX (Task Load IndeX): results of empirical and theoretical research," in *Human Mental Workload*, P. Hancock and N. Meshkati, Eds. North-Holland Press, 1988, pp. 239–250.

[76] H. Hashemipour, S. Roy, and A. Laub, "Decentralized structures for parallel Kalman filtering," *IEEE Transactions on Automatic Control*, vol. 33, pp. 88–93, 1988.

[77] G. Hoffman and C. Breazeal, "Cost-based anticipatory action selection for human-robot fluency," *IEEE Transactions on Robotics*, vol. 23, no. 5, pp. 952–961, 2007.

[78] E. Horvitz, "Principles of mixed-initiative user interfaces," in *ACM SIGCHI Conference Human Factors in Computing Systems*, New York, U.S., 1999, pp. 159–166.

[79] E. Horvitz, J. Breese, D. Heckerman, D. Hovel, and K. Rommelse, "The Lumière project: Bayesian user modeling for inferring the goals and needs of software users," in *Fourteenth Conference on Uncertainty in Artificial Intelligence (UAI'98)*. Madison, WI: Morgan Kaufmann Publishers, 1998, pp. 256–265.

[80] E. Horvitz, A. Jacobs, and D. Hovel, "Attention-sensitive alerting," in *Conference on Uncertainty in Artificial Intelligence (UAI'99)*, Stockholm, Sweden, 1999, pp. 305–313.

[81] R. Howard and J. Matheson, "Influence diagrams," in *Readings on the Principles and Applications of Decision Analysis*, ser. Strategic Decisions Group, R. Howard and J. Matheson, Eds. Menlo Park, California: 721–762, 1984, vol. 2.

[82] D. Howell, *Statistical Methods for Psychology*, 4th ed. Thomson, 1997.

[83] C. Humphrey, C. Henk, G. Sewell, B. Williams, and J. Adams, "Assessing the scalability of a multiple robot interface," in *ACM/IEEE International Conference on Human-RobotInteraction (HRI'07)*, New York, USA, 2007, pp. 239–246.

[84] A. Jameson, "Adaptive interfaces and agents," in *Human-Computer Interaction Handbook*, J. Jacko and A. Sears, Eds. Mahwah, NJ: Erlbaum, 2003, pp. 305–330.

[85] F. Jensen, *Bayesian Networks and Decision Graphs*, ser. Information Science and Statistics. Springer, 2001.

[86] F. Jensen, B. Chamberlain, T. Nordahl, and F. Jensen, "Analysis in HUGIN of data conflict," in *Sixth Annual Conference on Uncertainty in ArtificialIntelligence (UAI'91)*. New York, NY, USA: Elsevier Science Inc., 1991, pp. 519–528.

[87] H. Jones and P. Hinds, "Extreme work groups: Using SWAT teams as a model for coordinating distributed robots," in *ACM 2002 Conference on Computer Supported Cooperative Work (CSCW'02)*, 2002.

[88] H. Jones and S. Rock, "Dialogue-based human-robot interaction for space construction teams," in *IEEE Aerospace Conference*, vol. 7, Big Sky, MT, 2002, pp. 3645–3653.

[89] M. Jordan, Ed., *Learning in graphical models*. MIT Press, Cambridge, 1999.

[90] S. Julier and J. Uhlmann, "General decentralized data fusion with covariance intersection (CI)," in *Handbook of Data Fusion*, D. Hall and J. Llinas, Eds. Boca Raton FL, USA: CRC Press, 2001.

[91] W. Kadous, R. Ka-Man Sheh, and C. Sammut, "Effective user interface design for rescue robotics," in *1st ACM SIGCHI/SIGART Conference on Human-robot interaction*, 2006, pp. 250–257.

[92] D. Kalashnikov, Y. Ma, S. Mehrotra, and R. Hariharan, "Modeling and querying uncertain spatial information for situational awareness applications," in *International Symposium on Advances in Geographic Information Systems*, 2006.

[93] A. Kapoor and E. Horvitz, "Principles of lifelong learning for predictive user modeling," in *Eleventh Conference on User Modeling (UM'05)*, 2005.

[94] T. Kaupp, "Scalable human-robot interactions in active sensor networks," Master Thesis, The University of Sydney, 2003.

[95] T. Kaupp and A. Makarenko, "Measuring human-robot team effectiveness to determine an appropriate autonomy level," in *IEEE International Conference on Robotics and Automation (ICRA'08)*, Pasadena, CA, USA, 2008, pp. 2146–2151.

[96] T. Kaupp, A. Makarenko, F. Ramos, and H. Durrant-Whyte, "Human sensor model for range observations," in *IJCAI Workshop Reasoning with Uncertainty in Robotics (RUR'05)*, Edinburgh, Scotland, 2005.

[97] S. Kiesler, “Fostering common ground in human-robot interaction,” in *IEEE International Workshop on Robot and Human Interactive Communication*, 2005, pp. 729–734.

[98] V. Klingspor, J. Demiris, and M. Kaiser, “Human-robot-communication and machine learning,” *Applied Artificial Intelligence*, vol. 11, no. 7/8, pp. 345–746, 1997.

[99] V. Klingspor, K. Morik, and A. Rieger, “Learning concepts from sensor data of a mobile robot,” *Machine Learning*, vol. 23, no. 2/3, pp. 305–332, 1996.

[100] K. Konolige, D. Fox, A. Ortiz, A. Agno, M. Eriksen, B. Limketkai, and J. Ko, “Centibots: very large-scale distributed robotic teams,” in *International Symposium on Experimental Robotics (ISER'04)*, Singapore, 2004.

[101] A. Krause and C. Guestrin, “Near-optimal nonmyopic value of information in graphical models,” in *21st Conference on Uncertainty in Artificial Intelligence (UAI'05)*, Edinburgh, Scotland, 2005.

[102] S. Kristensen, “Sensor planning with bayesian decision theory,” in *Reasoning with Uncertainty in Robotics*, 1995, pp. 353–367.

[103] G.-J. Kruijff, H. Zender, P. Jensfelt, and H. Christensen, “Clarification dialogues in human-augmented mapping,” in *1st ACM Human Robot Interaction Conference (HRI'06)*, Salt Lake City, Utah, USA, 2006, pp. 282–289.

[104] S. Kumar, F. Ramos, B. Upcroft, M. Ridley, L. L. Ong, S. Sukkarieh, and H. Durrant-Whyte, “A stochastic model for natural feature representation,” in *8th International Conference on Information Fusion (Fusion'05)*, vol. 2, Philadelphia, US, 2005.

[105] S. Lauritzen and D. Nilsson, “Representing and solving decision problems with limited information,” *Management Science*, vol. 47, pp. 1235–1251, 2001.

[106] M. Law and A. K. Jain, “Incremental nonlinear dimensionality reduction by manifold learning,” *IEEE Transactions of Pattern Analysis and Machine Intelligence*, vol. 28, no. 3, pp. 377–391, 2006.

[107] O. Lebeltel, P. Bessiere, J. Diard, and E. Mazer, “Bayesian robot programming,” *Autonomous Robots*, vol. 16, pp. 49–79, 2004.

[108] O. Lemon, A. Bracy, A. Gruenstein, and S. Peters, “An information state approach in a multi-modal dialogue system for human-robot conversation,” in *Perspectives on Dialogue in the new Millennium*, H. Reiser, P. Kunlein, and H. Zeevat, Eds. John Benjamins, 2003.

[109] L. Liao, D. Fox, and H. Kautz, “Learning and inferring transportation routines,” in *Nineteenth National Conference on Artificial Intelligence*, San Jose, CA, 2004, pp. 348–353.

[110] I. Llinas, “Information fusion for natural and man-made disasters,” in *Fifth International Conference on Information Fusion (Fusion'02)*, vol. 1, 2002, pp. 570–576.

[111] J. Llinas, C. Bowman, G. Rogova, A. Steinberg, E. Waltz, and F. White, "Revisiting the JDL data fusion model ii," in *The 7th International Conference on Information Fusion (FUSION'04)*, 2004.

[112] M. Lopez, R. Barea, L. Bergasa, and M. Escudero, "A human-robot cooperative learning system for easy installation of assistant robots in new working environments," *Journal of Intelligent and Robotics Systems*, vol. 40, pp. 233–265, 2004.

[113] A. Makarenko, "A decentralized architecture for active sensor networks," Ph.D. dissertation, The Australian Centre For Field Robotics (ACFR), The University of Sydney, 2004.

[114] A. Makarenko and H. Durrant-Whyte, "Decentralized data fusion and control in active sensor networks," in *7th International Conference on Information Fusion (Fusion'04)*, Stockholm, 2004.

[115] A. Makarenko, T. Kaupp, and H. Durrant-Whyte, "Scalable human-robot interactions in active sensor networks," *IEEE Pervasive Computing*, vol. 2, pp. 63–71, 2003.

[116] J. Manyika and H. Durrant-Whyte, *Data Fusion and Sensor Management: A Decentralized Information-Theoretic Approach.* Upper Saddle River, NJ, USA: Prentice Hall, 1995.

[117] J. Marble, D. Bruemmer, D. Few, and D. Dudenhoeffer, "Evaluation of supervisory vs. peer-peer interaction with human-robot teams," in *37th Hawaii International Conference on System Sciences*, 2004.

[118] M. Maris and G. Pavlin, "Distributed perception networks for crisis management," in *3rd International Conference on Information Systems for Crisis Response and Management (ISCRAM'06)*, B. V. d. W. Turoff and M., Eds., 2006.

[119] T. P. McNamara, "Mental representations of spatial relations," *Cognitive Psychology*, vol. 18, pp. 87–121, 1986.

[120] M. McNeese and D. Hall, "Computer aided cognition to support problem-centered decomposition of complex problems," in *Annual Meeting of the Human Factors and Ergonomics Society (HFES'03)*, 2003, pp. 523–527.

[121] H. Medicherla and A. Sekmen, "Human-robot interaction via voice-controllable intelligent user interface," *Robotica*, vol. 25, pp. 521–527, 2007.

[122] S. Mehrotra, C. Butts, D. Kalashnikov, N. Venkatasubramanian, R. Rao, G. Chockalingam, R. Eguchi, B. Adams, and C. Huyck, "Project Rescue: Challenges in responding to the unexpected," *SPIE Journal of Electronic Imaging, Displays, and Medical Imaging*, vol. 5304, pp. 179–182, 2004.

[123] E. Millan, J. Agosta, and J. Perez de la Cruz, "Bayesian student modeling and the problem of parameter specification," *British Journal of Educational Technology*, 2001.

[124] M. Mozer, "Lessons from an adaptive home," in *Smart Environments: Technologies, Protocols, and Applications*, D. Cook and S. Das, Eds. John Wiley & Sons, Inc., 2005.

[125] O. M. Mozos, R. Triebel, P. Jensfelt, A. Rottmann, and W. Burgard, "Supervised semantic labeling of places using information extracted from sensor data," *Robotics and Autonomous Systems*, vol. 55, no. 5, pp. 391–402, 2007.

[126] K. P. Murphy, "Dynamic bayesian networks: Representation, inference and learning," Ph.D. dissertation, University of California, 2002.

[127] K. P. Murphy, A. Torralba, and W. Freeman, "Using the forest to see the trees: A graphical model relating features, objects and scenes," in *Neural Information Processing Systems (NIPS'03)*, 2003.

[128] A. Nefian, L. Liang, X. Pi, and K. Murphy, "Dynamic bayesian networks for audio-visual speech recognition," *EURASIP, Journal of Applied Signal Processing*, no. 11, pp. 1–15, 2002.

[129] H. Nii, "Blackboard systems, part one: The blackboard model of problem solving and the evolution of blackboard architectures," *AI Magazine*, vol. 7, no. 2, pp. 38–53, 1986.

[130] D. Norman, "User centered system design: New perspectives on human-computer interaction," in *Cognitive Engineering*, D. Norman and S. Draper, Eds. New Jersey: Lawrence Erlbaum Associates, Inc., 1986, pp. 31–61.

[131] J. Nunnink and G. Pavlin, "A probabilistic approach to resource allocation in distributed fusion systems," in *Fourth International Joint Conference on Autonomous Agents and Multiagent Systems*. The Netherlands: ACM Press, 2005, pp. 846–852.

[132] N. Oliver, A. Garg, and E. Horvitz, "Layered representations for learning and inferring office activity from multiple sensory channels," *International Journal on Computer Vision and Image Understanding*, 2004.

[133] D. Olsen, S. Wood, and J. Turner, "Metrics for human driving of multiple robots," in *IEEE International Conference on Robotics and Automation (ICRA'04)*, vol. 3, 2004, pp. 2315–2320.

[134] L. Ong, M. Ridley, B. Upcroft, S. Kumar, T. Bailey, S. Sukkarieh, and H. Durrant-Whyte, "A comparison of probabilistic representations for decentralised data fusion," in *Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP'05)*, Melbourne, Australia, 2005, pp. 282–289.

[135] A. Papoulis, *Probability, Random Variables, and Stochastic Processes*, 3rd ed. McGraw-Hill, 1991.

[136] R. Parasuraman, T. Sheridan, and C. Wickens, "A model for types and levels of human interaction with automation," *IEEE Transactions on Systems, Man, and Cybernetics – Part A: Systems and Humans*, vol. 30, no. 3, pp. 286–297, May 2000.

[137] S. Parikh, V. Grassi, V. Kumar, and J. Okamoto, "Incorporating user inputs in motion planning for a smart wheelchair," in *IEEE International Conference on Robotics and Automation*, vol. 2, 2004, pp. 2043–2048.

[138] D. Patterson, L. Liao, D. Fox, and H. Kautz, "Inferring high-level behavior from low-level sensors," in *The Fifth International Conference on Ubiquitous Computing (UBICOMP'03)*, 2003.

[139] G. Pavlin, P. M. Maris, and F. Groen, "Causal Bayesian Networks for robust and efficient fusion of information obtained from sensors and humans," in *Instrumentation and Measurement Technology Conference*, Warsaw, Poland, 2007.

[140] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. San Mateo: Morgan Kaufmann, 1988.

[141] F. Ramos, S. Kumar, B. Upcroft, and H. Durrant-Whyte, "Representing natural objects in unstructured environments," in *NIPS Workshop: Machine Learning Based Robotics in Unstructured Environments*, Whistler, Canada, 2005.

[142] M. Richardson and P. Domingos, "Learning with knowledge from multiple experts," in *20th International Conference on Machine Learning (ICML'03)*, 2003.

[143] G. Rodriguez and C. Weisbin, "A new method to evaluate human-robot system performance," *Autonomous Robots*, vol. 14, pp. 165–178, 2003.

[144] M. Rosen-Zvi, T. Griffiths, M. Steyvers, and P. Smyth, "The author-topic model for authors and documents," in *Twentieth Conference on Uncertainty in Artificial Intelligence(UAI'04)*, 2004.

[145] J. Rosenblatt, "DAMN: a distributed architecture for mobile navigation," Ph.D. dissertation, CMU Robotics Institute, 1995.

[146] K. Ross, R. Chaney, G. Cybenko, D. Burroughs, and A. Willsky, "Mobile agents in adaptive hierarchical Bayesian Networks for global awareness," in *IEEE International Conference on Systems, Man, and Cybernetics*, vol. 3, 1998, pp. 2207–2212.

[147] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, pp. 2323–2326, 2000.

[148] S. Russel and P. Norvig, *Artificial Intelligence – A Modern Approach*, 2nd ed. Prentice Hall, 2003.

[149] P. Scerri, D. Pynadath, and M. Tambe, "Adjustable autonomy in real-world multi-agent environments," in *Fifth International Conference on Autonomous Agents (Agents'01)*, 2001, pp. 300–307.

[150] P. Scerri, D. Pynadath, and M. Tambe, "Towards adjustable autonomy for the real world," *Journal of Articial Intelligence Research*, vol. 17, pp. 171–228, 2002.

[151] B. Scholkopf, A. J. Smola, and K. R. Muller, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural Computation*, vol. 10, pp. 1299–1319, 1998.

[152] J. C. Scholtz, "Human-robot interaction: Creating synergistic cyber forces," in *NRL Workshop on Multi-Robot Systems*, Washington, DC, USA, 2002, pp. 177–184.

[153] J. Scholtz, B. Antonishek, and J. Young, "Implementation of a situation awareness assessment tool for evaluation of human-robot interfaces," *Transactions on Systems, Man, and Cybernetics, Part A*, vol. 35, no. 4, pp. 450–459, 2005.

[154] B. Sellner, R. Simmons, and S. Singh, "User modeling for principled sliding autonomy in human-robot teams," in *Third International Multi-Robot Systems Workshop, NRL*, Washington, DC, 2005.

[155] B. Sellner, F. Heger, L. Hiatt, R. Simmons, and S. Singh, "Coordinated multi-agent teams and sliding autonomy for large-scale assembly," *Proceedings of the IEEE - Special Issue on Multi-Robot Systems*, vol. 94, no. 7, pp. 1425–1444, July 2006.

[156] B. Sellner, L. Hiatt, R. Simmons, and S. Singh, "Attaining situational awareness for sliding autonomy," in *First Annual Conference on Human-Robot Interaction(HRI'06)*, March 2006.

[157] R. Shachter, "Evaluating influence diagrams," *Operations Research*, vol. 34, no. 6, pp. 871–882, 1986.

[158] R. Shachter and P. Ndilikilihesha, "Using potential influence diagrams for probabilistic inference methods," in *Ninth Conference on Uncertainty in Artifical Intelligence.* San Mateo, CA: Morgan Kaufmann, 1993, pp. 383–390.

[159] T. Sheridan, *Telerobotics, Automation and Human Supervisory Control.* MIT Press, 1992.

[160] T. Sheridan, "Eight ultimate challenges of human-robot communication," in *Sixth IEEE International Workshop on Robot and HumanCommunication*, 1997.

[161] M. Skubic, D. Perzanowski, A. Schultz, and W. Adams, "Using spatial language in a human-robot dialog," in *IEEE International Conference on Robotics and Automation (ICRA'02)*, vol. 4, 2002, pp. 4143–4148.

[162] J. Sola, A. Monin, M. Devy, and T. Lemaire, "Undelayed initialization in bearing only SLAM," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'05)*, Edmonton, Canada, 2005, pp. 2499–2504.

[163] D. Song and K. Goldberg, "Approximate algorithms for a collaboratively controlled robotic camera," *IEEE Transactions on Robotics*, vol. 23, no. 5, pp. 1061–1070, 2007.

[164] H. Sorenson and D. Alspach, "Recursive Bayesian estimation using Gaussian sums," *Automatica*, vol. 7, no. 4, pp. 465–479, 1971.

[165] A. Steinfeld, T. Fong, D. Kaber, M. Lewis, J. Scholtz, A. Schultz, and M. Goodrich, "Common metrics for human-robot interaction," in *2006 Human-Robot Interaction Conference.* ACM, March 2006.

[166] T. Stewart, “Notes for the development of a philosophy of computational modelling,” Carleton University Cognitive Science, Tech. Rep., 2005.

[167] A. Subramanya, A. Raj, J. Bilmes, and D. Fox, “Recognizing activities and spatial context using wearable sensors,” in *22nd Conference on Uncertainty in AI (UAI'06)*, 2006.

[168] E. Sudderth, A. Torralba, W. Freeman, and A. Willsky, “Learning hierarchical models of scenes, objects, and parts,” in *Intl. Conference on Computer Vision (ICCV'05)*, 2005.

[169] A. Sullivan, “A review of wildland fire spread modelling, 1990-present, mathematical analogues and simulation models,” *arXiv:0706.4130v1 [physics.geo-ph]*, 2007.

[170] J. Tenenbaum, V. DeSilva, and J. C. Langford, “A global geometric framework for nonlinear dimensionality redution,” *Science*, vol. 290, pp. 2319–2323, 2000.

[171] A. Tews, M. Mataric, and G. Sukhatme, “A scalable approach to human-robot interaction,” in *IEEE International Conference on Robotics and Automation (ICRA'03)*, vol. 2, Taipeh, Taiwan, 2003, pp. 1665–1670.

[172] G. Theocharous, K. Murphy, and L. Kaelbling, “Representing hierarchical POMDPs as DBNs for multi-scale robot localization,” in *IEEE International Conference on Robotics and Automation (ICRA'04)*, vol. 1, 2004, pp. 1045–1051.

[173] S. Thrun, “Probabilistic algorithms in robotics,” *AI Magazine*, vol. 21, no. 4, pp. 93–109, 2000.

[174] S. Thrun, “Towards programming tools for robots that integrate probabilistic computation and learning,” in *IEEE International Conference on Robotics and Automation (ICRA'00)*, vol. 1, 2000, pp. 306–312.

[175] S. Thrun, “Robotic mapping: a survey,” in *Exploring Artificial Intelligence in the New Millenium*, G. Lakemeyer and B. Nebel, Eds. Morgan Kaufmann, 2002.

[176] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*, R. C. Arkin, Ed. MIT Press, 2005.

[177] E. A. Topp, H. Huettenrauch, H. Christensen, and K. S. Eklundh, “Bringing together human and robotic environment representations - a pilot study,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'06)*, Beijing, China, 2006, pp. 4946–4952.

[178] A. Torralba, K. Murphy, W. Freeman, and M. Rubin, “Context-based vision system for place and object recognition,” in *Intl. Conference on Computer Vision (ICCV'03)*, 2003.

[179] C. Tseng and P. Gmytrasiewic, “Real-time decision support and information gathering system for financial domain,” *Physica A: Statistical Mechanics and its Applications*, vol. 363, no. 2, pp. 417–436, 2006.

[180] H. Tu, J. Allanach, S. Singh, K. Pattipati, and P. Willett, "Information integration via hierarchical and hybrid Bayesian Networks," *IEEE Transactions on Systems, Man, and Cybernetics – Part A: Systems and Humans*, vol. 36, no. 1, pp. 19–33, 2006.

[181] B. Upcroft, B. Douillard, T. Kaupp, M. Ridley, L. Ong, S. Kumar, T. Bailey, F. Ramos, S. Sukkarieh, and H. Durrant-Whyte, "Non-Gaussian state estimation in an outdoor decentralised sensor network," in *45th IEEE Conference on Decision and Control*, San Diego, CA, USA, 2006, pp. 366–372.

[182] B. Upcroft, M. Ridley, L. Ong, B. Douillard, T. Kaupp, S. Kumar, T. Bailey, F. Ramos, A. Makarenko, A. Brooks, S. Sukkarieh, and H. Durrant-Whyte, "Multilevel state estimation in an outdoor decentralised sensor network," in *10th International Symposium on Experimental Robotics 2006 (ISER'06)*, Rio de Janeiro, Brazil, 2006.

[183] S. Vasudevan, S. Gachter, V. Nguyen, and R. Siegwart, "Cognitive maps for mobile robots - an object based approach," *Robotics and Autonomous Systems*, vol. 55, no. 5, pp. 359–371, 2007.

[184] F. Wang, Y.-F. Ma, Z. H.-J., and J.-T. Li, "A generic framework for semantic sports video analysis using dynamic bayesian networks," in *11th International Multimedia Modelling Conference (MMM'05)*, 2005, pp. 115–122.

[185] X. Wang, S. Kumar, F. Ramos, T. Kaupp, B. Upcroft, and H. Durrant-Whyte, "Learning in hyperspectral images using a nonlinear dimensionality reduction technique," in *9th International Conference on Information Fusion (Fusion 2006)*, Florence, Italy, 2006.

[186] Y.-J. Wen, J. Granderson, and A. Agogino, "Towards embedded wireless-networked intelligent daylighting systems for commercial buildings," in *IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing*, 2006, pp. 326–331.

[187] F. White, "A model for data fusion," in *1st National Symposium on Sensor Fusion*, vol. 2, Orlando, FL, USA, 1988.

[188] Wikipedia, "Amazon Mechanical Turk," 2007. [Online]. Available: http://en.wikipedia.org/wiki/Amazon_Mechanical_Turk

[189] Wikipedia, "Disappearance of Steve Fossett," 2007. [Online]. Available: http://en.wikipedia.org/wiki/Disappearance_of_Steve_Fossett

[190] Wikipedia, "Exploration of Mars," 2007. [Online]. Available: http://en.wikipedia.org/wiki/Exploration_of_Mars

[191] Wikipedia, "Colonization of Mars," 2008. [Online]. Available: http://en.wikipedia.org/wiki/Colonization_of_Mars

[192] Wikipedia, "Driverless car," 2008. [Online]. Available: http://en.wikipedia.org/wiki/Driverless_car

[193] Wikipedia, "Occam's razor," 2008. [Online]. Available: http://en.wikipedia.org/wiki/Occam's_Razor

[194] D. Winterfeldt and W. Edwards, *Decision Analysis and Behavioral Research.* Cambridge University Press, UK, 1986.

[195] H. Yanco and J. Drury, ""Where am I?" Acquiring situation awareness using a remote robot platform," in *IEEE International Conference on Systems, Man and Cybernetics*, 2004, pp. 2835–2840.

[196] S. Zilberstein and V. Lesser, "Intelligent information gathering using decision models," Computer Science Department, University of Massachusetts, Tech. Rep. 96-35, 1996.