MACSimJX: A Tool for Enabling Agent Modelling with Simulink Using JADE

Charles R. Robinson, Peter Mendham, and Tim Clarke

***Abstract*—MACSimJX provides the means for advanced modelling and development of multiagent driven control systems. This is achieved by drawing together two modelling tools used extensively in their respective communities. These are Simulink, a tool used for control systems development and JADE, an environment for developing agents. Thus the strengths of their particular domains of application may be drawn upon to facilitate research and development in the joint field of decentralised systems control. To the authors knowledge no other implementation such as this exists. MACSimJX, otherwise known as the extension of MACSim with JADE, is available for download at www.agentcontrol.co.uk.**

***Index Terms*—MACSimJX for Decentralised Control, Simulink with JADE, Agent-Based Systems, Sensor Fusion, Control Architectures and Programming.**

I. INTRODUCTION

IN general, a decentralised system has its processing distributed such that each element in the system is capable of functioning in isolation. However, there is the potential for enhanced system performance because these processing units can communicate and cooperate with each other. Decentralised systems offer a number of advantages over their centralised counterparts that includes greater robustness, timeliness and fault tolerance. This paper reports an integrated software framework that connects control system simulation with multiagent theory to support the modelling of real-time decentralised systems.

Multi-agent architecture, a concept that began to be properly developed in the latter half of the 1980s [1], provides a natural software support structure for decentralised systems. In this context the word *agent* is used to refer to a software entity capable of operating by itself, with the ability to obtain information from, and effect changes on, its environment. These operations may include communication with other agents and are carried out in order to achieve some predefined objectives. There are many potential advances in system design that might be achieved through the development and application of the emerging multi-agent technology.

Simulink is a widely used tool in industry and academia. It is a graphical front-end for MATLAB (Matrix Laboratory) which allows representation of time varying systems through matrix manipulation. Simulink provides a graphical representation of these systems modelled through matrices. A vast array of libraries are available that provide the ability to connect a series of subsystem elements to construct and represent the internal mechanics of such dynamic and embedded systems.

Charles R. Robinson is with THALES R&T France.

Peter Mendham is with SciSys UK Ltd.

Tim Clarke is with The University of York.

MACSimJX provides access from Simulink to such a multiagent architecture, this facilitates the development of software control structures with features such as:

- Robustness, so that if part of the system fails the rest of the system will continue to operate with minimal loss of functionality.
- Scalability of the processing architecture.
- Fewer constraints on a system caused by computational bottlenecks or communication bandwidth.
- Modularity in terms of both the design and implementation.
- Synergy of sensors such that the overall machine perception is improved beyond basic fusion of data.
- An enhanced awareness of the state of the world.

Thus MACSimJX is an interface that enables models of systems created in Simulink to exchange data with a multiagent system created using JADE. A brief description of the agent paradigm and JADE follows, after which the manner in which MACSimJX integrates this with Simulink is discussed.

II. INTELLIGENCE AND MULTIPLE AGENTS

The word *agent* refers to a concept rather than to a particular entity or event. Concepts are general ideas that describe a class or category of things or events that have unique features but share common characteristics. Like other concept words such as tree or dinosaur the meaning of the word agent encompasses a set of ideas that people believe represent its general properties. This means that although it is possible to describe agents and trees, many traits or characteristics will depend on circumstances, the environment and our own experience. With this important qualification, a general definition for an agent is:

***Definition*:** An agent is an autonomous entity in an embedded environment that either solves problems by itself, or cooperates with other agents to find a solution. It has control over its internal state as well as its outputs and can run without external intervention.

An embedded environment implies a system that receives its information about the environment through sensors and acts on the environment through effectors. A multi-agent system is a collection of these interacting agents and can exhibit all the features required in a decentralised setup. It has the capacity to achieve tasks through the combined efforts of the individual agents that could not be done alone. This is particularly the case where the agents have different functional capacities. Agent systems have also been shown to exhibit emergent problem-solving behaviours through cooperation not explicitly part of the original design, one such example being swarm intelligence.

Jennings *et al.* [2] list many of the applications to which agents have been applied in industry, commerce, medicine and the entertainment sector. However, these authors stated that the agent community suffered from the lack of a systematic design methodology or an industrial strength multi-agent toolkit. Since then, these shortcomings have been addressed, or are being addressed, by FIPA and the development of several agent frameworks. A software agent framework is one that incorporates the various agent theories and provides a support structure of and for functions that facilitate the rapid development of a system based on these concepts.

JADE provides a framework that allows quick implementation of many of the inherent features one would expect for developing a multi-agent system. Much of the complexity is kept hidden from the user to allow ease of use. In addition, JADE has been steadily gaining support and has been more widely used over the last few years. For these reasons, and with support from a review in [3], JADE was chosen to be the framework to assist agent modelling for Simulink through MACSim.

A. The Java Agent Development Environment (JADE)

JADE was originally developed under TILab, formerly CSELT, in Italy [4] to address the lack of support available for building agent systems. As its name suggests, this framework offers an environment in which to create agents written in Java. It provides the runtime environment, which the agents require in order to operate. It also provides an extensive library of classes with methods built around the FIPA specification of agent characteristics and graphical interfaces for monitoring active agents.

Each instance of a runtime environment is called a *container* and several of these make up a *platform*. The first container to be created needs to be designated as the *main container*; subsequent containers then register with this as they join the platform. Containers can be spread across several networked computers. The main container hosts an agent management service (AMS) and a Directory Facilitator (DF). The AMS ensures that each agent has a unique name and can be used for loading and removing agents from the platform. The DF provides a means for agents to publicise their specialised services or for looking up the services provided by other agents. It is often referred to as the *Yellow Pages* [5].

Agents operate from within the containers. The structure of an agent consists of a *setup()* method, one or more behaviour methods, and a *takeDown()* method. The setup() method is executed the first time an agent is created and runs only once. It sets all the initial conditions needed to get the agent up and running and includes the behaviours required for the agent.

The behaviour methods can run concurrently and are responsible for carrying out the main tasks of an agent. This includes communicating with other agents. An agent can be put to *sleep* if it has no behaviours operating and can be awakened again after a specified period, or on receipt of a message requesting the execution of an action. This can be very useful because the agent consumes no processing power when it is in sleep mode.

To date, JADE has been applied, at least in theory, to a wide variety of areas including urban and aircraft traffic control [6], [7], providing travel industry support [8], manufacturing [9] and robotics [10], [11].

III. ARCHITECTURE

Whilst Simulink is really effective for carrying out simulations, it falls short of offering the tools necessary to set up an agent framework. One very useful aspect of Simulink, however, is that it provides a work-around for adding functionality in the form of S-functions. These allow programs to be written in other languages, particularly C, that can be encapsulated in the Simulink environment and then used where desired, running in their native language.

Despite this prospect of a solution, where the agents could be created through C++ or Java code in one of these functions and run in Simulink, there is a further complication. S-functions are unable to handle multiple threads of execution: they become unstable if several processes run concurrently inside Simulink [12], [13]. Unfortunately, this functional property is essential for a multi-agent system. To overcome this problem, a program called MACSim was created which still utilises the S-function ability of Simulink, but only as a gateway to pass data to a program outside MATLAB with parallel processing capacity.

A. Structure of MACSim

MACSim, or the *Multi-Agent Control for Simulink* program, described in [12], was purposely developed as a medium through which a program for implementing agent designs developed in C/C++ or Java might pass data to and from Simulink. Although MACSim is written primarily in C++, it includes a wrapper to enable interaction with Java programs. MACSim has a client-server architecture, where the client part is embedded in Simulink through an S-function, and the server code is then incorporated in the separate program as indicated in Figure 1. Communication between the client and server is then performed through the use of *named pipes* in Windows. Use of MACSim circumvents the multi-threading issue because a separate program can now be used with protocols in place to ensure synchronicity if so desired.

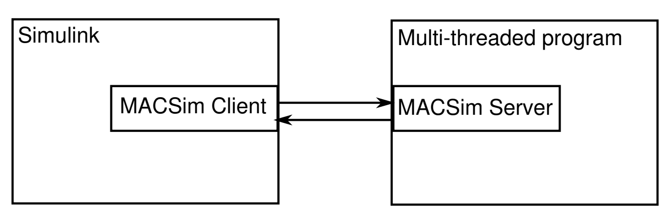

Fig. 1. Structure of MACSim.

While a developer has the option of writing their own C++ or Java agents from scratch, it is frequently more efficient to build on what is already available. Thus MACSimJX extends the functionality of the server side of MACSim to allow Simulink to interact specifically with JADE. The developer thus has the added capabilities of this agent development environment at their disposal. This extension using JADE is described next.

B. Extending MACSim to use JADE

MACSimJX provides the means, utilising JADE, to receive data from Simulink via the MACSim interface and to pass this on to relevant agents for processing. Once the agents have finished working on the data, the data must be returned to Simulink along the same channels. The agents are designed to accomplish some goal, such as optimisation of incoming data.

For this purpose it seemed logical to divide the agent model into two parts, the Agent Environment (AE) and the Agent Task Force (ATF). The environment is a transformation of the MACSim server, previously mentioned, to provide a transparent connection between Simulink and the JADE agents. It contains the ground-work required for any generic agent model spread across Simulink and a JADE program, including responsibility for passing any data between the two programs.

The other part, the ATF, contains the agents responsible for interacting with the Simulink data. Some simple protocols need to be followed by the agents of the ATF to ensure the appropriate exchange of data with the AE. For all other purposes, these agents may be developed as normal, with the behaviours and goals one may wish to see implemented in a real-life system. The arrows in Figure 2 outline the communication paths for the three sections of the complete model.

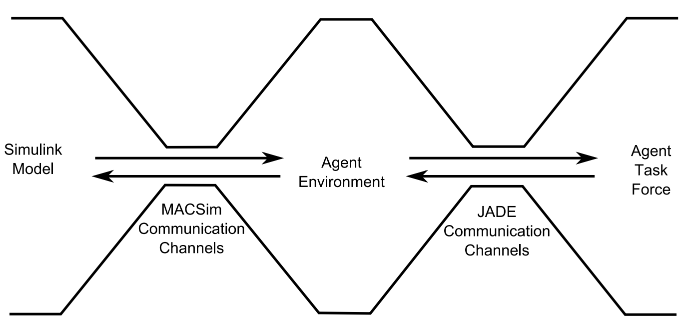

Fig. 2. Outline of the complete model.

The rest of this section considers the JADE classes for the agent environment and the basic communication standards that need to be built into agents designed for the ATF. It focuses on the characteristics of the code and classes provided by the framework to get an agent up and running. JADE has a framework where a good proportion of the underlying structure, used for implementing agents, is deliberately hidden from the user to avoid making the development cycle over-complicated.

Someone using JADE will have functions they wish their agents to perform and, at least initially, will only want to concern themselves with the programming of these functions. It should be possible to place these straight into some agent template in order to get their agents up and running. JADE provides what is termed an API (Application Programming Interface) which is effectively a library describing the different classes and functions with the parameters they require and return after running. These classes provide the backbone for agent development with JADE, at least in terms of the general agent properties, and make the whole process relatively painless. In the API it is possible to search for the properties one wishes an agent to exhibit, including those related to agent behaviours, communication methods between agents, and the resulting interaction of agents.

To assist with this rapid prototyping of agents, a template is suggested. Derived from [14] and [4], the skeletal code for this is shown in Listing 1. It can be divided into several major parts. The agent code commences by importing the various libraries of functions that include those required for use in the agent being designed. The class name of the agent type being created is then declared. Inside this are the two main functions, *setup()* and *takeDown()*, their names being sufficient to describe their purpose. Following these is a selection of inner classes, in this case one, containing the various behaviours the agent will exhibit and which are to be initialised through the setup() method. There are some basic behaviour types provided by JADE, such as the OneShotBehaviour (executes once) and the CyclicBehaviour (repeats its code continuously). These behaviours can be utilised to create customised behaviours in which the programmer places the desired agent functions and also the code for communication with other agents.

Thus, the code of Listing 1 can be filled out fairly easily, with the desired agent properties expressed inside the behaviour functions. Implementing the agent is then simply a case of executing the JADE *runtime environment* from command prompt and calling the relevant agents. Detailed examples are provided in the JADE tutorial guide [14], Programmer's Guide [4] and Administrators Guide [15].

C. The Agent Environment

The AE acts as an interface for the JADE agents and Simulink. It has been suggested [12] that such an interface should be responsible for the following:

- Keeping track of all current agents and facilitating *the dynamic 'birth' and 'death' of agents*.
- Synchronisation with Simulink through MACSim.
- Providing the current input and time step data when requested.
- Storage of data to be output back to Simulink and allowing for these data to be altered.
- Having the capability to broadcast messages to the agent population.

The first requirement indicated above is handled automatically by JADE through its DF which is, in effect, an agent that acts like the 'yellow pages' where agents register with the services they can offer and can search for those they require. Synchronisation is optional, agents can either wait for data

Code listing 1

```
import jade.lang.acl.*;
import jade.core.Agent;
import jade.core.behaviours.*;

public class SkeleAgent extends Agent {
// Initialise class variables.

  protected void setup() {
/* Attempt to initialise agent,
 * including its various behaviours.
 * Add the behaviour for receiving
 * agent messages.
 */

  addBehaviour(new AgentBehaviour1());
  Object[] args = getArguments();
  if (args != null && args.length > 0) {
  // Operate on the received parameters
  // provided by agent initiator as args.
  }
  else {  // Unable to create agent.
    System.out.println(No arguments.);
    System.out.println(Ending agent.);
    takeDown();
    }
  }

  protected void takeDown() {
  // Remove agent from system.
  }

  class AgentBehaviour1
    extends CyclicBehaviour {
    public void action() {
      ACLMessage msg = receive();
      if (msg != null) {
      // Process the message
      }
    }
  }
}
```

from Simulink before carrying out any actions, or also perform tasks while waiting for new data. The other requirements mentioned above are handled by the various components of the AE that will now be described.

The code for the agent environment is composed of seven classes, two of which represent actual agents. The core classes are the *AgentServer*, *AgentCoordinator*, *EnvironmentAttributes* and the *TimeStepData* class. The other three provide some extra functionality for designing agents and are the *TimeProfiler*, *UsefulAgentMethods* and the *Matrix* class. The methods contained by these classes are detailed in the MACSimJX API. The AE is outlined in Figure 3. Here you see a more detailed representation of the Agent Environment with the two JADE agent classes, AgentServer being responsible for interacting with Simulink and AgentCoordinator managing exchanges with the ATF.

The afore mentioned responsibility of current input and time step is handled by the EnvironmentAttributes class, along with the number of inputs to and from Simulink (these can be accessed by the relevant *get* and *set* methods). Storage of data is done in the TimeStepData class which keeps track of changes made by the agents until it is ready to be sent back to Simulink. Finally through the JADE services functionality, any agent subscribed to the ”Agent” service will receive any broadcast to this service.

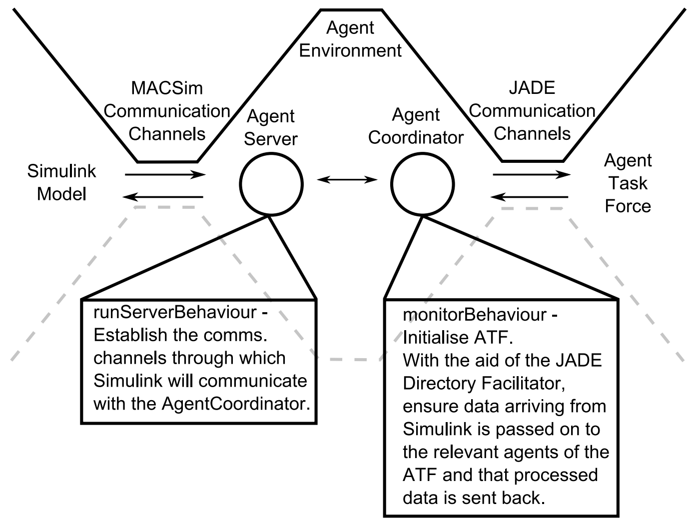

Fig. 3. An outline of the Agent Environment.

D. The Agent Task Force

The ATF consists of all the agents that jointly operate on the data arriving from Simulink in order to accomplish some task. All of these agents will have *setup()* and *takedown()* functions and probably some behaviour functions. However, with the exception of communication protocols with the AE, the implementation of the functions will take on very different forms depending on the particular task the designer wishes them to achieve.

An example is shown shortly to demonstrate the implementation of an ATF. However, there are several common features it is appropriate to draw to the reader’s attention beforehand. An overview is provided in Figure 4, where between the *First Agent* and *Last Agent*, there can be any number of other agents. Each having the ability to communicate with the other agents of the task force, and with the AgentCoordinator from the AE. The diagram depicts the general behaviour of the *Last Agent*. As an example, it is given a filter behaviour, but with the exception of applying the Kalman filtering, the same series of steps apply to the other agents.

Each designed ATF is ideally given its own *sub-package* with inner packages for each type of agent it uses. The agent package contains the agent class along with any associated classes to supplement the agent class. Inspection of the ATF diagram shows that there are also three standard messages that a designer may wish to implement in their agents. These messages being identified as:

- UpdateData - Provides the new data arriving from Simulink at each sample step.
- DataAmended - A confirmation that data amended by an agent has been received by the AE.
- Shutting Down - An instruction received from another agent to end current processes and terminate.

These three messages are primarily for interaction between the agents of the AE and ATF. It is assumed the developer would wish the freedom to determine the mechanisms of communication within their own ATF. Some of these agents may have no need to communicate with the AE, relying on others to pass on the needed data.

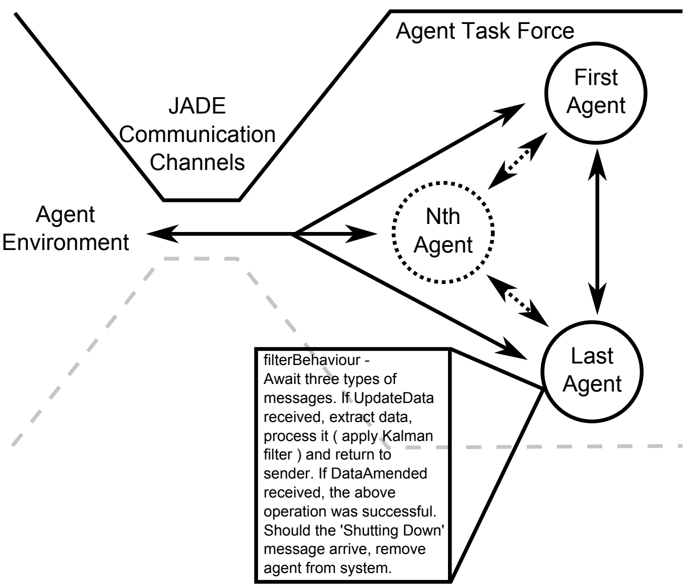

Fig. 4. Overview of the ATF.

IV. Demonstration

This section provides an example of the procedure that is followed in order to set up some agents for an ATF. For the sake of clarity a straightforward scenario is used. Two different sinusoidal-type signals are fed from Simulink through MACSimJX. Both signals are then communicated to the ATF. In this example, the ATF consists of two agents. These will apply some arithmetic to the signals, that is, one agent finds the sum of the signals and the other finds the difference, and these results are sent back to Simulink.

The initial step is to set up the model being used in Simulink. This is shown in Figure 5, where the two signal generation blocks are shown connected to the MACSim block. The MACSim block is the client that will be exchanging data with the agents. The outputs from the MACSim block are then connected to scopes for analysis. The final component of the Simulink model performs the same arithmetic operations as used by the agents. This is to provide a comparison with the agent output for the sake of validation.

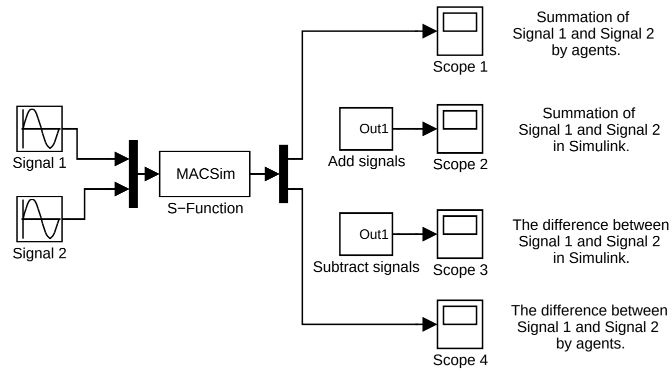

Fig. 5. Simulink example model.

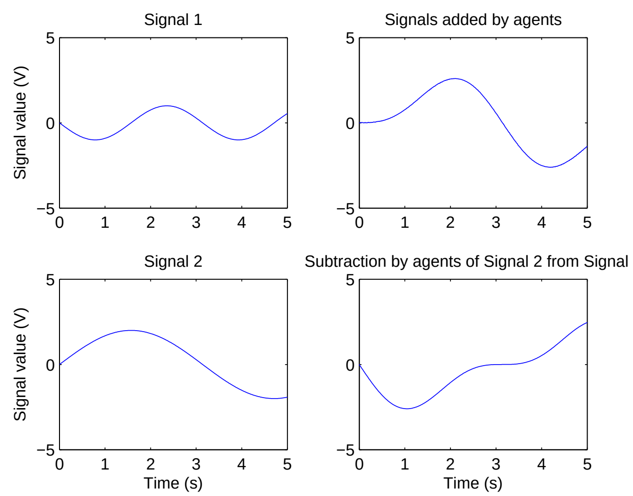

Fig. 6. Scope output.

The next stage is to develop the agents required for the ATF. This commences by customising the JADE agent template, shown in Listing 1, for use with MACSimJX, and then incorporating the arithmetic operations into the agent behaviours.

In order for agents to interact with MACSimJX, there are three message types the agent should be prepared to receive. These have the ID tags of *UpdateData*, *DataAmended* and *Shutting Down*, as mentioned in the previous section. The most important of these messages, for agent operation, is UpdateData which consists of a data structure containing an array representing the input ports of the MACSim block, and the current data sample at each of these ports. Upon receiving such a message, the agents carry out their processing, in this case some arithmetic, and then return a message to the sender, usually the AE, containing a data structure with the results and indicating the elements that were changed. To illustrate this part of the design, with an agent interacting with MACSimJX, the behaviour segment of Listing 1 is extended and shown in Listing 2.

Following the design of the agents, all that is left is to run this ATF alongside Simulink. To do this, the agent class files should be placed in an appropriately named folder under MACSimJX\ATFs. MACSimJX is then executed (by running a .jar file), which opens a GUI allowing the relevant details to be entered, such as adding the location of the agents class files, the number of inputs\outputs of the MACSim block in Simulink and the sample rate. With these details completed, the *Continue* button is clicked and our agents are now ready to operate. Finally, the Simulink model is opened, the simulation time specified and then set to run. Figure 6 shows sample data that has been run through the system for five seconds.

The TimeProfiler class provides some information about the ratio of time spent in Simulink and the AE to the time in the ATF. In the current example, due to the simplicity of the simulation, about 99 percent of the time was spent on the JADE side. TimeProfiler additionally provides methods to locate bottlenecks that might exist in a developers agent code in order to assist with this aspect.

A more advanced example of using MACSimJX may be found in [3], where a Boeing 747 is modelled in Simulink and data from its sensors are sent to agents for fusion where centralised and decentralised Kalman filters are tested.

V. CONCLUSION

An overview has been provided of a useful enabling tool called MACSimJX. The desirable nature of decentralised multi-agent-driven systems were discussed. Simulink, an industry standard program for modelling dynamic real-time systems, was introduced and its shortcomings with respect to multi-threading were described. MACSimJX provides a bridge to rectify this problem, the support taking the form of JADE.

Thus an integrated software framework is available for the development, testing and analysis of multi-agent control systems. It incorporates an interface (MACSimJX) that enables the co-simulation of dynamic systems (under Simulink) and a multi-agent system (JADE). This opens up a very wide field of investigation, in particular for the design of complex and dynamic multi-agent control systems.

ABBREVIATIONS

AE	Agent Environment.
AMS	Agent Management Service.
API	Application Programming Interface.
ATF	Agent Task Force.
DF	Directory Facilitator.
FIPA	The Foundation for Intelligent Physical Agents.
JADE	Java Agent Development Environment.

Code listing 2

```
class AgentBehaviour1
  extends CyclicBehaviour {
  public void action() {

/* Initialise data structure,
 * and other local variables.
 */
    // Prepare agent to receive a message.
    ACLMessage msg = receive();

    if (msg != null) {
    String message=msg.getConversationId();

      if (message.equals("UpdateData")) {
      //In this example, from the AE.

/* Try to extract data structure from
 * the new message, and from this,
 * the the data array and length.
 */
/* The agent then performs any data
 * operations required before other
 * ATF agents are considered.   If these
 * initial results are required by other
 * ATF agents, they are sent as a data
 * structure in a  new message.  Otherwise,
 * the agent waits until receiving all
 * expected data from other ATF agents.
 */
/* For this example the agent now finds
 * either the sum or the difference
 * between the two data array elements.
 * Having completed the calculations,
 * results are returned to the AE.
 */
      }
      if (message.equals("DataAmended")) {
      // i.e. a response from AE confirming
      // receipt of data from this agent.

      // If no further operations required,
      // end current conversation with AE.
        replyToAgent(msg.getSender(),
          "ProcessingComplete");
      }
      if (message.equals("Shutting Down")) {
        // Terminate the agent.
        takeDown(msg.getSender());
      }
    } else {
      // Put agent to sleep.
      block();
    }
  }
}
```

REFERENCES

[1] M. Wooldridge and N. R. Jennings, “Intelligent agents: theory and practice,” *The Knowledge Engineering Review*, vol. 10, no. 2, pp. 115–152, 1995.

[2] N. Jennings, K. Sycara, and M. Wooldridge, “A roadmap of agent research and development,” *Autonomous Agents and Multi-Agent Systems*, vol. 1, no. 1, pp. 7–38, 1998.

[3] C. R. Robinson, “Decentralised data fusion using agents,” Ph.D. dissertation, The University of York, 2008.

[4] F. Bellifemine, G. Caire, T. Trucco, and G. Rimassa, *Jade Programmers Guide*, TILab S.p.A., 2005.

[5] F. Bellifemine, G. Caire, A. Poggi, and G. Rimassa, “Jade - a white paper, in ”exp in search of innovation - special issue on jade,” TILAB, Tech. Rep., 2003.

[6] Z. Li, F.-Y. Wang, Q. Miao, and F. He, “An urban traffic control system based on mobile multi-agents,” in *IEEE International Conference on Vehicular Electronics and Safety*, Beijing, China, October 2006, pp. 103–108.

[7] M. Pěchouček, D. Šišlák, D. Pavlíček, and M. Uller, “Autonomous agents for air-traffic deconfliction,” in *Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems*, Hakodate, Japan, May 2006, pp. 1498 – 1505.

[8] B. Balachandran and M. Enkhsaikhan, “Development of a multi-agent system for travel industry support,” in *International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce*, Sydney, Australia, December 2006, p. 63.

[9] D. Naso and B. Turchiano, “A coordination strategy for distributed multi-agent manufacturing systems,” *International Journal of Production Research*, vol. 42, pp. 2497–2520, 2004.

[10] J. Eze, H. Ghenniwa, and W. Shen, “Distributed control architecture for collaborative physical robot agents,” in *IEEE International Conference on Systems, Man and Cybernetics*, Washington, D.C., USA, October 2003, pp. 2977 – 2982.

[11] P. Santana, V. Santos, and J. Barata, “Dsaar: A distributed software architecture for autonomous robots,” in *IEEE Conference on Emerging Technologies and Factory Automation*, Prague, Czech Republic, September 2006, pp. 1017–1020.

[12] P. Mendham and T. Clarke, “Macsim: A Simulink Enabled Environment for Multi-Agent System Simulation,” in *Proceedings of the 16th IFAC World Congress, Prague, Czech Republic*, 2005.

[13] J. M. Solanki and N. N. Schulz, “Using intelligent multi-agent systems for shipboard power systems reconfiguration,” in *Proceedings of the 13th International Conference on Intelligent Systems Application to Power Systems*, November 2005.

[14] G. Caire, *Jade Programming for Beginners*, TILab S.p.A., 2003.

[15] F. Bellifemine, G. C. amd Tiziana Trucco, G. Rimassa, and R. Mungenast, *Jade Administrator’s Guide*, JADE Board, 2006.

Children's Relationships with Robots: Robot is Child's New Friend

Meghann Fior, M.A.,Sarah Nugent, B.A., Tanya N. Beran, Ph.D., Alejandro Ramirez-Serrano, Ph.D. and Roman Kuzyk, B.Sc.

Abstract— **The purpose of this study was to examine how children think about and attribute features of friendship to a robot after a brief interaction with one. Children visiting a science centre located in a major Western Canadian city were randomly selected to participate in an experiment set up at the centre. A total of 184 children ages 5 to 16 years (M = 8.18 years) with an approximate even number of boys and girls participated. Children were interviewed after observing a traditional robot, a small 5 degree of freedom robot arm, perform a block stacking task. Content analysis was used to examine responses to nine open-ended questions. Results indicate that the majority of children were willing to engage in friendship with the robot by showing positive affiliation and social support towards it, as well as sharing activities, and communicating with it.**

Significant sex differences in how children ascribe characteristics of friendship to a robot were also found.

Index Terms— **Robotics, Developmental Psychology, Friendships, Human-Machine Relationships.**

I. INTRODUCTION

CHILDREN are becoming increasingly adept at operating computers and spend considerable time doing so. According to Statistics Canada [1], in 2000, 82% of parents reported that their children (aged 5 to 18 years) use computers [2]. Because of the increase in computer access among youth, studies have investigated the implications of this usage on their physical and psychological well-being [3-5]. Results are mixed with studies documenting adverse and positive outcomes, as well as no effects [6-8]. While it remains unclear as to how computer use is related to children's social development, research has also to examine how children's interactions with robots affect their development. With robots being built to mimic human expression and behavior it is possible that when children interact with a robot they may develop feelings of friendship towards it. The development of friendships in childhood is crucial to subsequent mental and physical health [9-11]. Thus, it is important to understand children's perceptions of friendship they may have in relation to a robot. The focus on the present study involves investigating what constitutes children's friendships and examining whether such patterns transfer to child-robot interactions.

Meghann Fior, M.A. is with Division of Applied Psychology, University of Calgary
Sarah Nugent, B.A., Division of Applied Psychology, University of Calgary
Tanya N. Beran, Ph.D., Community Health Sciences, University of Calgary
Alejandro Ramirez-Serrano, Ph.D., Department of Mechanical Engineering, University of Calgary
Roman Kuzyk, B.Sc., Department of Mechanical Engineering, University of Calgary
Correspondence to: Dr. Tanya Beran, Community Health Sciences, Faculty of Medicine, University of Calgary, 3330 Hospital Drive NW, Calgary, AB, Canada T2N 4N1, tnaberan@ucalgary.ca

II. RELATED WORK

A. Human-Robot interactions

In recent years, the course of development of robots has moved away from creating machines to work independently from humans to now creating robots for the purposes of interacting with humans in daily life [12]. In today's society, some robots function as physical aids for elderly people [13], as museum tour guides [14, 15], or as peer tutors and educational tools [16, 17]. With such a trend toward social robots, questions arise as to the extent of children's knowledge and understanding of humanistic versus robotic characteristics and how this may impact children's social relationships. According to Turkle [18] children who regularly use electronic devices (e.g., computers, video games, electronic toys) are more likely to attribute psychological characteristics to such devices, such as having the ability to talk, sing, or do activities. A recent study by Melson and colleagues [19] examined children's understanding of robotic versus living animals by comparing Sony's AIBO robotic dog to a living dog. The authors found that although more children (aged 7 to 15 years) attributed physical characteristics (i.e., mental states, sociality, and moral standing) to the live dog, the majority of children also ascribed these attributes to the robotic dog. In addition, children were as likely to give commands to the robotic dog as to the living dog. This suggests that children may treat technological devices as if they were social beings, which suggests the existence of a child-robot companionship.

B. Children's social relationships

Friendships are undoubtedly important for childhood development, and, as such, set the stage for the development of communication skills, emotional regulation, and emotional understanding [20]. Friendships that are considered to be of high quality include many prosocial behaviours and deep intimacy [21]. Several additional characteristics shared between friends include a sense of caring or fondness, emotional support, and enjoyment of activities [21, 22]. Moreover, meaningful friendships between children are based on openness, affection, mutual support and trust, as well as a willingness to share thoughts, feelings, stories, and secrets [23]. With friends children display a higher frequency of play behaviour and positive affect such as smiling and laughing [22, 24, 25]. From the aforementioned studies the primary characteristics of friendships can be summarized as a sense of positive affiliation, social support, shared activities, and communication between friends. In addition, sharing secrets plays an integral role to developing and maintaining friendships [26, 27].

Robots are under development for a wide variety of purposes that will become commercialized and available to children as toys and companions. It is important to understand children's receptivity to a robot, their thoughts/feelings towards it, and their social responsiveness towards it. This, in combination with the significance of friendship for children's healthy development, compels us to examine their perceptions of friendship with a robot. Specifically, the question concerning the present study is whether children would attribute some of the features of friendships documented in previous research, to relationships children may experience with a robot that exhibits some minimal social cues. Friendship can be characterized as demonstrating positive affiliation, social support, shared activities, and communication (including sharing secrets). We examine whether children would ascribe these same characteristics to a robot after briefly interacting with it. Also, we selected a relatively simple robot, assuming that if children provide positive responses about a robot of this type, that they would also do so with more sophisticated ones. The advantage of such assumption was to eliminate from the study a number of variables such as the robot's physical appearance and its communication skills (e.g., Aibo, NAO and Wowwee robots). By doing so we were able to focus on the intended study and provided more definite conclusions. Given that friendship is exhibited from each child to the other, we asked children about their perceptions of friendship behaviors towards the robot and from the robot.

III. METHOD

A. Sample and procedure

A total of 184 children (n = 98 female, n = 86 male) between the ages of 5 to 16 years (M = 8.18, SD = 2.37 years) were included in the study. Participants were visitors to a science centre located in the downtown area of a large city in Western Canada. Data collection occurred in the summer during opening hours from Monday to Sunday. Families with a child in the specified age range, who were visiting the science centre, were approached by a researcher and asked if their children would like to visit with a robot. Then the accompanying guardian was informed about the study and asked to sign a consent form. The researcher then escorted the child independently into the robot exhibit while the family waited at an adjacent exhibit. The response rate was approximately 95%.

The robot exhibit was a small booth 10 by 7 feet located in a quiet area of the science centre. It was built with heavy curtains and dividers designed to reduce noise and discourage interruptions by visitors. The booth contained a robotic arm on a platform with a chair facing it for the child to sit and observe the robot completing a task. There was also an adjoining space behind a divider serving as the testing booth and contained two laptops. One laptop produced diverse task commands for the researcher to control the robotic arm while performing the task. The second laptop was connected to a camera mounted on the wall behind and to the side of the robot and facing the child. This allowed researchers to observe the child on the laptop from behind the divider. Children were not informed that they were being watched through the camera, and most children did not look at it. Of those who did, some thought the camera was used to control the robot and not necessarily monitor them. Based on these observations we believe that children did not know that they were being watched because almost no one looked at, or seem to notice, the camera. We believe that this aspect of how the tests were conducted is relevant because children seem to be comfortable and natural while interacting with the robot (which was one of the intentions while conducting the study).

The researcher escorted the child behind the curtain and gave the request to be seated on the chair in front of the robot. The child was then informed that the researcher would be right back and then went behind the divider. The researcher then executed the command on the laptop to run the robot on a specific task and observed the child on the second laptop. The child's behaviours were recorded on a record form. The robot was programmed to stack blocks, and once the robot stopped, the researcher returned to the child and conducted an interview. Children were then thanked and guided back to their families.

The block stacking task was selected because children recognize this as a familiar play behavior. There is no existing research to suggest that having the robot engage in a different activity would result in a different outcome, so there was no basis to believe that this was the case. The questions were asked at the end of the sequence of movements to allow children an opportunity to focus on the robot, and then afterwards focus on the questions.

B. Description of robot

The self-contained electric D.C. servo driven robotic arm used was a CRS-Plus small 5 degree of freedom articulated arm having a base (±175˚ rotation), shoulder (+110˚, 0˚ rotation), upper (0˚, -130˚ rotation) and lower arm (±115˚ rotation), and wrist (±180˚ rotation) motions controlled by a RSC-M1A robot system controller. During the experiment the robot moved objects weighing only a few grams (i.e., small rectangular wood pieces). The robot joints include optical encoders for position feedback and a speed setting (both program and hardware) set to slower speeds for safety purposes. For added safety, children were positioned outside of the workspace of the robot (i.e., 0.56 meters) at all times.

The robotic arm was covered in craft foam and corrugated plastic to appear pleasing to look at (see figure 1). Gender neutral colors yellow, white, and black were chosen. To ensure that the robot appeared to pick up blocks with its mouth, the two finger gripper of the arm was covered with a head so that its grip was situated in the mouth. The head contained two eyes made of smooth silver buttons. Due to its design and construction, the robotic arm made a low humming noise when turned, but this was barely audible.

The rectangular blocks that the robot picked up were 2 cm x 2 cm x 4 cm. They were placed in a line to the side of the robot in the craft foam that covered the platform (see Figure 1). An outline of the blocks was cut into this foam to ensure that the blocks were correctly positioned every time for the robot grip. The arm was positioned in the center of the platform with the head raised to the height of the child, appearing to 'look' at the child.

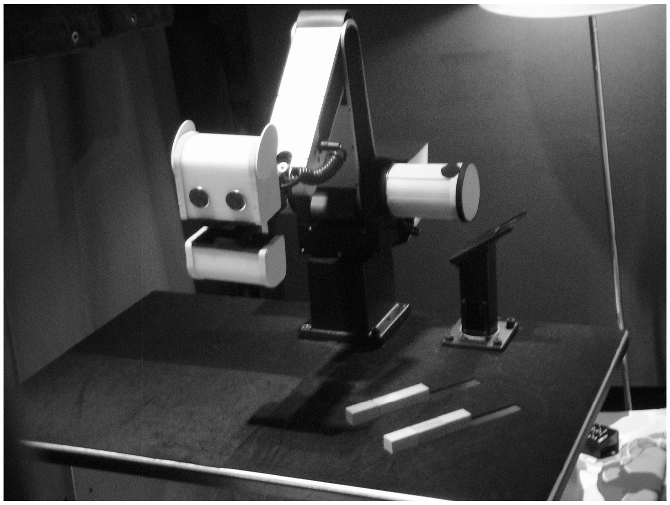

Fig. 1. 5 Degree of freedom robot arm on platform with blocks

C. Robot's Task

The robot was pre-programmed at the university and then controlled at the science centre by a researcher via a graphical user interface. The first movement of the arm was to the side where the blocks were positioned and to pick up the first block. Then the arm returned to the center with its head opposite the child's face so as to appear to be 'looking' at the child. The robot then lowered the block, placing it on the platform in front of the child. These actions were repeated with the second block to stack it on the first block. The robot grasped the third block, picked it up, but slightly opened its grip as it turned toward the child, thus, dropping it. The grip opened wider while facing the child to make the facial appearance of the mouth opening and appearing surprised at dropping the block. The arm then returned to the original location and moved back and forth for 25 seconds, to 'look' for the block it dropped. It lowered twice, attempting to pick up the block, but 'missed' both times. Then the arm returned to the center with the head raised and positioned in front of the child's face. These movements were programmed to be smooth so as not to appear as machine movements.

D. Measure

A total of nine questions were asked during a 5-10 minute interview, which took place once the robot stopped. The first three were about children's use of electronic devices at home. These include whether they watched TV or played on a computer at home, as well as whether they owned electronic toys. Responses were coded as 'yes' or 'no'. The following six questions about children's affiliation with the robot were then asked. Positive affiliation was measured by two questions: "Does the robot like you?" and "Can the robot be your friend?" Social support was assessed with one question: "If you were sad would the robot make you feel better?" Shared activities were measured by asking children: "Would you play with the robot?" Finally, communication was assessed through two questions: "Would you talk to the robot?", and "Would you tell the robot secrets?" The first three questions ask about the robot's friendship behaviors towards the child, and the latter three questions ask about the child's friendship behaviors towards the robot. After each question they were asked to explain why they indicated "yes" or "no". These responses were coded according to guidelines recommended for qualitative data [28]. One researcher on the project and a research assistant examined the responses for themes, then a coding scheme was developed, which was used to code the responses. The intraclass correlations for two raters, who coded separately, ranges from .92 to .96 across the six questions, indicating very good inter-rater reliability.

IV. Results

A. Use of electronic devices

A total of 95.5% of children (n = 169, n = 7 missing) stated they watch television, 81.9% of children (n = 145, n = 7 missing) reported playing on a computer at home, and 84.5% (n = 147, n = 10 missing) indicated they had electronic toys (e.g., robotic dog, remote control cars). Thus, the majority of children demonstrated familiarity with electronic devices.

To identify if there are any differences between children of different ages and the attitude to the robot (as would be normally expected) children were grouped by age into three groups. Each group (5-6 yrs, 7-9 yrs and 10-16 yrs) consisted of approximately of 50 children. There were no significant chi square results for age and any of the six questions about friendship characteristics.

B. Positive affiliation

More than half of the children (64.0%) stated the robot liked them (see Table 1). A frequently stated explanation for this belief is that the robot looked at them and appeared friendly (e.g., "his mouth looks like he is smiling at me"). Other children thought the robot had positive intentions (e.g., "he wanted me to know my numbers by counting blocks"). Absence of harm was another reason for thinking the robot liked them (e.g., "never tried to bite me"), and their kind actions towards the robot led them to believe the robot would like them (e.g., "I was encouraging the robot"). Few children (8.7%) stated the robot did not like them, citing reasons such as it not having the ability to think or feel, or that it ignored them by stacking the blocks and not allowing them to help it. Many children did not know if the robot liked them or were unable to explain why. Some children provided a response that did not address the question (e.g., “hard to tell with robots”). There was no significant difference between the number of girls (n = 60) compared to boys (n = 58) who thought the robot like them, X2(1) = 0.28, $p > 0.05$.

In addition to feeling liked, the majority of children (85.9%) believed that the robot could be their friend and provided a variety of explanations (see Table 1). Most of these children stated they could be friends pending the robot’s actions. For example, it would depend on whether the robot was nice, helpful, or engaged in conversation. Many other children thought that friendship was based on spending time together or participating in activities together (e.g., “watching a movie together”, “teaching me something”). The robot assisting the child or vice versa was also often mentioned, as was a sense of familiarity with (e.g., “robot knows me”) and kindness towards the child (e.g., “can hand me things”). In addition, characteristics of the robot for friendship include children’s perceptions of it being friendly and likeable. Some children also judged their friendship with a robot based on their friendly acts towards it (e.g., “saying hi to the robot”). One child made a poignant statement about friendships with robots, “Man’s best friend is a dog so a robot can be child’s best friend”. Few children (10.3%) indicated that a robot could not be their friend and explained that the robot has limited ability to move, communicate, or understand their thoughts or feelings. Some children stated they did not know if or why the robot could be their friend, and some responses did not answer the question (e.g., “every robot is my friend”). There was a significant difference found with more girls (n = 90) than boys (n = 68) saying the robot could be their friend, X2(1) = 4.40, $p < 0.05$, effect size (Φ) = 0.15.

C. *Social support*

In regards to social support, a large majority of children (78.8%) indicated that the robot could improve their mood (see Table 2). The most frequent explanation was to perform an action for them such as stacking blocks in a funny shape, or any other type of action (e.g., “making me laugh by doing something weird”). Other children stated the robot could perform an action with them such as playing together (e.g., playing games together). Many children thought the robot appeared cheerful (e.g., “gives me a smile”), and many stated it could emotionally connect with them (e.g., “be beside me”, “understands me”). Some children also stated that the robot could help them (e.g., “helping me if I’m hurt”). Fewer children (14.7%) stated the robot could not improve their mood, with most of them explaining that it has limited abilities (e.g., “can’t talk”, “only stacks blocks”, “no brain”). One boy stated the robot did not like him and so would not cheer him up. Some children did not know if the robot could cheer them up and another 25 children did not know why the robot could or could not cheer them up. Ten responses did not address the question (e.g., “just mom”). There was a significant difference found with more girls (n = 86) than boys (n = 59) saying the robot would cheer them up, X2(1) = 8.09, $p < 0.05$, effect size (Φ) = .21.

TABLE I
NUMBER AND PERCENTAGE OF CHILDREN REPORTING POSITIVE AFFILIATION WITH ROBOT (N = 184)

Robot likes you		Robot can be your friend	
Yes	118 (64.0%)	Yes	158 (85.9%)
Looks/smiles at me, friendly	38	Conditional	31
I was nice/did something nice	20	Being or doing things together	30
Did not hurt me	13		
It had positive intentions	9	Helpful	17
		Knows me	12
Do not know why	33	Kind	11
Not coded	5	Friendly	6
No	16 (8.7%)	Likeable	7
No thoughts/feelings	4	Friend to robot	4
Ignored me/didn’t let me help	10	Do not know why	28
Do not know why	2	Not coded	12
Not coded	0	No	19 (10.3%)
Do not know	50 (27.3%)	Limited mobility	3
		Limited communication	2
		No familiarity	3
		No brain, feelings	4
		Do not know why	4
		Not coded	3
		Do not know	7 (3.8%)

D. *Shared activities*

Similarly, the vast majority of children (83.7%) stated they would play with the robot and provided a variety of ideas about how they would play together (see Table 2). Most often mentioned were games of construction such as building towers and castles with toy building blocks. Several active types of games were also suggested by many children including playing catch or fetch with a ball, and running. Less physically inten-

TABLE II
NUMBER AND PERCENTAGE OF CHILDREN REPORTING SUPPORT AND ACTIVITIES WITH ROBOT (N = 184)

Robot can cheer you up		Play with robot*	
Yes	145 (78.8%)	Yes	154 (83.7%)
Perform action for me	61	Construction	103
Perform action with me	12	Ball game	26
Cheerful appearance	20	Running game	12
Connects with me	20	Board game	12
Help me	7	Other	17
Do not know why	17	Do not know why	5
Not coded	8	Not coded	5
No	27 (14.7%)	No	25 (13.6%)
Limited abilities	16	Physical limitation	11
Does not like me	1	Other	4
Do not know why	8	Do not know why	6
Not coded	2	Not coded	4
Do not know	12 (6.5%)	Do not know	5 (2.7%)

TABLE III
NUMBER AND PERCENTAGE OF CHILDREN REPORTING COMMUNICATION WITH ROBOT (N = 184)

Talk to robot		Tell robot secrets	
Yes	124 (67.4%)	Yes	84 (45.7%)
I like the robot	16	Robot will keep secret	30
To get to know each other	6	Friendship with robot	13
Robot has mouth	6	Positive response to secret	7
If robot could talk	22	Other	4
Gave examples	30		
Do not know why	37	Do not know why	22
Not coded	7	Not coded	8
No*	53 (28.8%)	No	92 (50.0%)
Robot cannot talk	20	Secrets are wrong	24
Robot cannot hear	6	Robot has limitations	18
Not human	5	Robot not trustworthy	24
Looks unfriendly	9	Robot is not alive	9
Do not know why	11	Do not know why	12
Not coded	4	Not coded	5
Do not know	7 (3.8%)	Do not know	8 (4.3%)

*Some children provided more than one reason

sive games were also identified such as playing board games. Several other suggestions were provided such as video games, coloring, hand games, or riding on the robot. Few children (13.6%) stated they would not play with the robot with most of them stating it was because it had a limitation such as no legs or arms. Other reasons include "not one of my interests", and "not this one". Some answers did not address the question (e.g., "can be best friend"), and some children did not know how to answer the question. There was no significant difference between the number of girls (n = 80) and boys (n = 74) who stated they would play with the robot, $X2(1) = 0.88$, $p > 0.05$.

E. *Communication*

More than half of the respondents (67.4%) indicated they would talk to the robot (see Table 3). Many stated they would do so because they like the robot (e.g., "it looks friendly"), or to become acquainted with it (e.g., "so get to know me better"). Some children believed that its physical appearance of a mouth would be reason to talk with it (e.g., "he has a mouth and me too and we can talk"). Many children stated the condition that if the robot could talk, then they would talk. Other children who stated they would talk to the robot provided examples (e.g., "you're good at building blocks", "how's it going?"). More than a quarter of the children (28.8%) stated they would not talk to the robot. Many stated that it is because the robot cannot talk (e.g., "doesn't speak English, speaks robot talk"), or hear. Some children stated that it is not human or alive, and others stated that it does not look friendly. Several children stated they did not know if or why they would or would not talk to the robot, and several provided a response that did not address the question (e.g., "most robots talk"). There was a significant difference found with more girls (n =

70) than boys (n = 54) saying they would talk to the robot, X2(1) = 18.56, $p < 0.05$, effect size (Φ) = 0.32.

F. Secrets

Almost half of the children (45.7%) stated that they would tell the robot secrets and provided a variety of reasons (see Table 3). Most of them thought the robot would not tell the secret or could not because of the inability to speak. Other children provided a reason that indicated an affiliation with the robot (e.g., "he's friendly", "he's my friend"). Some children thought the robot would respond positively to the secrets (e.g., "robot would remember them", or "robot would forget them so would not repeat them", "the robot will tell me some secrets"). Other responses include telling the robot secrets if they had no other friends, and that it feels good to tell secrets. Half of the children (50.0%) stated they would not tell the robot secrets. Several of them stated it is wrong to tell secrets, that they should be kept private. Others stated that the robot has limitations preventing them from sharing secrets (e.g., "robot can't listen or understand"), or that the robot is not trustworthy (e.g., "can't trust robot, robot might tell"). Some children stated the robot is not alive or that it does not care about secrets. Several children did not know why they would or would not tell the robot secrets or did not provide an applicable response (e.g., "depends on type of secret"). There was a significant sex difference showing that more girls (n = 59) than boys (n = 25) would tell the robot secrets, X2(1) = 19.52, $p < 0.05$, effect size (Φ) = 0.33. Given that 24 children stated they would not tell secrets to anyone, we examined whether most of them were boys, as a possible explanation for why more girls would tell the robot secrets. There was no significant difference in the number of boys compared to girls who thought secrets should not be told, X2(1) = 1.49, $p > 0.05$.

To determine the extent to which the different types of relationship characteristics are related, Pearson's Product Moment Correlation analyses were conducted (see Table 4). Children who thought the robot could be their friend were also likely to report that they would play with it, talk to it, tell it secrets, and that the robot could cheer them up and likes them. Many of these variables were low to moderately inter-correlated. Moreover, these results suggest that children who stated they would engage in these behaviors towards a robot, were also likely to state that robots could engage in these behaviors towards them. Thinking that the robot liked them was not significantly related to whether they would talk to the robot or that it could improve their mood.

TABLE IV
SPEARMAN'S RANK CORRELATION COEFFICIENTS OF FRIENDSHIP CHARACTERISTICS (N = 184)

	1.	2.	3.	4.	5.	6.
1. Likes you	1.00					
2. Friend	.34**	1.00				
3. Cheer up	.16	.49**	1.00			
4. Play	.20*	.36**	.16*	1.00		
5. Talk	.09	.35**	.40**	.23**	1.00	
6. Tell secrets	.17*	.31*	.31**	.20**	.34**	1.00

V. Discussion

It is plausible that in the future people will spend a significant amount of time with robots. These robots, moreover, will likely emanate various social cues that are familiar to people, which may facilitate people-robot interactions [29]. This raises numerous complex questions about the nature of the interactions people will have with these robots. We foray into this topic by examining children's perceptions of friendship with a robot that displays minimal social cues. We asked them if they would engage in friendship-type behaviors with one. The majority of children responded affirmatively to questions about exhibiting friendship towards the robot in the form of sharing activities (playing with the robot), and communicating (talking to the robot and sharing secrets). Moreover, more than half of the children would recognize friendship characteristics about the robot that include a sense of affiliation (robot likes them and could be their friend) and support (robot would cheer them up). The extent to which these characteristics are related to friendship was also examined: children who thought the robot could be their friend were also likely to report that they would play with it, talk to it, tell it secrets, and that the robot could cheer them up and likes them.

A. Positive affiliation

In regards to a positive affiliation with the robot, almost two thirds of the children thought the robot liked them. The predominant reason for this belief was that the robot appeared friendly. For example, some children stated it appeared to look at and smile at them. This may suggest interest and curiosity in the child, which, according to Kohn and Rosman [30] is a characteristic of friendship. In addition, the child's own positive behaviors towards the child (e.g., helping stack the third block) and absence of harmful behaviors may have directly impacted children's perceptions of liking the robot - "I helped the robot, therefore, I must like the robot.", as suggested by Gambrill [31]. That children ascribed positive intentions to the robot was rather surprising. This suggests that many children believed the robot was autonomous and deliberately showing kindness towards them even though it executed all tasks via a specific prerecorded set of programs.

In addition to thinking the robot liked them, more than three quarters of the children stated the robot could be their friend. This could be interpreted in two ways. It may suggest that children believe the robot was capable of being their friend. It is also possible that children believed that they would be or are capable of being friends with the robot. This latter possibility is suggested by the response given by four children that they would be a friend to the robot. It is interesting that many reasons children articulated for friendship with the robot are those that were asked in the interview (after they stated why the robot could be their friend). These reasons include doing activities together, helping each other, kindness, likeability, and shared understanding ("robot knows me"). Interestingly, some children who believed the robot could be their friend stated that they thought the robot did not like them or that they did not know if the robot liked them. This suggests that how children define their friendships is not solely dependent on feeling liked but on other friendship characteristics as well. Future research could also explore whether children who viewed the robot as a friend had a greater propensity towards friendships with others than those who did not view it as a friend. There was also a significant sex difference whereby more girls than boys thought the robot could be their friend. This effect size was small, however, and research on peer friendships does not suggest that girls have more friends than boys [32]. Perhaps girls have a greater interest in the robot, suggesting greater curiosity or inclination to explore friendship possibilities.

B. Social support

Children were asked if the robot could provide support in the form of improving their mood if they felt sad. This type of prosocial behavior is typically seen in friendships [21]. More than three quarters of the children did believe that the robot could improve their mood. The most often mentioned means of doing so is by the robot doing something for them such as stacking blocks. This action may provide distraction from negative feelings [33]. Some children identified that its friendly appearance, emotional connection or physical proximity would improve their mood. These have been shown to improve health outcomes [34]. More girls than boys believed the robot could improve their mood. This result is consistent with research on child-child friendships. That is, girls tend to be more prosocial in their friendships than are boys [35, 36].

C. Shared activities

More than three quarters of the children believed that they could play with the robot, and provided a variety of ideas of what they could play together. The most commonly mentioned type of game involved construction most likely due to the nature of the task the robot was performing. That so many children in our study believed the robot could play with them and proposed means of play with the robot, suggests that they are willing to include robots in their world of imagination and social-emotional expression, which play activities are known to provide [37-39]. Furthermore, play is a cornerstone of children's friendships which shows that they have friendship aspirations for the robot to include them in this world.

D. Communication

The final aspect of friendship that was examined was talking to the robot and sharing secrets. About two thirds of the children stated they would talk to the robot. Their explanations generally involved sharing examples of what they would say to it ("How's it going?), and liking the robot or wanting to become better acquainted, which again shows an interest in friendship. Indeed, these are types of social behaviors expected among friends [21]. When asked if children would tell the robot secrets, more than a third said that they would. Those who would share a secret said they would do so because they believed that the robot would not share the secret since it cannot speak. This is a very practical response and is likely a direct result of their observation of the robot not speaking. Several other children, however, mentioned they would share a secret with the robot because it seemed friendly or was their friend. This suggests that children may be willing and desire to share secrets with the robot as a means of creating a social bond with it. Of those who would not share a secret, many replied that it is wrong to tell secrets and that they should be kept private. Not surprising was the finding that more girls than boys would talk to the robot and tell it secrets. Research has shown that girls generally tend to talk more with their friends than do boys and engage in sharing secrets with their peers as a means of bonding [26, 27, 40, 41].

Although the majority of children responded affirmatively to the questions about friendship, some children responded to the contrary. The most frequent reason was because of the robot's limitations such as the absence of thoughts or feelings. This suggests that these children consider the robot to be a machine rather than human and recognize its true abilities. Reasons for these different perceptions of the robot have not yet been explored in the research but may plausibly include variation in children's knowledge of the mechanics of robots.

Across all questions about friendship a significant proportion of children provided ambiguous or uncertain responses. For example, some stated they did not know how to respond to the question or could not explain why they answered affirmatively or negatively to the question. Moreover, many children provided conditional responses based on the robot's abilities and behaviors. This may reflect the perplexity between understanding the robot as a machine, and recognizing its social behaviors such as "looking at the child", performing a task for the child, and appearing to need help for stacking the blocks. Perhaps some children neither have a well developed understanding of how robots function and the concept of programming. Rather, they may simply project their own understanding of people's behaviors based on their experiences of interacting with people. Over time, and as a result of interactions with robots, children may develop a new system or schema of understanding, and subsequent vocabulary to articulate their sense of friendship with a robot, that is likely distinct from their friendships with children.

E. Limitations

Although our exploratory study provides evidence of characteristics of children's friendships applicable to child-robot relationships, there are some limitations. First, children experienced a brief interaction with the robot which may have created some initial excitement that may not be maintained over a longer period, which more accurately reflects a friendship. Related to this is the inconsistency in the research as to a definition and description of what constitutes friendship. Indeed, Fabes, Martin, and Hanish [42] describe measurement of friendship as consisting of "a diverse array of conceptually and empirically based constructs designed to measure what they [researchers] consider to be key components of children's peer behaviors and interactions" (p. 48). Thus, alternate characteristics other than those used in the present study should be explored in future research. Moreover, the maintenance and regulation of friendship was outside the scope of this study and are interesting topics for future research. Second, it is possible that earlier questions may have influenced answers to later questions. For example, after children were asked if the robot liked them, some children stated that the robot was their friend because it appeared to like them. Thus, replicating a study with a different order of questions that are less leading could strengthen our findings. Third, results of our study are based on children's own reports of their sense of friendship with a robot. Although this is the predominant means of researching friendship, these results must be substantiated with observations of children friendship-based behaviors towards a robot [43]. Also, it is possible that a social desirability effect occurred whereby children felt compelled to respond favorably to the questions about the robot. It would be worthwhile in future research to determine if children would respond similarly about the robot to someone who was seemingly unrelated to the robot exhibit. Fourth, children observed a robot conduct a task unsuccessfully, thereby eliciting a possible need for assistance from the child. This type of engagement, although prevalent in child-child relationships, may have created a sense of vulnerability and inclination towards friendship with the robot. Replication with other robots, differing tasks, and in a context outside of the science centre is needed. Our robot was not as sophisticated as more recently developed and more expensive robots, so it is rather remarkable that children held thoughts in favor of friendship towards it.

The method of our study is based on the premise that a willingness to engage in activities together with a robot, communicate including sharing secrets, and feeling a sense of affiliation with a robot suggest that children would befriend one. Perhaps children can have these perceptions about a robot without having feelings of friendship. To explore this possibility we examined the degree of association between children's willingness to have a robot as a friend and the aforementioned friendship-type characteristics. We found a low to moderate relationship whereby children who thought the robot could be their friend were also likely to state they believed the robot liked them and could cheer them up, and that they would play, talk, and share secrets with the robot. Moreover, the majority of children did state they thought the robot could be their friend. Thus, we conclude that many children may befriend a robot given the large number of children who responded affirmatively to our questions, while future research must examine whether children actually do befriend a robot. In addition, we cannot conclude from these results that children's experiences of friendship with a robot are similar to those with another child. Research has yet to explore similarities/differences between child-robot and child-child friendships. Moreover, friendships are reciprocal [21]. Although we included questions about the child's friendship behaviors towards the robot, and the robot's friendship behaviors towards the child, this complex bi-directional relationship warrants considerable research. Moreover, we wonder if dyadic friendships [44] between children can somehow be experienced in the future as a reciprocal emotional commitment with a robot that creates a sense of interdependence and provides a source of security. Our preliminary study into this massive issue suggests that the answer is possibly.

Our study demonstrates that children are willing to perceive themselves as befriending robots – that is, as social beings. The majority of children believed that the robot liked them and could be their friend. Furthermore, most children stated they would engage in friendship-like activities with the robot such as play and telling secrets. Overall, the children in our study held the belief that robots are friendly entities that will not only provide entertainment and support, but are worthy of children's affection and communication. Robotic devices sold to children are becoming more technologically and socially advanced. Our study suggests that children will readily accept these types of devices as companions or friends even when they exhibit minimal social cues.

ACKNOWLEDGMENT

We give special thanks to the TELUS World of Science - Calgary for collaborating with us. This research would not have been possible without their support

REFERENCES

[1] Statistics Canada, "Television Viewing, by Age and Sex, By Province," Retrieved from http://www40.statcan/gc/cal01/cst01/arts23-eng.htm, 2004.

[2] W. Clark, "Kids and Teens on the Net," Retrieved from http://www.statcan.gc.ca/kits-trousses/social/edu04_0096-eng.htm, Canadian Social Trends, 11, Autumn 2001.

[3] American Psychological Association, "Internet Paradox:. A Social Technology That Reduces Social Involvement and Psychological Well-Being?" American Psychologist, 53, 1017- 1031, 1998.

[4] J. Cantor & M. Mares, "Effects of Television on Child and Family Emotional Well-Being." In B. Jennings & B. J. Alison (Eds.), Television and the American Family, 2nd ed, pp. 317-332, Mahwah, NJ: Lawrence Erlbaum Associates, 2001.

[5] K. Yeora, "The Impact of The Internet on Children's Daily Lives: Physical, Social and Psychological Well-Being," (Doctoral dissertation). Available from ProQuest Dissertations & Theses database. (Publication No. AAT 0805053), 2003.

[6] R. Kraut, M. Patterson, V. Lundmark, S. Kiesler, T. Mukopadhyay, & W. Scherlis, "Internet Paradox: A Social Technology That Reduces Social Involvement and Psychological Well-Being?" American Psychologist, 53, 1017-1031, 1998.

[7] R. Kraut, S. Kiesler, B. Boneva, J. Cummings, V. Helgeson, & A. Crawford, "Internet Paradox Revisited," Journal of Social Issues, 58, 49-74, 2002.

[8] M.K. Shields, & R.E. Behrman, "Children and Computer Technology: Analysis and Recommendations," The Future of Children, 10(2), 4-30,

2000.http://support.sony-europe.com/aibo/1_1_3_aibo_story.asp?language=en
[9] J.G. Parker, & S.R. Asher, "Peer Relations and Later Personal Adjustment: Are Low Accepted Children At Risk?" Psychological Review, 102, 357-389, 1987.
[10] J.G. Parker, K.H. Rubin, J.M. Price, & M.E. De Rosier, "Peer Relationships, Child Development, and Adjustment: A Developmental Psychopathology Perspective." In D. Cicchetti & D. J. Cohen (Eds.), Developmental Psychopathology, Vol. 2: Risk, Disorder, and Adaptation. New York: Wiley, 1995.
[11] D.W. Nangle, & C.A. Erdley, (Eds.), The Role of Friendship In Psychological Adjustment, New Directions for Child and Adolescent Development, Vol. 21, San Francisco, CA: Jossey-Bass, 2001.
[12] A. Kerepesi, E. Kubinyi, G.K. Jonsson, M.S. Magnusson, & A. Miklósi, "Behavioural Comparison of Human-Animal (Dog) and Human-Robot (AIBO) Interactions," Behavioural Processes, 73, 92-99, 2006.
[13] J. Pineau, M. Montemerlo, M. Pollack, N Roy, & S. Thrun, "Towards Robotic Assistants in Nursing Homes: Challenges and Results," Robotics and Autonomous Systems, 42, 271-281, 2003.
[14] W. Burgard, A.B. Cremers, D. Fox, D. Hähnel, G. Lakemeyer, D. Schulz, W. Steiner, S. Thrun, "Experiences With an Interactive Museum Tour-Guide Robot," Artificial Intelligence, 114, 3-55, 1999.
[15] I.R. Nourbakhsh, J. Bobenage, S. Grange, R. Lutz, R. Meyer, & A. Soto, "An Affective Mobile Robot Educator With a Full-Time Job." Artificial Intelligence, 114, 95-124, 1999.
[16] A. Billard, "Robota: Clever Toy and Educational Tool," Robotics and Autonomous Systems, 42, 259-269, 2003.
[17] T. Kanda, T. Hirano, & D. Eaton, "Interactive Robots as Social Partners for Children: A Field Trial," Human-Computer Interaction, 19, 61-84, 2004.
[18] S. Turkle, "Life On the Screen: Identity in the Age of the Internet," NY: Simon & Schuster, 1995.
[19] G.F. Melson, P.H. Kahn Jr., A. Beck, B. Friedman, T. Roberts, E. Garrett, & B.T. Gill, "Children's Behaviour Toward and Understanding of Robotic and Living Dogs," Journal of Applied Developmental Psychology, 30, 92-102, 2009.
[20] Salkind, N. J., Encyclopedia of Educational Psychology, Vol. 2, CA: Sage, 2008.
[21] T.J. Berndt, "Friendship Quality and Social Development," Current Directions in Psychological Science, 11(1), February 2002. Retrieved from www.psychologicalscience.org/journals.
[22] A. Newcomb, & C. Bagwell, "Children's Friendship Relations: A Meta-Analytic Review, Psychological Bulletin, 117, 306-347, 1995.
[23] T. Heiman, "Quality and Quantity of Friendship: Students' and Teachers' Perceptions," Journal of School Psychology International, 21 (3), 265-280, 2000.
[24] J. Nelson & F.E. Aboud, "The Resolution of Social Conflict Between Friends," Child Development, 56 (4), 1009-1017, 1985.
[25] S.D. Simpkins & R.D. Parke, "Do Friends and Non-Friends Behave Differently? A Social Relations Analysis of Children's Behavior," Merrill-Palmer Quarterly, 48 (3), 263-283, 2002.
[26] K.J. Rotenberg, "Same Sex Patterns and Sex Differences in the Trust-Value Basis of Children's Friendships," Sex Roles, 15(11/12), 614-628, 1986.
[27] D. Tannen, "Put Down That Paper and Talk To Me: Rapport-Talk and Report-Talk," In L. F. Monaghan & J. E. Goodman (Eds.), A Cultural Approach to Interpersonal Communication: Essential Readings, 179-194, Oxford, UK: Blackwell Publishing, 2007.
[28] J. Corbin, & A. Strauss, "Basics of Qualitative Research," 3rd ed., Thousand Oaks, CA: Sage, 2008.
[29] C.L. Breazeal, "Designing Sociable Robots," Massachusetts Institute of Technology, 2002.
[30] M. Kohn, & B.L. Rosman, "A Social Competence Scale and Symptom Checklist for the Preschool Child," Developmental Psychology, 6, 430-444, 1972.
[31] E.D. Gambrill, "Social Work Practice: A Critical Thinker's Guide," 2nd ed., New York: Oxford University Press, 2006.
[32] A.J. Rose, & R.L. Smith, "Sex Differences in Peer Relationships," In K. Rubin, W. M. Bukowski, & B. Laursen (Eds.), Handbook of Peer Interactions, Relationships, and Groups, pp. 379-393, New York: Guilford, 2009.
[33] C.R. Snyder, "The Psychology of What Works," New York, Oxford University Press, 1999.
[34] H.T. Reis, P. Franks, "The Role of Intimacy and Social Support in Health Outcomes: Two Processes or One?" Personal Relationships, 1, 185-197, 2005.
[35] W.M. Bukowski, B. Hoza, & M. Boivin, "Measuring Friendship Quality During Pre and Early Adolescence: The Development and Psychometric Properties of the Friendship Qualities Scale," Journal of Social and Personal Relationships, 11, 471-484, 1994.
[36] J.G. Parker, & S.R. Asher, "Friendship and Friendship Quality in Middle Childhood: Links With Peer Group Acceptance and Feelings of Loneliness and Social Dissatisfaction," Developmental Psychology, 29, 611-621, 1993.
[37] A.D. Pellegrini & P.K. Smith, "Physical Activity Play: The Nature and Function of a Neglected Aspect of Play," Child Development, 69 (3), 577-598, 1998.
[38] J. Shonkoff & D. Phillips, "From Neurons to Neighbourhoods," Washington D.C, National Academy Press, 2000.
[39] C.S. Tamis-LeMonda, J.D. Shannon, N.J. Cabrera & M.E. Lamb, "Fathers and Mothers at Play With Their 2 and 3 Year Olds: Contributions to Language and Cognitive Development," Child Development, 75 (6), 1806-1820, 2004.
[40] M. Raffaelli, & E. Duckett, "We Were Just Talking…:Conversations in Early Adolescence," Journal of Youth and Adolescence, 18, 567-582, 1989.
[41] A.J. Rose, "Co-Rumination in the Friendships of Girls and Boys, Child Development, 73, 1830-1843, 2002.
[42] R.A. Fabes, C.L. Martin & L.D. Hanish, "Children's Behaviors and Interactions With Peers," In K. H. Rubin, W. M. Bukowski, & B. Laursen (Eds.), Handbook of Peer Interactions, Relationships, and Groups, pp. 45-62, New York: Guilford, 2009.
[43] K.H. Rubin, W.M. Bukowski, & J.G. Parker, "Peer Interactions, Relationships, Groups," In W. Damon & R. M. Lerner (Editors-in-chief), N. Eisenberg (Vol. Ed.), Handbook of Child Psychology, 6th ed.: Vol. 3, Social, Emotional, and Personality Development, pp. 571-645, New York: Wiley, 2006.
[44] T.A. Rizzo, "Friendship Development Among Children in School," Westport, CT: Ablex, 1989.

Towards Semi-Supervised Learning of Semantic Spatial Concepts for Mobile Robots

Jesus Martinez-Gomez and Barbara Caputo

***Abstract*—The ability of building robust semantic space representations of environments is crucial for the development of truly autonomous robots. This task, inherently connected with cognition, is traditionally achieved by training the robot with a supervised learning phase. We argue that the design of robust and autonomous systems would greatly benefit from adopting a semi-supervised online learning approach. Indeed, the support of open-ended, lifelong learning is fundamental in order to cope with the dazzling variability of the real world, and online learning provides precisely this kind of ability. Here we focus on the robot place recognition problem, and we present an online place classification algorithm that is able to detect gap in its own knowledge based on a confidence measure. For every incoming new image frame, the method is able to decide if (a) it is a known room with a familiar appearance, (b) it is a known room with a challenging appearance, or (c) it is a new, unknown room. Experiments on ImageCLEF database and a subset of the challenging COLD database show the promise of our approach.**

***Index Terms*—place recognition, semantic place representation, online learning, kernel methods.**

I. INTRODUCTION

WHO wouldn't want a robot at home to make the daily chores? It could bring you a beer from the fridge, do the laundry, iron the shirts, collect things from the floor before cleaning, etc. A major requirement for having robots at home is that their representation of space, objects, and more generally concepts must at least partially overlap with our own. A vast literature in cognitive psychology (see [1] and reference therein) shows clearly that humans explain and categorize perceived multi-sensory patterns using semantic representations, of which language represents the synthesis. To fix ideas, let us focus here only on the semantic representation of space. We refer to rooms, and talk about them, in terms of their visual appearance (the corridor), the activities we usually perform in them (the fitness room) and the objects they contain (the bedroom). If we want to share our daily environment with robots, we need to share with them our own representation and understanding of it.

How do we make a robot learn the typical semantic space representation of humans? Robots have perceptual channels and cognitive abilities very different from our own. For instance, the typical service robot will use laser range scanners and an omnidirectional camera to collect data about an indoor place like an office environment. If programmed to learn the environment autonomously, i.e., in an unsupervised manner, the robot's interpretation of the data will result in a space representation very different from that of humans. Therefore, to make a robot have our own semantic representation of space, it is necessary to have a learning phase supervised by the user.

But how long should this supervised learning phase be? The current mainstream approaches (see Section II for a brief review of the relevant literature) assume a training phase, well separated from the actual working of the robot, where the human labels the data. Training usually stops when it is achieved a pre-defined threshold level of performance on a validation set of data, or when the user decides it. From that moment on, the robot is on its own. We argue that this approach is doomed to fail: rooms change around us continuously over time as furniture is added, replaced or relocated. It is impossible to predict how a user is going to redecorate its living room in the future, and therefore it is impossible to train the robot beforehand on such data.

Our vision is that the supervised learning mode should always be accessible to the robot, and it should be triggered by its ability to explain the incoming data. The transition from fully supervised to unsupervised should be smooth, robot driven, and competence-based. In other words, our vision is that semi-supervised online learning should become the mainstream approach for enabling robots to learn semantic concepts.

To move towards this goal, here we present an algorithm able to learn semantic spatial concepts in an open ended fashion, i.e. continuously updating its internal model with a bounded memory growth. The robot switches from a fully autonomous, unsupervised learning phase to a supervised one (where assistance by a human teacher might be required) on the basis of its capability to interpret the data with a high degree of confidence. The capability to detect hard-to-explain incoming data is done at the classifier level, frame by frame, as well as at a higher level, by exploiting the temporal continuity of the image sequences. This permits to distinguish between challenging instances of a known spatial concept (a view of the known class kitchen where it is perceived for the first time a new piece of furniture) and a new concept (a room never seen before).

Concretely, our algorithm consists of two components: the first is an online learning algorithm with performance comparable to that of the batch method and a bounded memory growth; the second is a mechanism for assigning labels to incoming data, detecting challenging frames imaging known

Jesus Martinez-Gomez is with the I3A Research Institute, Campus Universitario s/n, 02071, Albacete, Spain. E-mail: jesus_martinez@dsi.uclm.es

Barbara Caputo is with the Idiap Research Institute, Rue Marconi 19, 1920 Martigny, Switzerland. E-mail: bcaputo@idiap.ch

This work was supported by the Spanish "Junta de Comunidades de Castilla-La Mancha" under PCI08-0048-8577 and PBI-0210-7127 projects (J. M.-G.) and by the SS2Rob project (B.C.). The support is gratefully acknowledged.

concepts and ultimately recognizing when being in a whole new room. We take an discriminative approach and we build on previous work on online learning [2], [3] and confidence-based place classification [4]. Experiments on a subset of the challenging COLD database [5] and on the database used for the Robot vision Task at the ImageCLEF 2010 challenge evaluation (www.imageclef.org) show promising results.

The rest of the paper is organized as follows: after a brief review of the related literature, we describe the two components of our approach: the online learning algorithm (Section III) and the detection of confidence/ignorance (Section IV). Section VI describes our experimental setup, while section VII reports our experimental findings. We conclude with an overall discussion and possible future avenues for research.

II. Related Works

The ability to learn and interpret complex sensory information based on previous experience, inherently connected with cognition, has been recognized as crucial and vastly researched [6], [7], [8]. In most cases, the recognition systems used are trained offline, i.e., they are based on batch learning algorithms. However, in the real dynamic world, learning cannot be a single act. It is simply not possible to create a static model which could explain all the variability observed over time. Continuous information acquisition and exchange, coupled with an ongoing learning process, is necessary to provide a cognitive system with a valid world representation.

In the last few years, the need for solutions to such problems as the robustness to long-term dynamic variations, or the transfer of knowledge, is more and more acknowledged. In [7], the authors tried to deal with long-term visual variations in indoor environments by combining information acquired using two sensors of different characteristics. In [9], the problem of invariance to seasonal changes in appearance of an outdoor environment is addressed. Clearly, adaptability is a desirable property of a recognition system. At the same time, Thrun and Mitchell [10], [11] studied the issue of exchanging knowledge related to different tasks in the context of artificial neural networks and argued for the importance of knowledge-transfer schemes for lifelong robot learning. Several attempts to solve the problem have also been made from the perspective of Reinforcement Learning, including the case of transferring learned skills between different RL agents [12], [13].

III. Step 1: Memory Controlled Online Learning and Recognition of Visual Places

This section describes the first component of our overall approach, namely an online learning algorithm with a bounded memory growth and an accuracy comparable to the classic, off-line method. We take a discriminative approach, and derive an approximate version of the Online Independent-SVM. As opposed to the original algorithm, our approach does not require to store all incoming data but it allows to discard most of them in a principled manner. This leads to a bounded memory growth, where the upper bound is set by the user and the lower bound by theoretical constraints. In the rest of this section we first review basic concepts on SVM (section III-A), then we summarize the OI-SVM algorithm (section III-B). Our Memory Controlled OI-SVM is described in section III-C.

A. Support Vector Machines

Due to space limitations, this is a very quick account of SVMs — the interested reader is referred to [14] for a tutorial, and to [15] for a comprehensive introduction to the subject. Assume $\{\mathbf{x}_i, y_i\}_{i=1}^{l}$, with $\mathbf{x}_i \in \mathbb{R}^m$ and $y_i \in \{-1, 1\}$, is a set of samples and labels drawn from an unknown probability distribution; we want to find a function $f(\mathbf{x})$ such that $sign(f(\mathbf{x}))$ best determines the category of any future sample $\mathbf{x}$. In the most general setting,

$$f(\mathbf{x}) = \sum_{i=1}^{l} \alpha_i y_i K(\mathbf{x}, \mathbf{x}_i) + b \tag{1}$$

where $b \in \mathbb{R}$ and $K(\mathbf{x}_1, \mathbf{x}_2) = \Phi(\mathbf{x}_1) \cdot \Phi(\mathbf{x}_2)$, the kernel function, evaluates inner products between images of the samples through a non-linear mapping Φ. The α_is are Lagrangian coefficients obtained by solving (the dual Lagrangian form of) the problem

$$\begin{array}{rl} \min\limits_{\mathbf{w},b} & \frac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^{l}\xi_i^p \\ \text{subject to} & y_i(\mathbf{w}\cdot\mathbf{x}_i + b) \geq 1 - \xi_i \\ & \xi_i \geq 0 \end{array} \tag{2}$$

where $\mathbf{w}$ defines a separating hyperplane in the feature space, i.e., the space where Φ lives, whereas $\xi_i \in \mathbb{R}$ are slack variables, $C \in \mathbb{R}^+$ is an error penalty coefficient and p is usually 1 or 2. In practice, most of the α_i are found to be zero after training; the vectors with an associated α_i different from zero are called support vectors. Notice that, from (1), the testing time of a new point is proportional to the number of SVs, hence reducing the number of SVs implies reducing the testing time.

B. Online Independent Support Vector Machines

Let the *kernel matrix* K be defined such that $K_{ij} = K(\mathbf{x}_i, \mathbf{x}_j)$, with $i, j = 1, \ldots, l$. The possibility to obtain a more compact representation of $f(\mathbf{x})$ follows from the fact that the solution to a SVM problem (that is, the α_is) is not unique if K does not have full rank [14], which is equivalent to some of the SVs being linearly dependent on some others in the feature space [16]. Orabona et al [2] applied this idea to the online learning framework. As it would be unfeasible a simplification of the solution each time a new sample is acquired, they suggested to use independent SVs only, that is to decouple the concept of "basis" vectors, used to build the classification function (1), from the samples used to evaluate the ξ_i in (2). If the selected basis vectors span the same subspace as the *whole sample set*, the solution found will be equivalent.

The OI-SVM algorithm adds incrementally a new incoming sample if it is linearly independent in the feature space from those already present in the basis itself. The solution found is *the same* as in the classical SVM formulation; therefore, no approximation whatsoever is involved.

Denoting the indexes of the vectors in the current basis, after l training samples, by $\mathcal{B}$, and the new sample under judgment by $\mathbf{x}_{l+1}$, the algorithm can then be summed up as follows:

1) check whether $\mathbf{x}_{l+1}$ is linearly independent from the basis in the feature space; if it is, add it to $\mathcal{B}$; otherwise, leave $\mathcal{B}$ unchanged.
2) incrementally re-train the machine.

Hence the testing time for a new point will be $O(|\mathcal{B}|)$, as opposed to $O(l)$ in the standard approach; therefore, keeping $\mathcal{B}$ small will improve the testing time without losing any precision whatsoever. A major drawback of OI-SVM is that it requires to store in memory all the incoming training data in order to guarantee that the online solution is the same as in the classical SVM formulation.

In the following, the notations A_{IJ} and $\mathbf{v}_I$, where A is a matrix, $\mathbf{v}$ is a vector and $I, J \subset \mathbb{N}$ denote in turn the sub-matrix and the sub-vector obtained from A and $\mathbf{v}$ by taking the indexes in I and J.

Linear independence In general, checking whether a matrix has full rank is done via some decomposition, or by looking at the eigenvalues of the matrix; but here one wants to check whether a *single* vector is linearly independent from a matrix which is already known to be full-rank. Inspired by the definition of linear independence, the algorithm checks how well the vector can be approximated by a linear combination of the vectors in the set [17]. Let $d_j \in \mathbb{R}$; then let

$$\Delta = \min_{\mathbf{d}} \left\| \sum_{j\in\mathcal{B}} d_j \phi(\mathbf{x}_j) - \phi(\mathbf{x}_{l+1}) \right\|^2 \quad (3)$$

If $\Delta > 0$ then $\mathbf{x}_{l+1}$ is linearly independent with respect to the basis, and $\{l+1\}$ is added to $\mathcal{B}$. In practice, it checks whether $\Delta \le \eta$ where $\eta > 0$ is a tolerance factor, and expects that larger values of η lead to worse accuracy, but also to smaller bases. As a matter of fact, if η is set at machine precision, OISVMs retain the exact accuracy of SVMs. Notice also that if the feature space has finite dimension n, then no more than n linearly independent vectors can be found, and $\mathcal{B}$ will never contain more than n vectors.

Expanding equation (3) one gets

$$\Delta = \min_{\mathbf{d}}\Big(\sum_{i,j\in\mathcal{B}} d_j d_i \phi(\mathbf{x}_j)\cdot\phi(\mathbf{x}_i) \\ -2\sum_{j\in\mathcal{B}} d_j \phi(\mathbf{x}_j)\cdot\phi(\mathbf{x}_{l+1}) + \phi(\mathbf{x}_{l+1})\cdot\phi(\mathbf{x}_{l+1})\Big) \quad (4)$$

that is, applying the kernel trick,

$$\Delta = \min_{\mathbf{d}} \left(\mathbf{d}^T K_{\mathcal{B}\mathcal{B}} \mathbf{d} - 2\mathbf{d}^T \mathbf{k} + K(\mathbf{x}_{l+1}, \mathbf{x}_{l+1}) \right) \quad (5)$$

where $k_i = K(\mathbf{x}_i, \mathbf{x}_{l+1})$ with $i \in \mathcal{B}$. Solving (5), that is, applying the extremum conditions with respect to $\mathbf{d}$, one obtains

$$\tilde{\mathbf{d}} = K_{\mathcal{B}\mathcal{B}}^{-1}\mathbf{k} \quad (6)$$

and, by replacing (6) in (5) once,

$$\Delta = K(\mathbf{x}_{l+1}, \mathbf{x}_{l+1}) - \mathbf{k}^T \tilde{\mathbf{d}} \quad (7)$$

Note that $K_{\mathcal{B}\mathcal{B}}$ can be safely inverted since, by incremental construction, it is full-rank. An efficient way to do it, exploiting the incremental nature of the approach, is that of updating it recursively: after the addition of a new sample, the new $K_{\mathcal{B}\mathcal{B}}^{-1}$ then becomes

$$\begin{bmatrix} & & & 0 \\ & K_{\mathcal{B}\mathcal{B}}^{-1} & & \vdots \\ & & & 0 \\ 0 & \cdots & 0 & 0 \end{bmatrix} + \frac{1}{\Delta}\begin{bmatrix} \tilde{\mathbf{d}} \\ -1 \end{bmatrix}\begin{bmatrix} \tilde{\mathbf{d}}^T & -1 \end{bmatrix} \quad (8)$$

where $\tilde{\mathbf{d}}$ and Δ are already evaluated during the test (this method matches the one used in Cauwenberghs and Poggio's incremental algorithm [18]). Thanks to this incremental evaluation, the time complexity of the linear independence check is $O(|\mathcal{B}|^2)$, as one can easily see from Equation (6).

With this method OI-SVM approximates the original kernel matrix K with another matrix $\widehat{K}$ [19]; the quality of the approximation depends on η. In fact it is possible to show that $trace(K - \widehat{K}) \le \eta|\mathcal{B}| \le \eta l$, where l is the number of samples acquired [20]. If one considers a normalized kernel, that is a kernel for which $K(x,x)$ is always equal to 1, we can write $trace(K - \widehat{K})/trace(K) \le \eta$. On the other hand a bigger η means of course a smaller number of SVs, hence it controls the trade-off between accuracy and speed of the OISVM.

Training the machine The training method largely follows Keerthi et al. [21], [22], adapted for online training. The algorithm directly minimizes problem (2) as opposed to the standard way of minimizing its dual Lagrangian form, allowing to select explicitly the basis vectors to use. OI-SVM sets $p = 2$ in (2) and transform it to an unconstrained problem. Let $\mathcal{D} \subset \{1, \ldots, l\}$; then the unconstrained problem is

$$\min_{\boldsymbol{\beta}} \left(\frac{1}{2}\boldsymbol{\beta}^T K_{\mathcal{D}\mathcal{D}}\boldsymbol{\beta} + \frac{1}{2}C\sum_{i=1}^{l} max\left(0, 1 - y_i K_{i\mathcal{D}}\boldsymbol{\beta}\right)^2 \right) \quad (9)$$

where $\boldsymbol{\beta}$ is the vector of the Lagrangian coefficients involved in $f(\mathbf{x})$, analogously to the α_is in the original formulation. If one sets $\mathcal{D} = \mathcal{B}$, then the solution to the problem is unique since $K_{\mathcal{B}\mathcal{B}}$ is full rank by construction. Newton's method as modified by Keerthi et al. [21], [22] can then be used to solve (9) after each new sample. When the new sample $\mathbf{x}_{l+1}$ is received the method goes as follows:

1) let $\mathcal{I} = \{i : 1 - y_i o_i > 0\}$ where $o_i = K_{i\mathcal{B}}\boldsymbol{\beta}$ and $\boldsymbol{\beta}$ is the vector of optimal coefficients with l training samples; if $\mathcal{I}$ has not changed, stop.
2) otherwise, let the new $\boldsymbol{\beta}$ be $\boldsymbol{\beta} - \gamma\mathbf{P}^{-1}\mathbf{g}$, where $\mathbf{P} = K_{\mathcal{B}\mathcal{B}} + CK_{\mathcal{B}\mathcal{I}}K_{\mathcal{B}\mathcal{I}}^T$ and $\mathbf{g} = K_{\mathcal{B}\mathcal{B}}\boldsymbol{\beta} - CK_{\mathcal{B}\mathcal{I}}\left(\mathbf{y}_{\mathcal{I}} - \mathbf{o}_{\mathcal{I}}\right)$.
3) go back to Step 1.

In Step 2 above, γ is set to one. In order to speed up the algorithm, OI-SVM maintains an updated Cholesky decomposition of $\mathbf{P}$. It turns out that the algorithm converges in very few iterations, usually 0 to 2; the time complexity of the re-training step is $O(|\mathcal{B}|l)$, as well as its space complexity; hence, keeping $\mathcal{B}$ small will speed up the training time as well as the testing time.

C. Memory Controlled Online Independent Support Vector Machines

The need to store all incoming data makes in practice unusable the OI-SVM algorithm for open-ended learning of semantic spatial concepts, especially for a mobile robot platform: while the dimension of the solution would remain constant over time, the overall memory requirement would grow linearly with the number of perceived frames, leading quickly to a memory explosion.

To overcome this problem, we propose to apply a forgetting strategy over the stored Training Samples (TSs), while preserving the stored Support Vectors in order to approximate reasonably well the original optimal solution. The idea of keeping under control the memory growth of online learning algorithms is not new: several authors tried in the past to address this problem, mainly by bounding a priori the memory requirements. The first algorithm to overcome the unlimited growth of the support set was proposed by Crammer et al. [23]. The algorithm was then refined by Weston et al. [24]. The idea of the algorithm was to discard a vector of the solution, once the maximum dimension has been reached. The strategy was purely heuristic and no mistake bounds were given. A similar strategy has been used also in NORMA [25] and SILK [26]. The very first online algorithm to have a fixed memory "budget" and at the same time to have a relative mistake bound has been the Forgetron [27]. Within the context of semantic scene recognition, Ullah et al [3] proposed instead a random forgetting strategies, which should be more robust to possible unbalancing into the class-by class distribution of the TSs.

Here we take the approach proposed in [3] and define the following random forgetting strategy:

1) we introduce a threshold value that corresponds to the allowed maximum number of stored Training Samples ($MaxTSs$);
2) whenever $TS > MaxTSs$, we randomly discard TSs until their value is again below threshold. This concretely means discarding old TSs, selected randomly, for each new incoming TS.
3) With this strategy, the memory requirements of the algorithm are always between the number of SVs of the testing solution and the number of SVs plus $MaxTSs$.

We will show experimentally that this approximation of the original OI-SVM algorithm does not affect the accuracy of the solution for a wide range of values of $MaxTSs$.

IV. Step 2: Detection of Ignorance

The second, key component of our method is the capability to autonomously assign labels to new, incoming images, without the need for human supervision. The core issue here is the ability to estimate the level of confidence of each potential decision: a frame classified as corridor should be used to update the internal representation for the class corridor only if the confidence of the decision is high enough. If this would not be the case, then there would be a very strong risk of adding wrongly labeled data to the model, with a consequent degradation of the overall performance over time.

At the same time, one could argue that the most challenging frames, for each known class, are the most important to be added as they are those bringing new valuable information. An obvious way to do so would be to store the challenging frames and then, periodically, asking for labels to a human supervisor. Our solution here is instead to exploit the temporal continuity between frames and the intrinsic constraints of the problem: once a robot has traversed a door, all frames perceived until crossing another door must belong to the same semantic spatial concept. This same line of reasoning gives us a useful tool to determine if the robot has entered a new, unknown room.

Section IV-A describes how we estimate the level of confidence of the classifier and how to exploit temporal continuity to label challenging frames. Section IV-B illustrates how these two ingredients can be also used to identify new semantic spatial concepts. Finally, Section IV-C shows how challeging frames can be used to improve future classifications without having identified a new room.

A. Detecting Challenging Frames

To use incoming data to update the internal models, the algorithm needs to assign reliably class labels to each new frame. This in turns means that it should be able to detect frames that cannot be properly classified, i.e. that cannot be classified with a high confidence level. The problem therefore becomes that of defining effective confidence measures for evaluating the reliability of the label assignement process.

This issue has been widely studied in the literature [28]. Here we follow the classic approach proposed by Platt [29], which turns decision margins into conditional probabilities:

$$Pr(y = 1|x) \approx P_{A,B}(f) \equiv \frac{1}{1 + exp(Af + B)}. \qquad (10)$$

A study comparing different methods for estimating confidences from the output of SVMs indicated this approach as the most stable [28]. The value of f in equation(10) is the decision margin obtained for the input x at time t. The A and B values are parameters that should be estimated using a set of well annotated decision margins. Here we use the Platt implementation, proposed in [30], to estimate the values of A and B.

We expect that the obtained probabilities will have values close to 1 when a frame should be classified using the selected class (hard acceptance) and close to 0 for a hard rejection. We therefore replace the output margins with the conditional probabilites obtained via equation (10). We denote them in the following as M. On the basis of the conditional probabilities ${M^i_n}_{i=1}^{C}$, with C= number of classes, for each frame n, we define the two following conditions for detecting challenging frames:

1) $M^i_n < {M_{max}}_{i=1}^{C}$: for each of the possible classes C none obtains a high level of confidence;

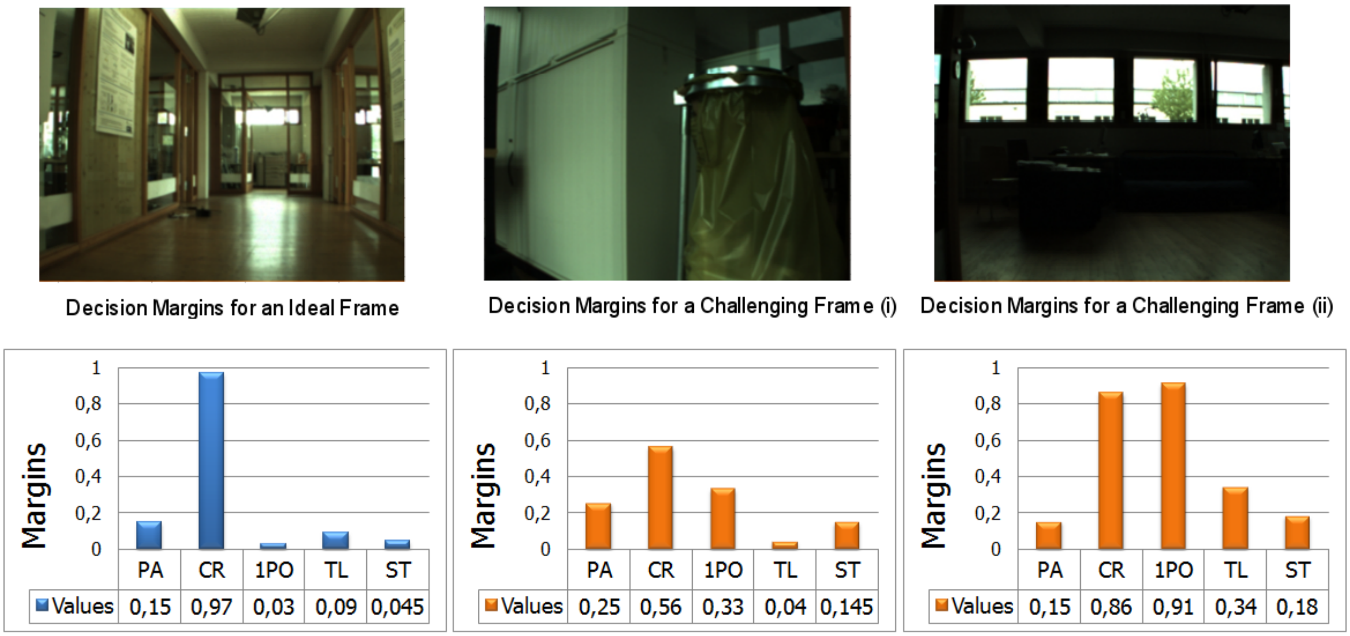

Fig. 1. Decision margins obtained for an ideal frame and two types of challenging frames.

2) $|M_n^i - M_n^j| < \Delta_{i=1}^{C}$: there are at least two classes with high level of confidence, but their difference is too small to allow for a confident decision.

In order to show why these two conditions are important, Figure 1 shows three examples of the conditional probabilities obtained for a frame correctly classified with high confidence (left) and two frames labelled as challenging frames (centre and right). Figure 1, centre, show an example of frame classified as challenging because of a low level of confidence (condition (1)); Figure 1, right, shows instead an example of challenging frame where there are two high and very close levels of confidence (condition (2)).

Once a frame has been identified as challenging, it is added to a set of challenging frames that will be used to retrain the classifier or discarded when the robot cross a new door. This decision will depend on the conditional probabilities for all classes since the robot crossed the last perceived door, as it will be discussed in the next section.

The value of M_{max} and Δ was selected after several preliminary results, taking into account the potential risk of retraining the classifier with a frame incorrectly classified. Using a conservative approach, a test frame p is not labelled as challenging only if the confidence value M_p^j was higher than 0.95 and also higher than $0.8 * \sum_{i=1}^{C} M_n^i$.

B. Detecting New Rooms

Once a frame has been classified as challenging, we might be facing two very different situations: (a) the robot has entered a room never seen before, or (b) the robot has entered a room previously seen under some unusual imaging conditions. In this Section we discuss how to detect the first case.

When a robot enters a room not seen during training, we would expect that most of the conditional probabilities for all known classes should be close to 0. Furthermore, we would expect that by looking at all frames since crossing the last detected door one would not be able to detect a dominant class label. Our proposal is to use these two behaviors as indicators of being in a new room, to do so, we have developed the following quantitative approach: starting from the moment when the robot detects crossing a door, we consider n frames and their associated conditional probabilities. If the majority of the frames are classified as challenging, then the robot has entered a new room. Quantitatively this can be measured as follows: We define the quantity $S_{i}{}_{i=1}^{C} = \sum M_{n}^{i}{}_{i=1}^{C}$, with C number of classes.

To detect a new room, at least one of the two following conditions needs to be met:

- $S_i < n * T1$: for each of the possible classes C none obtains a high level of confidence;
- $\sum_{i=1}^{C} S_i < n*T2$: the sum of all conditional probabilities over all frames is below n.

$T1$ and $T2$ are threshold values determined experimentally; in this paper the selected values where $T1 = 0.3$ and $T2 = 0.4$. If at least one of the two conditions is satisfied, we assume that the robot has visited a new room. In this situation, the robot will use all challenging frames (that should be similar to n) to retrain the classifier using a new label. That label can be directly generated by the robot, or the robot can ask for a new label to a human supervisor.

C. Detecting Old Rooms

After detecting a door crossing situation and having computed S_i for all classes, we can have two cases:

- The robot has entered a new room
- The robot has entered a known room.

The first case has been discussed in the last section. Here we focus on the second. Detecting challenging frames when entering a known room is most likely related to some substantial changes in the imaging conditions, such as varying illuminations of furniture relocated. These frames, if correctly labeled, would be very valuable for the algorithm because they might contribute to avoid misclassifications in the future. Again, we propose to detect known rooms by studying the behavior of S_i. We say that the robot has entered the room class C_j if the two following conditions are satisfied:

- $S_j > T3 * n$
- $S_j > T4 * \sum_{i=1}^{C} S_i$

This situation is represented in Figure 1 left. The $T3$ and $T4$ values were experimentally selected to $T3 = 0.5$ and $T4 = 0.65$. After detecting a known room, all challenging frames are used to retraining the classifier, using C_j as the correct class. If the system is not able to assign the challenging frames to a new room, or to a known room, the frames are discarded. The whole process is illustrated in Figure 2

V. Door detection algorithm

Our system is based on the ability of the robot to detect door crossing. Current indoor robots are usually equipped with a large number of sensors, mainly visual cameras and distance sensors. Door detection has been deeply studied in literature and we can implement for instance the algorithm presented in [31].

Not all databases available as benchmarks for robot localization provide laser data, so we decided to use only the visual information for doors detection. This section illustrates a simple door detector developed for the Robot Vision challenge inside the ImageCLEF [1] campaign.

Door crossing within CLEF training sequences can be observed as two vertical rectangles with the same colour that increase their side and suddently the dissappear. We will detect that situation by extracting these rectangles and studing their width evolution when new frames are acquired. The image processing starts with a Canny filter [32] to extract all the edges of the images. After this preliminary step, we apply the Hough transform [33] for lines detection discarding all the non vertical lines. The last step to extract the rectangles is to measure the average colour value between each two vertical lines, removing non homogeneous colour distributions (blobs). Fig. 3 shows this process, where three colour homogeneous blobs are detected and two could be used to detect the door crossing.

After extracting all the key blobs from a frame, we have to study the time correspondence for these blobs between this frame and the last frames. If two blobs with the same average colour are increasing for new frames we are reaching a door and both blobs are marked as candidates. Preliminary candidates will be selected as definitive ones if one of the two blobs starts decreasing after reaching the largest size at the left (right) of the image. Figure 4 shows four consecutive training frames, where white rectangles represent blobs, preliminary candidates are labelled with a P and definitive candidates with a D. Green rectangles for the bottom images represent the time correspondence for each blob in the last frames. This idea can be extended to develope simple and efficient door crossing detectors for corridor environments, and was used to detect the door crossing in this paper.

[1]http://www.robotvision.info/

VI. Experimental Setup

In this section we describe the experimental setup used to validate our approach. Section VI-A describes the data used, and section VI-B the feature descriptors. The description of each experiment with the corresponding result is given in section VII.

A. The Database

For our experiments we used two different databases that we describe in the rest of the section.

The COLD Database The COLD database [5] contains three separate sub-datasets, acquired at three different indoor labs, located in three different European cities: the Visual Cognitive Systems Laboratory at the University of Ljubljana, Slovenia; the Autonomous Intelligent System Laboratory at the University of Freiburg, Germany; and the Language Technology Laboratory at the German Research Center for Artificial Intelligence in Saarbrücken, Germany. For each lab, image sequences of several rooms are provided, all acquired with the same camera settings.

Here we used the sub-dataset acquired in the Autonomous Intelligent System Laboratory at the University of Freiburg, Germany (COLD-Freiburg): it consists of three sets of sequences, both acquired under varying illumination conditions. Of these three sets, we chose the following two: In the first set, the robot travels across five rooms: corridor, 2-person office, printer area, bathroom and stairs area. In the second set, the robot travels across the rooms of the first set, plus other four rooms: a 1-person office, a printer area, a kitchen and a large office. Figure 5 shows some exemplar views from the second set of sequences. Each of the sequences described above were acquired under three different illumination conditions -sunny, cloudy and night. Three sequences were acquired, one after the other, for each weather condition, for a total of nine data sequences for each set.

The ImageCLEF 2010 Database The image sequences used for the Robot Vision Task at the ImageCLEF 2010 challenge evaluation were taken from the previously unreleased COLD-Stockholm database [34]. The sequences were acquired using a MobileRobots PowerBot robot platform equipped with a stereo camera system consisting of two Prosilica GC1380C cameras. The acquisition was performed on three different floors of an office environment, consisting of 36 areas (usually corresponding to separate rooms) belonging to 12 different semantic and functional categories. 8 of these semantic categories are shown in Fig.5

The robot was manually driven through the environment while continuously acquiring images at a rate of 5fps. Each data sample was then labeled as belonging to one of the areas

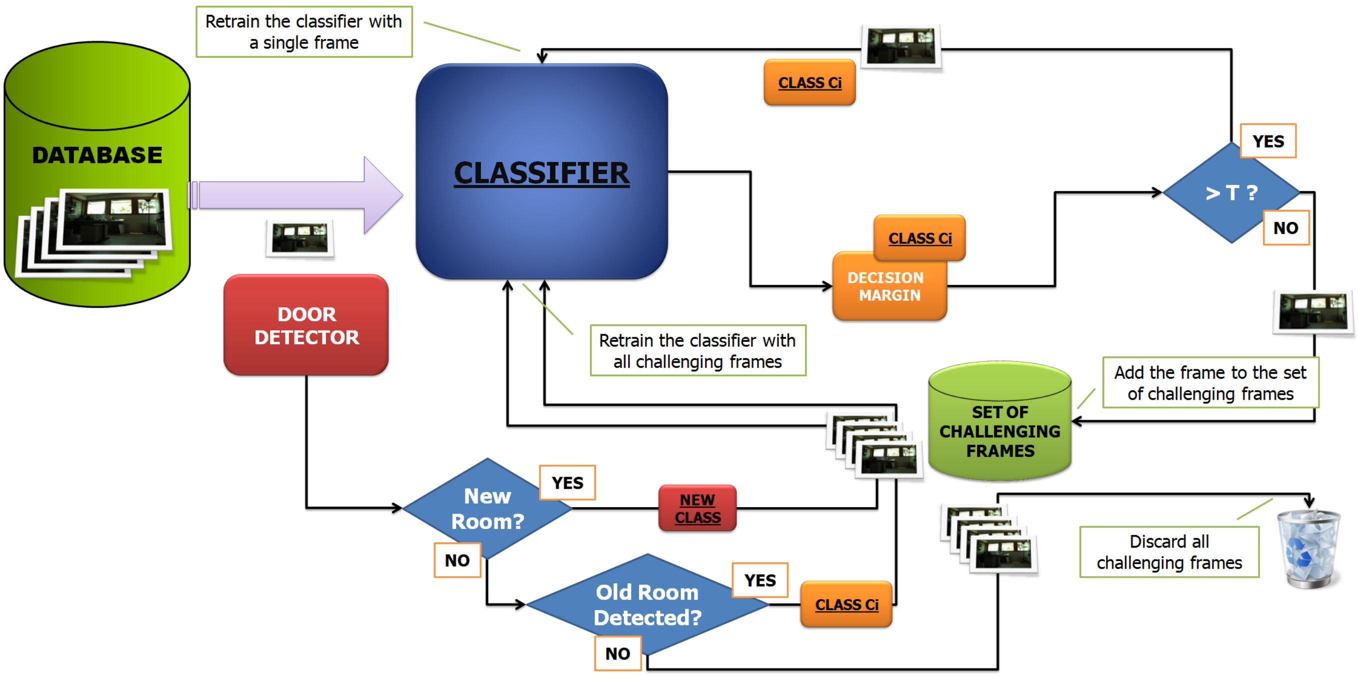

Fig. 2. Complete classification process performed by our proposal, where frames classified that obtained promising decision values (not challenging frames) are used to retrain the classifier and the set of challenging frames is processed after crossing a door.

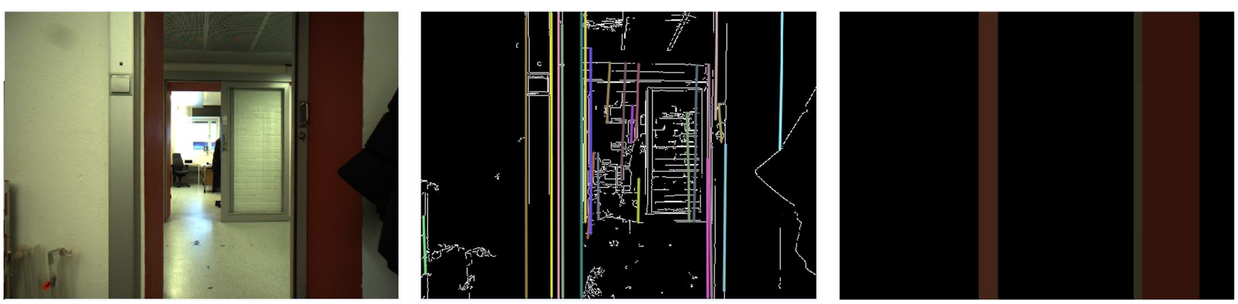

Fig. 3. Complete process to extract blobs. Left: original image. Centre: Vertical lines detection. Right: Homogeneous colour distributions between two vertical lines (BLOB extraction)

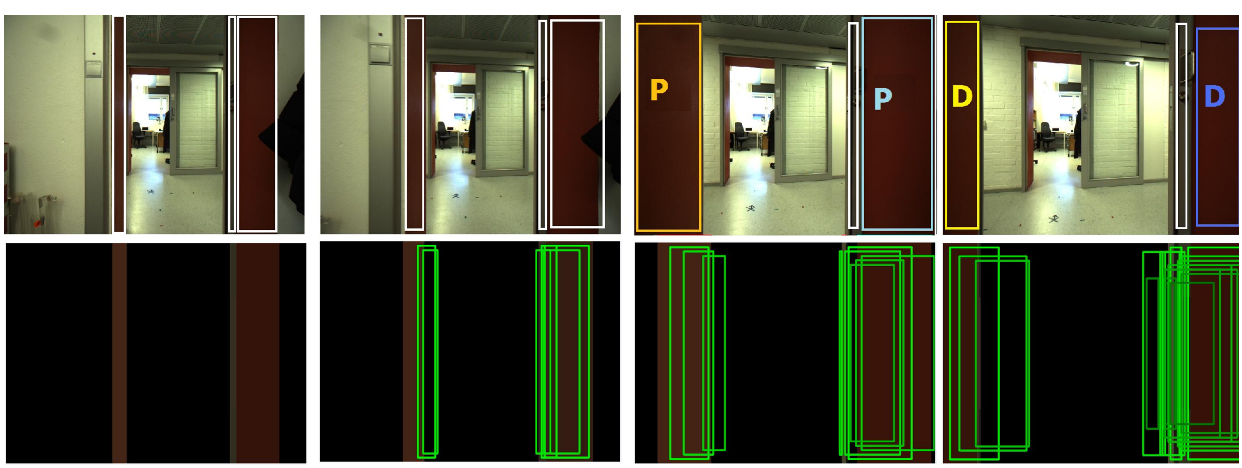

Fig. 4. Door detection for four consecutive frames. Top images are the original frames using P for preliminary candidates and D for definitive ones. Bottom images show the blobs extracted and time correspondence between them

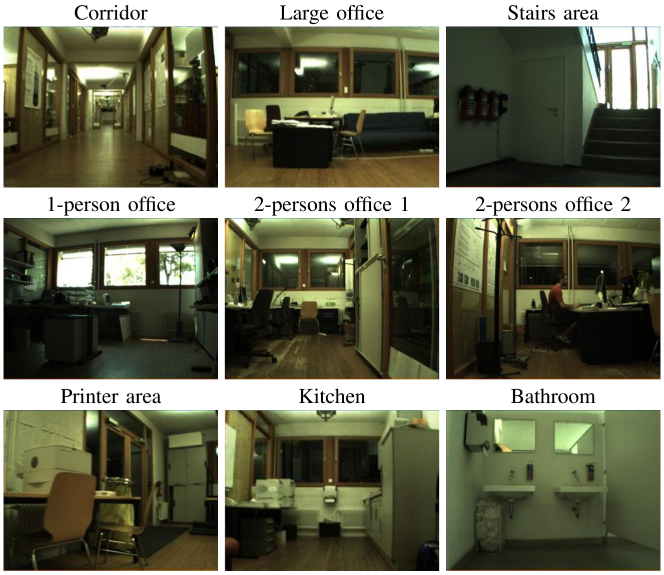

Fig. 5. Examples of images from the COLD-Freiburg database.

Fig. 6. Examples of images from the CLEF 2010 database.

according to the position of the robot during acquisition (rather than contents of the images).

Three sequences were selected for the contest: a training sequence, a sequence that should be used for validation and a sequence for testing:

- training sequence: Sequence acquired in 11 areas, on the 6th floor of the office building, during the day, under cloudy weather. The robot was driven through the environment following a similar path as for the test and validation sequences and the environment was observed from many different viewpoints (the robot was positioned at multiple points and performed 360 degree turns).
- validation sequence: Sequence acquired in 11 areas, on the 5th floor of the office building, during the day, under cloudy weather. Similar path was followed as for the training sequence; however without making the 360 degree turns.
- testing sequence - Sequence acquired in 14 areas, on the 7th floor of the office building, during the day, under cloudy weather. The robot followed similar path as in case of the validation sequence.

B. The Features

As features, we chose a variety of global descriptors representing different features of the images. We opted for histogram-based global features, mostly in the spatial-pyramid scheme introduced in [35].

This representation scheme was chosen because it combines the structural and statistical approaches: it takes into account the spatial distribution of features over an image, while the local distribution is in turn estimated by mean of histograms; moreover it has proven to be more versatile and to achieve higher accuracies in our experiments.

The descriptors we have opted to extract belong to five different families: Pyramid Histogram of Orientated Gradients (PHOG) [36], Sift-based Pyramid Histogram Of visual Words (PHOW) [37], Pyramid histogram of Local Binary Patterns (PLBP) [38], Self-Similarity-based PHOW (SS-PHOW) [39], and Compose Receptive Field Histogram (CRFH) [40]. Among all these descriptors, CRFH is the only one which is not computed pyramidly. For the remaining families we have extracted an image descriptor for every value of $L = \{0, 1, 2, 3\}$, so that the total number of descriptors extracted per image is equal to 25 ($4+4$ PHOG, $4+4$ PHOW, 4 PLBP, 4 SS-PHOW, 1 CRFH). In order to select the best visual cues to be combined together we performed a pre-selection step, namely we run some preliminary experiments to decide which combination of features was more effective. This eventually made us settle on two descriptors, PHOG L0 and Oriented PHOG L2. Their exact settings are summarized in Table I for the experiments done on the COLD database, and in Table II for the experiments done on the ImageCLEF database. These features are concatenated to generate a single feature that will be used as input for the classifier.

Descriptor	Settings	L
PHOG_{180}	range= [0, 180] and $K = 20$	$\{0\}$
PHOG_{360}	range= [0, 360] and $K = 40$	$\{2\}$

TABLE I
SETTINGS OF THE IMAGE DESCRIPTORS USED FOR THE COLD DATABASE

Descriptor	Settings
PHOG_{180}	range=[0, 180], K=20, L=$\{0, 1, 2, 3\}$
PHOG_{360}	range=[0, 360], K=40, L=$\{0, 1, 2, 3\}$
$\text{PHOW}_{\text{gray}}$	$[M, V]$=[10, 300], $r = \{4, 8, 12, 16\}$, L=$\{0, 1, 2, 3\}$
$\text{PHOW}_{\text{color}}$	$[M, V]$=[10, 300], $r = \{4, 8, 12, 16\}$, L=$\{0, 1, 2, 3\}$
$\text{PLBP}^{riu2}_{8,1}$	$[P, R]$=[8,1], RotationInvariantUniform2 version, L=$\{0, 1, 2, 3\}$
SS-PHOW	$[M, V, S, R, nRad, nTheta]$=[5,300,5,40,4,20], L=$\{0, 1, 2, 3\}$
CRFH	Gaussian-Derivatives=$\{L_x, L_y\}$, K=14, s=$\{1, 2, 4, 8\}$

TABLE II
SETTINGS OF THE IMAGE DESCRIPTORS USED FOR THE IMAGECLEF DATABASE.

VII. RESULTS

The main objective of all experiments performed is to test the feasiblity of using our system for real-case scenarios. to this end, we need to demonstrate experimentally two points: (1) that the performance of the Memory controlled OI-SVM is similar to that of the original method, and (2) that when retraining the system with new self-labeled frames, the performance of the algorithm increases.

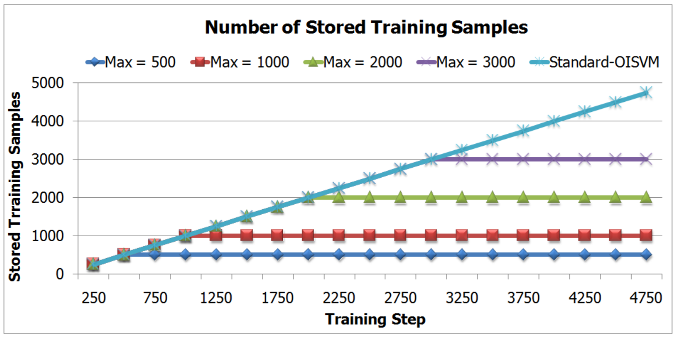

Fig. 8. Number of stored training samples for different $MaxTSs$ values

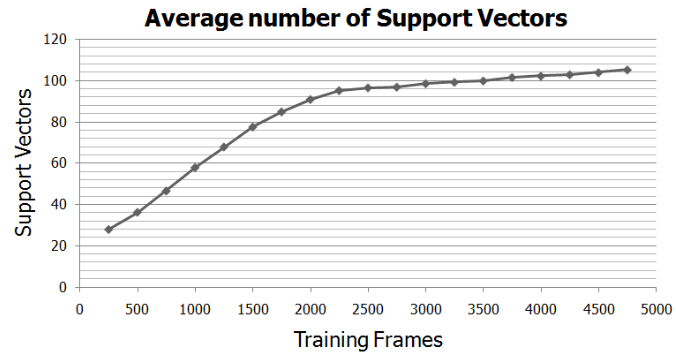

Fig. 10. Average number of support vectors stored for Experiment 1

All the experiments were performed with an SVM classifier using the χ^2 kernel, with $C = 1$, $\gamma = 1$ and $\eta = 0.25$. The ImageCLEF and the COLD-Freiburg were the databases used for the experiments. The sequences for training and testing were selected as follows:

- ImageCLEF: We just had a single combination for training/testing: the proposed training sequence was used for training and the obtained classifier was tested with the validation sequence proposed for the task and correctly labelled (the ImageCLEF test sequence is not labelled).
- COLD-Freiburg: Training always consisted of three sequences (from those proposed for training in database, left column in Figure 7), acquired one after the other, with the same illumination conditions. Testing consisted of one sequence (from those proposed for testing, right column in Figure 7), taken from those not used for training. The COLD-Freiburg database consists of sequences of images acquired with three lighting conditions (cloudy, night and sunny), and so we used 9 training/testing combinations.

A. Experiment 1: Memory-Controlled OISVM

In a first set of experiments we compared the performance of the Memory Controlled OI-SVM and the original method. We determined a priori several values for the maximum number of stored training samples $MaxTSs$ and we measured the classification rate obtained for the selected training sequence and testing sequence combination.

To compare the performance of MC-OI-SVM with that of OI-SVM, we used the 9 sequence combinations from the COLD-Freiburg. Because of the different illumination conditions (cloudy, sunny, night) there are 9 different combinations of training and test data. We performed experiments on all of them, chosing for MC-OI-SVM four different values of $MaxTSs$: (500,1000,2000,3000).

In order to show how this threshold affects the memory requirements of the system, Fig 8 shows the number of training samples stored by the algorithm common for all different $MaxTSs$ values.

Experiments were performed as follows: the classifier was incrementally trained using the training sequence and evaluated periodically (250 frames). These evaluations were performed by classifing the whole testing sequence with the classifier obtained in that time. Figure 9 presents these results, where the 9 combinations are shown: columns represents the lighting conditions used for testing and rows for training. The $x - axis$ represents the number of training samples used to generate the classifier in that point and the $y - axis$ the error rate.

It can be observed how the error rate is not affected when using $MaxTSs$ values greater than 2000, because the performance of the two algorithms is essentially the same. From this Figure, and from Figure 8, we can make two remarks: (1) with MC-OI-SVM it is possible to obtain an impressive reduction on the memory requirements of the original method with a very negligible decrease in accuracy; (2) when the $MaxTSs$ value is too close to a certain limit L_{MaxTs} the approximation affects the optimality of the solution, and the error rate starts to increase.

The optimal value of this limit L_{MaxTs} depends of the number of support vectors stored by the optimal solution (without removing any training frame). In our experiments, we also stored the number of support vectors of the original OISVM, and the average number of stored support vectors for each number of training frames is shown in Figure 10.

From this Figure and Figure 9 we can observe that problems with $L_{MaxTs} = 500$ started when the number of training frames was higher 750 and became effective for most of the combinations when the number of training frames was 1000. For $L_{MaxTs} = 1000$, problems started (if they happened) when the number of training frames was 1500. If we translate this number of training frames into support vectors (using Fig.10), problems presented for $L_{MaxTs} = 500$ when having a number of support vectors higher than 46 and for $L_{MaxTs} = 1000$ when that number was higher than 77.

It should be studied the relationship between the number of stored support vectors and the minimum value of L_{MaxTs} without decrease in accuracy. This relationship could be used for a dynamic establishment of L_{MaxTs} using the number of stored support vectors.

B. Experiment 2: Retraining the classifier with the MC-OI-SVM

In a second set of experiments, we studied the impact of retraining the system using testing frames. For this experiment, our MC-OISVM classifier included all the processing shown in Fig. 2. After classifying a new frame, such test frame can be used to retrain the classifier or added to a set of

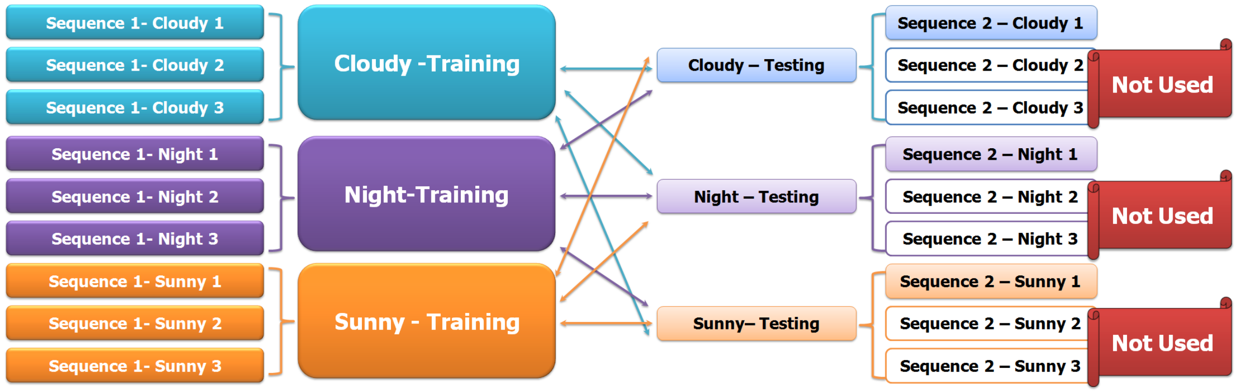

Fig. 7. 9 combinations of training and testing sequences selected from COLD-Freiburg database for the experiments 1 and 2

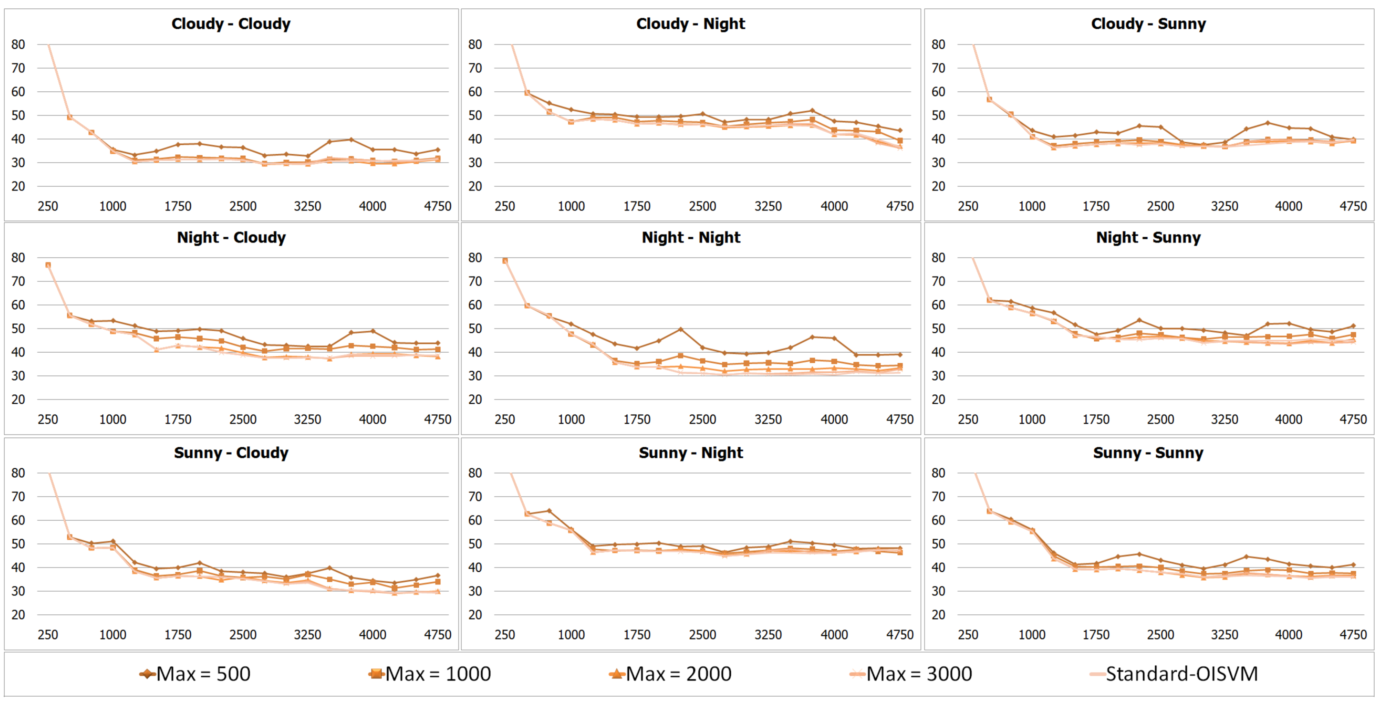

Fig. 9. Error rate (processing the whole test sequence) obtained incrementally for the 9 combinations of training and testing sequences from COLD-Freiburg database, obtained for different $MaxTSs$ values

challenging frames. The set of challenging frames can also be used to retrain the classifier, after detecting a door crossing. All settings used for the algorithm (and mentioned in Section IV-A, IV-B and IV-C) were common for all experiments. The value for L_{MaxTs} was 3000 for all experiments.

We used the same combinations and sequences for training and testing as for Experiment 1. For this experiment, instead of measuring the error rate over the complete test sequence while the classifier was being generated, we measured the relative accumulated error (RAE) while the test sequence was processed.

1) COLD-Freiburg Database: Fig.11 shows the results obtained for the 9 combinations of the COLD-Freiburg database, where each row was used for a different testing sequence and new rooms are marked using dark areas.

Adding MC-OISVM always obtained a smaller error rate than the original one, with an average improvement of 2.56% over all combinations. It can be observed how the relative error always increased for new rooms, regardless of the method used. This is because new rooms could only be detected after leaving them, so all their frames were incorrectly classified.

2) CLEF Database: The same experiment was performed using the CLEF database, with a single combination of training/testing. The obtained results can be observed in Fig. 12

The improvement obtained with the CLEF database (9.56%) was notoriously higher than those obtained with COLD-Freiburg. These results were promising for us, due to our proposal had a better behaviour facing a more challenging database (CLEF training/testing sequences were acquired on different floors).

Retraining our classifier with non-challenging frames improved the adaptability of our system to new lighting conditions. Expected lighting changes were exposed in an extreme way for our experiment, where the system was trained with

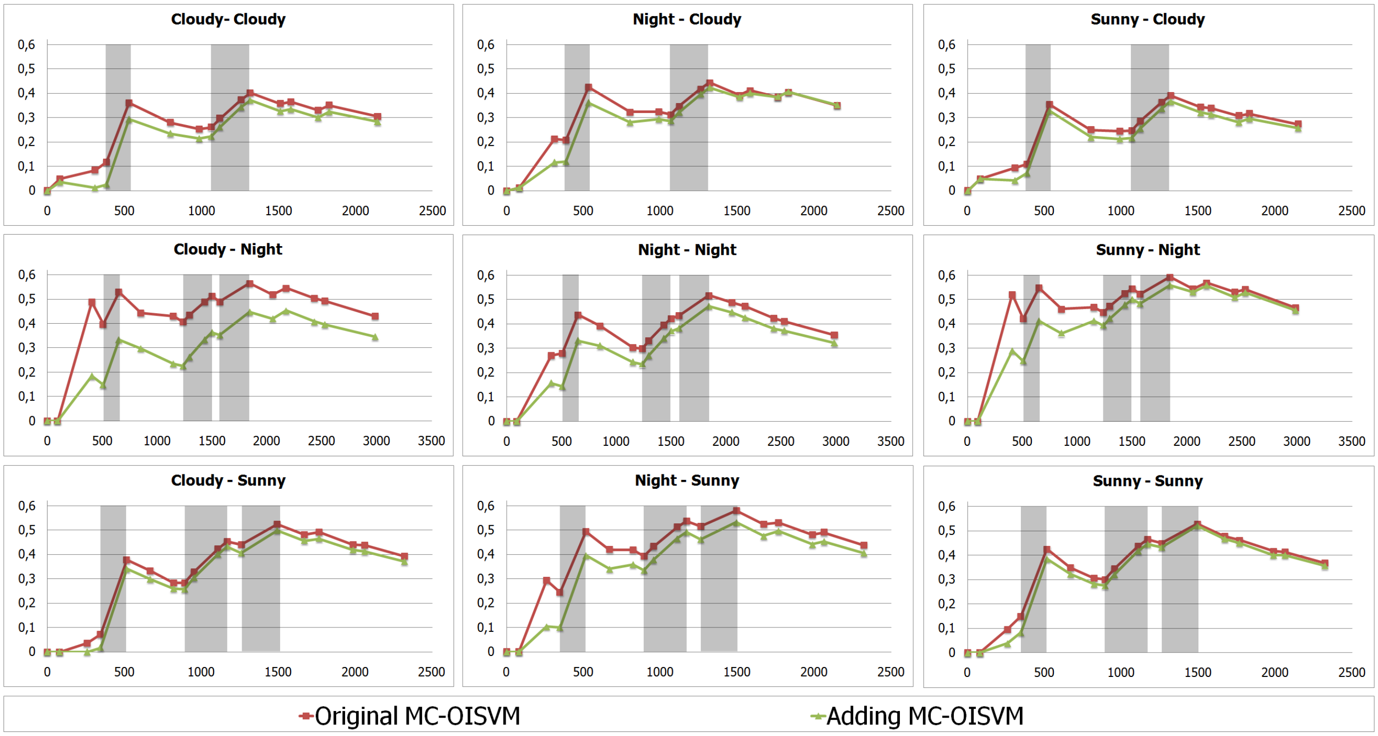

Fig. 11. Relative accumulated error obtained for the 9 combinations of training and testing sequences from COLD-Freiburg database, obtained with adding and original MC-OISVM. Dark areas represent new rooms not imaged previously

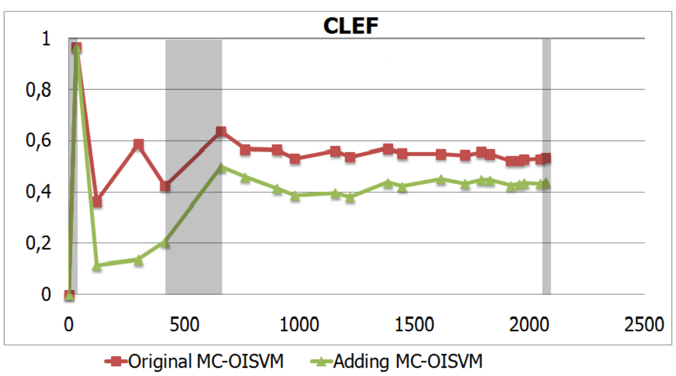

Fig. 12. Relative accumulated error obtained for CLEF database, obtained with adding and original MC-OISVM. Dark areas represent new rooms not imaged previously

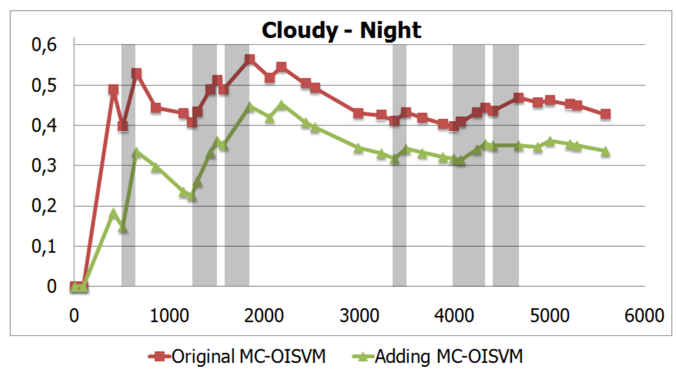

Fig. 13. Relative accumulated error for a Cloudy-Night combination

conditions totally different as those used for testing, without small or progressive variations.

C. Experiment 3: Impact of new room recognition for future recognition

The third set of experiments consisted of a deeper study on the capability of our system to recognize new rooms, and on its impact on performance over time. We performed the same experiments as in Experiment 2 but using only a new testing sequence generated by joining two of the proposed testing sequences obtained under the same lighting conditions (night). Because of that, new room detection it is supposed to improve considerably the error rate for future classifications.

After leaving a room, the set of challenging frames is studied and we determine the type of room we have visited: new room, known room or it is not possible to say the type of room without uncertainty. After a new room detection, the robot will ask for human supervision and a new label will be generated for the new spatial category. The complete set of challenging frames will be used to retrain the classifier using the new label as a class.

The experiment was performed with the same parameters as in Experiment 2, and the new testing sequence was processed after generating the classifier with the three training sequences used for the experiments 1 and 2: cloudy, night and sunny (exposed in Fig.7 left).

Figures 13, 14 and 15 present the results obtained for these combinations of training/testing sequences, where we measured the relative accumulated error.

The average improvement for the error rate using our approach over the original MC-OISVM was 8.10%. It can

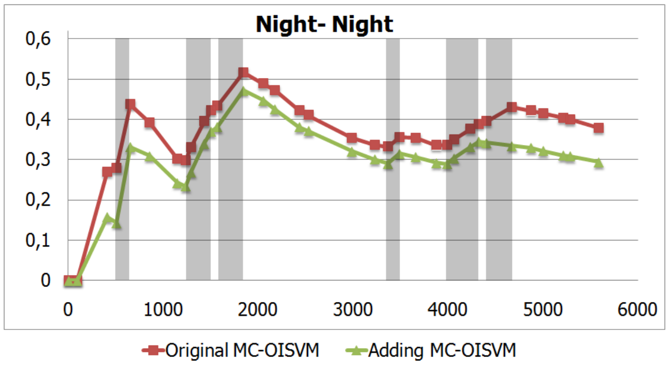

Fig. 14. Relative accumulated error for a Night-Night combination

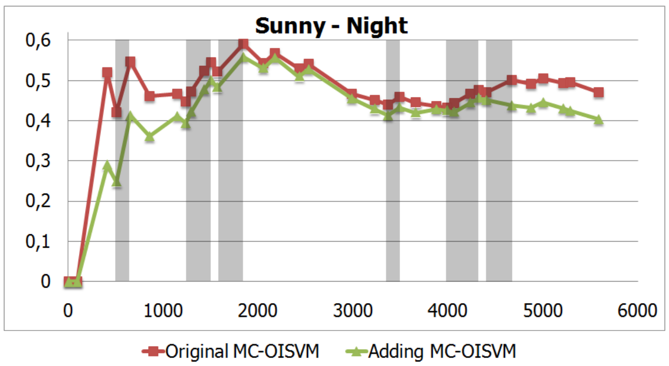

Fig. 15. Relative accumulated error for a Sunny-Night combination

be observed how the new room recognition increased the tolerance of the system to unknown rooms.

It should be pointed out the behaviour of the system with the third unknown room presented in the testing sequence: the kitchen. That room is represented (in Fig. 13, 14 and 15) by dark rectangles located between frames 1569-1841 and 4402-4672. First kitchen rectangle (third in figures) represents the first time the room appears, where all frames were misclassified for adding and original MC-OISVM. The second time this room appears (frames 4402-4672), all frames were incorrectly classified by the original MC-OISVM (the error rate increased notoriously) but perfectly labelled by the extra layer (the accumulated error rate decreased). The other two unknown rooms presented problems due to the similarity between these rooms and other previously trained: Large Office and 1 Persons Office were similar to 2 Persons Office. This similarity affected to the detection of the new room and also presented problems for future classifications.

VIII. Conclusions and Future Work

In this paper we presented an algorithm for online learning of semantic spatial concepts with a bounded memory growth, able to measure its own level of confidence when classifying incoming frames, and therefore able to decide when to ask for human annotation and when to trust its own decisions. Experiments on a subset of the challenging COLD database [5] show that our approach is able to minimize the false positives when classifying known frames, and it is able to detect new rooms, not seen during training.

This work can be continued in many ways. With respect to the confidence estimate, here we used the conditional probabilities of the SVM-based classifiers, but more elegant and sophisticated options should be explored here. Also, here we applied the method to only visual features, but this framework should work, and benefit from, multi-modal data such as laser range features. Future work will proceed in these directions.

References

[1] G. Lakoff, *Women, fire and dangerous things: what categories reveal about the mind*. The University of Chicago Press, 1990.

[2] F. Orabona, C. Castellini, B. Caputo, J. Luo, and G. Sandini, “Indoor place recognition using online independent support vector machines,” in *Proc. BMVC*, vol. 7. Citeseer.

[3] M. Ullah, F. Orabona, B. Caputo, I. IRISA, and F. Rennes, “You live, you learn, you forget: Continuous learning of visual places with a forgetting mechanism,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems, 2009. IROS 2009*, 2009, pp. 3154–3161.

[4] A. Pronobis and B. Caputo, “Confidence-based cue integration for visual place recognition,” *Proc. IROS07*.

[5] ——, “COLD: COsy Localization Database,” *The International Journal of Robotics Research (IJRR)*, vol. 28, no. 5, May 2009.

[6] C. Siagian and L. Itti, “Biologically-inspired robotics vision monte-carlo localization in the outdoor environment,” in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS’07)*, San Diego, CA, USA, October 2007.

[7] A. Pronobis, O. Martínez Mozos, and B. Caputo, “SVM-based discriminative accumulation scheme for place recognition,” in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA’08)*, Pasadena, CA, USA, May 2008.

[8] A. C. Murillo, J. Kosecka, J. J. Guerrero, and C. Sagues, “Visual door detection integrating appearance and shape cues,” *Robotics and Autonomous Systems*, vol. 56(6), pp. pp. 512–521, June 2008.

[9] C. Valgren and A. J. Lilienthal, “SIFT, SURF and seasons: Long-term outdoor localization using local features,” in *Proceedings of the European Conference on Mobile Robots (ECMR’07)*, 2007.

[10] S. Thrun and T. Mitchell, “Lifelong robot learning,” *Robotics and Autonomous Systems 15*, 1995.

[11] T. Mitchell, “The discipline of machine learning,” CMU, Tech. Rep. CMU-ML-06-108, 2006.

[12] J. Malak, R.J. and P. K. Khosla, “A framework for the adaptive transfer of robot skill knowledge using reinforcement learning agents,” in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA’01)*, 2001.

[13] G. Konidaris and A. G. Barto, “Autonomous shaping: knowledge transfer in reinforcement learning.” in *Proceedings of the 23rd International Conference on Machine Learning*, 2006.

[14] C. J. C. Burges, “A tutorial on support vector machines for pattern recognition,” *Knowledge Discovery and Data Mining*, vol. 2, no. 2, 1998.

[15] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines (and Other Kernel-Based Learning Methods)*. CUP, 2000.

[16] T. Downs, K. E. Gates, and A. Masters, “Exact simplification of support vectors solutions,” *Journal of Machine Learning Research*, vol. 2, pp. 293–297, 2001.

[17] Y. Engel, S. Mannor, and R. Meir, “Sparse online greedy support vector regression,” in *Proceedings ECML’02*, 2002.

[18] G. Cauwenberghs and T. Poggio, “Incremental and decremental support vector machine learning,” in *Proceedings of NIPS’00*, 2000, pp. 409–415.

[19] F. R. Bach and M. I. Jordan, “Predictive low-rank decomposition for kernel methods,” in *Proceedings of ICML’05*, 2005.

[20] Y. Engel, S. Mannor, and R. Meir, “The kernel recursive least squares algorithm,” *IEEE Transactions on Signal Processing*, vol. 52, no. 8, 2004.

[21] S. S. Keerthi and D. DeCoste, “A modified finite newton method for fast solution of large scale linear SVMs,” *Journal of Machine Learning Research*, vol. 6, pp. 341–361, 2005.

[22] S. S. Keerthi, O. Chapelle, and D. DeCoste, “Building support vector machines with reduced classifier complexity,” *Journal of Machine Learning Research*, vol. 8, pp. 1–22, 2006.

[23] K. Crammer, J. Kandola, and Y. Singer, “Online classification on a budget,” in *Advances in Neural Information Processing Systems 16*, 2003.

[24] J. Weston, A. Bordes, and L. Bottou, "Online (and offline) on an even tighter budget," in *Proceedings of AISTATS 2005*, R. G. Cowell and Z. Ghahramani, Eds., 2005, pp. 413–420.

[25] J. Kivinen, A. Smola, and R. Williamson, "Online learning with kernels," *IEEE Trans. on Signal Processing*, vol. 52, no. 8, pp. 2165–2176, 2004.

[26] L. Cheng, S. V. N. Vishwanathan, D. Schuurmans, S. Wang, and T. Caelli, "Implicit online learning with kernels," in *Advances in Neural Information Processing Systems 19*, 2007.

[27] O. Dekel, S. Shalev-Shwartz, and Y. Singer, "The Forgetron: A kernel-based perceptron on a budget," *SIAM Journal on Computing*, vol. 37, no. 5, pp. 1342–1372, 2007.

[28] S. Rüping, *A simple method for estimating conditional probabilities for svms*. Citeseer, 2004.

[29] J. C. Platt, "Probabilistic outputs for support vector machines," *Bartlett P. Schoelkopf B. Schurmans D. Smola, AJ, editor, Advances in Large Margin Classifiers*, pp. 61–74.

[30] H. Lin, C. Lin, and R. Weng, "A note on platt's probabilistic outputs for support vector machines," *Machine Learning*, vol. 68, no. 3, pp. 267–276, 2007.

[31] D. Anguelov, D. Koller, E. Parker, and S. Thrun, "Detecting and modeling doors with mobile robots," in *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, vol. 4. IEEE, 2004, pp. 3777–3784.

[32] J. Canny, "A computational approach to edge detection," *Readings in computer vision: issues, problems, principles, and paradigms*, vol. 184, 1987.

[33] D. Ballard, "Generalizing the Hough transform to detect arbitrary shapes," *Pattern recognition*, vol. 12, no. 2, pp. 111–122, 1981.

[34] A. Pronobis, M. Fornoni, H. I. Christensesn, and B. Caputo, "The robot vision track at imageclef 2010," in *Working Notes of ImageCLEF 2010*, 2010.

[35] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, 2006.

[36] A. Bosch, A. Zisserman, and X. Munoz, "Representing shape with a spatial pyramid kernel," in *Proceedings of the 6th ACM international conference on Image and video retrieval*. ACM, 2007, p. 408.

[37] ——, "Image classification using random forests and ferns," in *International Conference on Computer Vision*. Citeseer, 2007, pp. 1–8.

[38] T. Ojala, M. Pietikäinen, and T. Mäenpää, "Gray scale and rotation invariant texture classification with local binary patterns," *Computer Vision-ECCV 2000*, pp. 404–420, 2000.

[39] E. Shechtman and M. Irani, "Matching local self-similarities across images and videos," in *IEEE Conference on Computer Vision and Pattern Recognition, 2007. CVPR'07*, 2007, pp. 1–8.

[40] O. Linde and T. Lindeberg, "Object recognition using composed receptive field histograms of higher dimensionality," in *Proc. ICPR*. Citeseer, 2004.