Recognition of smart objects by a mobile robot using SIFT-based image recognition and wireless communication

M. Danieletto, M. Mina, A. Zanella, P. Zanuttigh, E. Menegatti
Department of Information Engineering
University of Padova
via Gradenigo 6/B
35131 Padova - ITALY

***Abstract*— In this work, we focus on the problem of object location and recognition by an autonomous mobile robot. In our approach, the robot does not have any prior knowledge about the form and multiplicity of the objects. The robot, however, is equipped with an onboard camera and both objects and robot are capable of exchanging data by using a common low-cost, low-rate wireless technology, namely a TmoteSky mote. The small storage memory of the mote is used to store a simple communication protocol and a description of the physical appearance of the object, encoded by means of a set of Scale Invariant Feature Transform (SIFT) descriptors. The mobile robot queries the surrounding smart objects by sending a broadcast query packet through the wireless interface. The smart objects that receive such a query reply by sending their ID and a selection of the SIFTS that describe their appearance. When a subset of the SIFT descriptors extracted by the current image of the robot's camera matches the SIFT descriptors received from a smart objects, the robot can locate the object in its current view and autonomously navigate towards the object, interacting with it.**

***Index Terms*— Object-Recognition, SIFT, WSN, mobile robot.**

Fig. 1. (Left) The autonomous mobile robot used in this work. (Right) Two of the smart objects used for test.

I. INTRODUCTION

Robots will be more and more common in our life in the near future. In particular, there is a huge expectation for a blow-out in the market of personal robots. Personal robots will need to interact with and manipulate objects in our houses or in office environment and, to this end, a personal robot needs to be able to recognize the different objects and locate each of them in the environment. Several approaches have been proposed for object detection and recognition, most of which perform visual recognition of the object by using a dataset of object-models stored on the robot [19, 20, 21]. This approach, however, requires the robot to know the object models beforehand, thus limiting the applicability of the solution. Additionally, with the current hardware and software technologies visual feature recognition is not stable, especially when two objects have very similar appearance. If both visual appearances of the two objects match the model stored in the robot, they become undistinguishable for the robot.

In this paper, we propose a system that overcomes these limits by tagging the objects with small wireless devices that provide some communication and processing capability. An object equipped with such a device, hence, acquires some intelligence, becoming a so-called *smart object*. In order to limit the economic and energetic costs of the wireless interfaces, we propose the use of the technology developed for Wireless Sensor Networks (WSN), i.e., the so-called *motes*. Smart objects are, hence, potentially capable of self-establishing a multi-hop communication network in order to relay messages to the robot in case direct communication is not available. Moreover, the mote applied to a smart object may be able to perform environmental measures, which may either involve the object itself (like object temperature, level of filling, inclination, wight, deterioration, etc), or the surrounding environment (temperature, pollution, humidity, and so on).

Unfortunately, it is well known that the most common WSN technologies do not support a precise geographical location of the nodes [1]. In our approach, instead of relying on the (very poor) self-location capability of the WSN, we exploit the much more advanced vision, motion and processing capacity of autonomous mobile robots. The basic idea is as follows. The mobile robot is equipped with both a mote, which is used to interact with the wireless network of smart objects, and an on-board camera. The robot inquiries the surrounding smart objects by using the wireless interface. The objects, which are in coverage range and get the robot inquiry, reply by transmitting a description of their appearance. The robot keeps comparing the images taken by the on-board camera with those received from the smart object. When a match is found, the robot localizes the object in its range of vision and, then, it can move towards it to perform some type of action.

The application scenarios for this type of systems include industrial and home applications. For instance, the system may be used in room storages to make it possible for robots to find and retrieve objects on request. The system does not require to store in the robot any information concerning the position or the actual appearance of the objects. Therefore, the robot may be instructed to find an object with a specific (and known) ID number and the robot will be able to recognize that object in a group of others by using its vision capabilities in conjunction with the information provided by the object itself upon request. Moreover, the proposed approach is scalable also in the number of robots. Indeed, every time a smart object transmits its appearance, more than one robot could listen to the description. Distributing the knowledge in the environment (i.e. in the objects) makes it possible to seamlessly work with one single robot or with a team of cooperating robot.

Another possible application scenario may be in the context of office/home automation. Here, examples of smart objects may be the books of a library (motes may be used to collect and reposition the books on the shelves after the closure), printers (motes may control the toner and/or the paper level and signal when recharge is needed) and so on. In these cases, the smart objects establish a WSN that routes the request messages to the nearest robot. The robot reaches the object guided by the WSN. Once in place, the robot receive the appearance from the mote attached to the smart object and locate it in the camera image. Eventually, the robot performs the required action. In a home scenario, the same system can be used to tied-up the play-room of kids. At the end of the day, a robot can locate all toys in the room and store them in the right containers. Both the toys and the containers are smart objects in the sense explained above.

Our work is related to the concept of ambient intelligence in which the intelligence is distributed among the agents and the objects that the agents can manipulate. There are many point in common with the work of Broxvall et al. [2], in which the concept of *Ecology of Physically Embedded Intelligent Systems*, or PEIS-Ecology is introduced. In fact, in the PEIS framework a large number of sensors are attached to objects and can transmit to the robot useful information [3] (i.e. the position of objects, the light in the ambient, etc.). Similarly, in this work each mote is associated with only one object and can help the robot to recognize that given object and locate it in the environment. The proposed approach is scalable to the number of objects and capable of discriminating between objects with similar appearance (i.e., different items with similar shape and color).

A. *Related work*

Our approach was inspired mainly by the papers reported in the followings. In [10] the poses are incrementally added to the cloud descriptor of the object; during the insertion, the descriptors of the new pose are compared to the model built. Similar sights are fused in unique cluster. If the new sight is different from some portions of the model then the correspondent descriptors are linked. This strategy improves the robustness of the objects description without considerably increasing the number of descriptors/sight. This method is quite similar to our method (differences arise in the data structures used).

In [12] a method to rebuild a 3D geometry of an object is described starting from a series of photos. The system uses the correspondences between the descriptors of various poses to build a 3D object model keeping track of descriptor positions. Another method ([11]) has used the vocabulary tree to increase exponentially performances during the comparison of many objects with very huge databases of lots of objects. The descriptors are patterned using the hierarchical k-means clustering recursively applied to descriptors space. Each level of the tree represents the division of the descriptors space into a certain number (k) of regions. Every sub-tree iteratively divides again each region. Similar descriptors of different objects will end up in similar clusters, therefore assuming similar paths in the tree. The search phase of the objects in a query image is very efficient.

II. DESCRIPTION OF THE EXPERIMENTAL TESTBED

In this section, we describe the main components of the testbed we have set up to provide evidence of the feasibility of the proposed system.

A. *Robot*

The robot used in this work it is a custom-built wheeled differential drive robot called Bender. Its hardware is based on a Pioneer-2DX ActivMedia motor control-board and a the motherboard of a desktop PC with a CPU Intel Pentium 4 a 1,6 GHz and 256 MB of RAM. The sensor mounted on the robot are: a camera, encoders mount on the wheels, and a mote attached to the robot via USB.

B. *Mote*

The mote attached to the robot is that same as those constituting the WSN. They are Tmote Sky IEEE 802.15.4 compliant. The main core of the mote is the MSP430, a Texas Instrument microcontroller that is well designed for the application on which low-power consumation is desiderable. These motes have integrated sensors, radio, antenna, microcontroller and programming capabilities. In particular, they have an integrated circuit measuring the power of the received signal which can calculate the Received Signal Strength Indicator (RSSI) of each received message. They are fit with 1MB of flash memory. They runs TinyOS, the open-source operating system developed for wireless sensor motes and they are programmed in NesC.

C. *Software Architecture*

The robot's software architecture is composed of four modules:

- **Robot navigation:** This module uses rather standard navigation techniques to reach the destinations decided by the higer levels modules. For space constraints, we will not further describe it.
- **Objects identification:** This module identifies objects in the environment. It extracts descriptors (SIFT) from camera images and compares them with those received by the smart objects;
- **Communication:** This module is in charge of the communication between objects and robot. Its aim is to send to the robot the descriptors of the object in an incremental and energy efficient way. This module is controlled by the behavior module so that only the information deemed necessary to find the object are requested.
- **Robot controller:** This is the high-level robot controller. It makes decisions using information provided by other modules and decides what actions the robot will make.

III. Object identification

As mentioned, the object recognition and location is visually performed, by exploiting **S**cale-**I**nvariant **F**eature **T**ransform (SIFT) descriptors. The SIFT, introduced by David Lowe in 1999 [8] [9], can be used to extract descriptors from an input image to find a match on a set of different reference images. Extracted features are particularly robust to affine transformation and occlusion. Unfortunately, they are not robust to perspective transformations nor to rotations of the object. Therefore, we developed an ad-hoc technique to deal with these transform.

In this work we use SIFT descriptors to identify smart objects in the enviroment. The SIFT features are so compact that can be stored in the motes. The robot compares these descriptors with SIFTs extracted from images grabbed by camera. The SIFT extraction and comparison is fast enough to allow object recognition in second half.

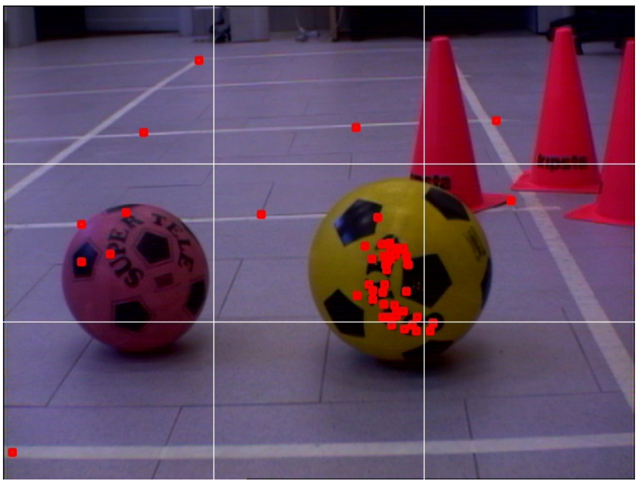

Fig. 2. What the robot sees. The red little squares are the SIFTs in the robot's live camera image that best match the SIFTs sent from the mote. In this example, the yellow ball is successfully identified among the red ball and the red cones.

A. Matching objects under affine transforms and occlusions: the naive approach

In order to present our method we need to introduce the original simple approach to object recognition using SIFTs. Our improvements will be presented in the next paragraph.
The object identification process begins by processing the images Im_1 taken by the onboard camera in order to extract a list I_1 of SIFT descriptors. This set of descriptors is compared with the list of descriptor I_2 associated to the reference image Im_2 of an object, and passed to the robot via wireless communication. If there is a minimum number of descriptors of I_1 fairly similar to I_2, we can assume the object appears in the image.

The recognition of the object in the image acquired by the onboard camera encompasses three steps, which are repeated for each descriptor x of I_1:

1) Find the descriptors y_1 and y_2 of I_2 that are closer to x (using the Euclidean norm as distance);
2) If the ratio between the distance of y_1 and y_2 from x is less than a certain threshold (in this case we used 0.49 as suggested by Lowe [9]) the correspondence between y_1 and x is accepted;
3) The process is repeated until I_1 is empty or the number of correspondences is greater than a threshold. In this case, it is likely that the target object is present in Im_2.

The object identification process does not require the association of all SIFT components of the object with all SIFT components extracted from the current frame. We required just 10 matches between the two SIFT descriptors.

B. Extending this approach to whole object surface and improving scale invariance

Using SIFT descriptors it is possible to identify an object only if both the original photo and current camera image are taken from a similar point of view. If this does not happen (i.e. the angle between the viewing direction corresponding to the original photo and the one of the current camera is larger than 20 degrees) the number of correspondences is generally not enough. To solve this problem one can take several images of objects acquired from different points of view. In this way, for every possible observation point the current input image can be mapped to the correct reference image with a transform that is approximately affine.

The naive solution could be of taking pictures of the object from all possible points of view and iterate the matching procedure described in the previous section for each reference image of the object and, in the end, to select the pose with the highest number of correspondences. If the number of correspondences is under a certain threshold the object is assumed to be not present in the image. This solution could potentially enable object recognition from any point of view. The drawback is that the time required to evaluate correspondences would drastically increase, because the number of descriptors increases quickly. We estimated that the scanning of a dataset of 100 images would require more than 60 seconds. To reduce the workload, it is hence necessary to decrease the number of comparisons between descriptors,

without loosing the identification capability of the procedure. Another problem regards the scale invariance of SIFTs. It is true that the SIFTs are good scale invariant descriptors, but we see that this statement holds until a certain limit. We tried to identify objects at 5 mt. from the robot with images taken at 1 mt. and we had poor matching scores. To address this problem we took images of the object at different distances from the robot. Obviously this approach generates a lot of useless redundant informations, slowing down the computation.
In the next paragraph we present the technique we used to address all these problems.

C. *Efficient object recognition with lots of images*

A smarter solution to object recognition using information coming from lots of images consists in exploiting the correlation among images taken from closer points of view to reduce the number of reference images. If an object is photographed from two points of view less than 20/30 degrees apart (or with similar distances from the object), similar descriptors will be extracted from them [9]. This fact can be used to avoid comparisons with all possible observation poses. This improves efficiency, though some redundant information remains.
Moreover, we would like to merge all descriptors into a big *descriptor cloud* and do comparisons between this cloud and descriptors extracted from the camera images (instead of compare each reference image with camera images). This leads to a big problem: if we take all the descriptors and stuff them into this cloud indiscriminately, we lose identification capability. This fact follows from some previous considerations: we said that similar reference images produce similar SIFTs, because the part of surface common to both the images generates very similar descriptors. Now, let think about the descriptor matching algorithm described at the beginning of this section. For each descriptor x coming from a camera image the algorithm finds y_1 and y_2 belonging to the cloud. Let suppose that y_1 and y_2 belong to similar reference images (and in particular that they describe the same portion of object surface). This means that probably they are similar. This raises the probability that the ratio between the distance of y_1 and y_2 from x is higher than the threshold (step 2 of the algorithm) and so the descriptor matching will be rejected. If we had used only a single image we would be able to match the descriptors. The problem is related to the presence of redundant descriptors in the cloud. We should build the cloud adding only the descriptors that are not too similar to the descriptors already present in the cloud.
The solution we adopted in our work removes the residual redundancy by merging the corresponding SIFT descriptors from the neighbor reference images in an incremental way. Let define $Z = \{Z_1, Z_2, ...\}$ as the set of reference images. The descriptor cloud is D_c.

1) Set $D_c = \emptyset$.
2) For each reference image Z_i in Z do:
   a) For each descriptor Y_j extracted from Z_i:
      - if $\exists\ Y_s \in D_c$ so that Y_j is similar to Y_s then discard Y_j
      - else add Y_j to D_c

The result of this union completely describes the external surface of the object. Searching for objects in a frame is hence done by looking for correspondences on this set. We will see that using this method to build the cloud of descriptors we reach the same identification capability of the naive algorithm but we speed up the entire computation.

D. *SIFTs robustness and matching results*

We will now describe the robustness of our object recognition approach.
SIFTs are highly discriminant descriptors. We saw, both in with datasets and real enviroments images, that target object is always identified (under certain conditions discussed later) (see Figure 3 and 4).

Fig. 3. Ball match test in a complex enviroment. Target object is the ball represented in Figure 1. Colored dots represent matching descriptors. There are only few incorrect descriptor matches

Fig. 4. match test in a complex enviroment. Target object is the carton represented in Figure 1. Colored dots represent matching descriptors. There are only few incorrect descriptor matches

SIFTs are invariant to rototraslations, occlusions and (partially) to scale. We address the problem of scale invariance taking images at different distances from the object surface. This allows to build a system completely robust to scale. We see that the robot is able to recognize objects indipendently from the distance from the camera (see Figure 5).

We also investigated the robustness to light changes. We saw that light changes in the enviroment doesn't compromise the identification capability of our system. On the other hand SIFTs are not robust to non-linear light noise. This means that it is hard to identify an object partially in shadow and partially hitted by a spot light (see Figure 6).
SIFTs are able to distinguish between similar shape objects with different colors and pattern on the surface (see Figure

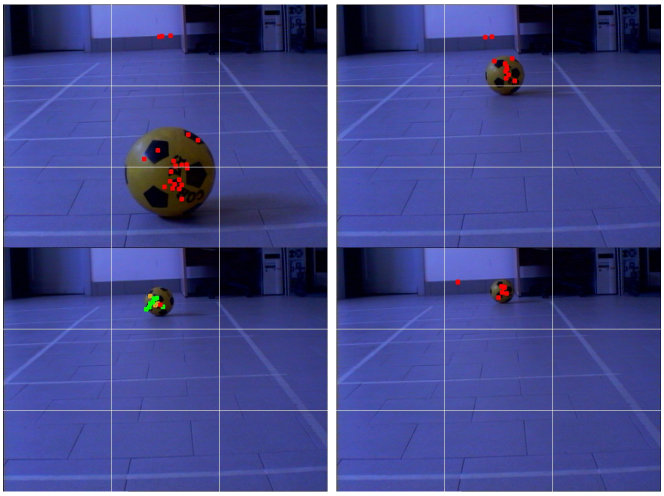

Fig. 5. Ball match test with increasing object ditance. Target object is the ball represented in Figure 1. Colored dots represent matching descriptors. There are only few incorrect matches

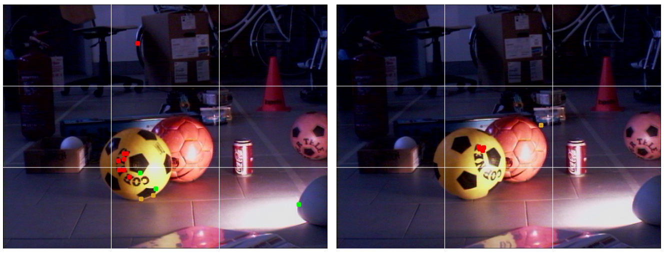

Fig. 6. Ball match test in presence of a spot light. The area hitted by light is not well identificable (right image).

2). The biggest problem is that the number of descriptors is low for uniform regions. This means that, while objects with patterned surface are well identified because there are lot of highly specific descriptors, uniform regions are not well characterized. These regions will produce few low discriminant descriptors. An example may be found in Figure 2: correspondences between descriptors are found only in the surface covered by the word *Corner*, while there is a lack of descriptors in the uniform yellow and black areas.
We suppose that the use of hybrid methods which use SIFTs for patterned regions and some other technique for uniform region matching would improve our method.

E. Decrease time extract SIFT and match

The extraction of the SIFT descriptors from images is accomplished by Hess implementation of Lowe SIFTs. [22]
We saw that extracting the features from the whole image requires a lot of time (at a resolution of 640x480). Though, we can reduce computing time extracting the descriptors only from a part of the image (ROI). We divide the whole image in three vertical band ROI. When we don't know where the object is, the ROI is chosen randomly.
There are two cases:

- object appears inside the active band; then we can select a subROI for next object recognition merging the region where the object appears and the zone where we guess the object will appear after the robot moves.
  If we know that the object had been identified in a certain zone of the image it's useless to extract descriptors from the whole image. This approach speeds up the computation if the estimated object position is good enough;
- if object doesn't appear we change the ROI and request more SIFTs and for next match we will have more features.

Time complexity linearly decreases in proportion to the size of the area considered.
In order to speed the matching process, we use kd-tree as data structure to organize the cloud of features. Each object has a separate cloud. The time complexity is $O(n \log n)$ instead of $O(n^2)$ of naive method.

IV. Communication Module

The communication module manages the trasmission of SIFT descriptors from sensors to mobile agents.

A. SIFT packet format

The SIFT descriptors are stored in the mote's memory with an ad-hoc program written in NesC. A single SIFT descriptor, named SIFT block, occupies approximately 182 bytes. Since the IEEE 802.15.4 protocol data unit (PDU) has a payload of 28 bytes only, each SIFT block needs to be fragments in approximately 7 PDUs for transmission. Each SIFT block is assigned a 2 bytes signature (Sift Identifier, SID) that marks all the fragments of that block, thus making it possible to recognize the fragments of the same original SIFT block at robot side. Beside the SID field, each PDU also carries a sequence number (SN) field, of only 3 bits, that specifies in which order the fragments have to be reassambled. The general frame format of an 802.15.4 PDU is shown in Fig. 7. Furthermore, we dedicated the first two bytes of the payload to carry the SID field, leaving the remaining 26 bytes for the SIFT data fragment.

B. The communication protocol

The communication protocol was designed to provide the following features.

- Identification of various motes (ID and type of mote). This guarantees the possibility of using a large number of (even heterogeneous) motes.
- Support of different packet types. Currently the protocol supports four types of packets: data packets for the transport of SIFT descriptors, a control packet for soliciting the transmissions of SIFTs, and two packets for management and signaling purposes (HELLO, SLEEP);
- Fragmentation and reassembly of data packets carrying SIFT descriptors, with a mechanism for the identification of corrupted/lost segments.

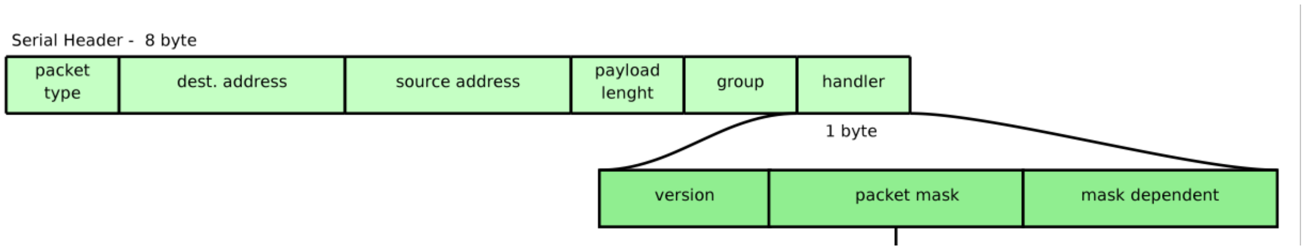

Fig. 7. General 802.15.4 PDU frame format (payload of 28 byte not shown).

- Incremental transmission of SIFTs upon request, to reduce the traffic, thus saving energy and better scaling in presence of multiple agents in the same area.

In order to allow the unambigous identification of each object, each mote is identified by a unique ID, which is also used for labeling each packet the mote sends. The protocol is connectionless and unreliable, i.e., no acknowledgement is required to confirm the correct reception of SIFT data packets. This choice is motivated by the robustness of the the SIFT encoding to the erasure of some descriptors. Anyway, if the number of corrupted or lost packets is too large, the robot can require the retransmission of the missing SIFT descriptors.

C. MoteObj program

The MoteObjs are ruled according to a finite-state machine. Upon receiving a SIFT soliciting message, the MoteObj begins sending packets containing the requested SIFTs. When finished, it returns to the idle state. If an HELLO message arrives from a MoteRobot, the MoteObj replies with its own HELLO to signal its presence. Finally, the robot can send a particular message (SLEEP) which invites the MoteObj to enter sleep mode. These messages are sent to all uninteresting MotesObj in the area of the robot to minimize energy consumption. Energy efficiency is also pursued by switching on and off the radio transceiver of the MoteObj according to a regular pattern. The radio interface can be used only during the active period. Clearly, this introduces a certain delay to collect the feedback from all the nodes in a given area. Another strategy to reduce energy consumption consists in requiring a progressive transmission of the SIFT descriptors, in a way that makes it possible to immediately discriminate between interesting and non interesting objects, thus permitting to the latter to immediately switch off their transceivers.

D. MoteRobot program

The mote connected to the robot is identified as MoteRobot and is connected via serial connection through the USB port. Its function is to provide an interface with the external world to allow the communication of the robot with the motes installed in the objects. There are two working modalities:

- The HELLO mode provides the periodical sending of packets to elicit the motes in the area to respond. In this phase, receiving power statistics are available in a packet's field and are passed to the robot in order to makes it possible to perform a (very rough) estimate of the distance between robot and mote. The received radio strength indicator (RSSI) can also be used to map the smart objects in the environment, as described in [18].
- The second mode is used when the robot requests the SIFT descriptors from the smart objects or when it requires a retransmission of lost packets, when needed.

V. Robot Controller

At the beginning, the robot enters the environment in *HELLO Mode* and waits for *HELLO* packets from the motes attached to the objects. The robot controller module saves the ID of the objects present in the environment and chooses which objects to identify based on the high-level plan.
When the robot finds the objects of interest it switchs to *Discovery Mode*.
The *Sleep* message is sent to uninteresting motes and the *SIFT Request* packets are sent to the potentially target motes. This is done to minimize the battery energy consumption of the sensors and to avoid interference with other active motes. Received SIFT descriptors are passed to the object identification module. The robot controller executes the following steps:

- Grab an image with the on-board robot's camera;
- Calculate the SIFT descriptors of this image;
- Match a subset of these SIFT descriptors with those arrived from objects' motes;
- Return object presence and position to navigation module.

VI. Results and Conclusions

To test our system we implemented a setup with one mobile robot and four smart objects. Tests took place in a room cluttered with forniture and other objects (i.e. *non-smart* objects) and with people moving around (see Fig. 2). The robot's task was to identify two of these smart objects. The robot is able to correctly identify the objects of interest in the environment and to move toward them. In our tests the robot looks for one object at a time. The robot is able to check object presence in the current camera frame at 20 fps. In this way, the robot can move while looking for the objects. No false positives were detected. The false negative detection events (i.e. missed object detection) are due to packet loss and in the consequent delay in SIFT descriptor re-trasmission. As already explained, the SIFT represents a descriptor particularly robust to these events compared to other approaches for object identification/recognition. However, like many object descriptors, they have limited robustness to strong lighting changes. We notice that strong variations of the light conditions may generate mismatches.

We wish to stress that, although in out experiments we used SIFT descriptors, the proposed system and WSN communication protocol can be as well used with other image encoding techniques. Furthermore, we remark that the communication protocol was designed with a particular concern about mote energy efficiency. To this end, the intervals of sleep, mote discover, and progressive description transmission are parametrized in order to make it possible tuning the protocol for different applications.

A. Future works

A possible advance could be to further improve the energy efficiency. We propose the use of the CQS-Pair [15] method allowing sensors to communicate with high efficiency in a scalable way. The method actually used is too simple to handle hundreds of motes. In order to improve real-time object recognition when hundreds of smart objects are in the environment, we think that the use of a Vocabulary Tree [11] will speed up the global recognition process. This allows to identify objects indipendently from the order they are specified. Actually our system looks for one object at a time. The use of vocabulary trees as described in [11] should extend out method merging all object descriptor clouds in a global *query descriptor cloud* reducing the number of total comparisons. Yet another improvement regards the recognition of objects with uniform surface. In this cases the number of SIFTs extracted is lower than in the case of an object with a patterned surface. A very small number of SIFT descriptors makes the match more difficult. We propose the hybrid use of SIFT descriptors and an uniform region match technique in order to support both cases (uniform and fully patterned object surfaces).

VII. ACKNOWLEDGMENTS

We wish to thank Alberto Pretto for his suggestions and the fruitful discussions on this project.

REFERENCES

[1] *G. Zanca, A. Zanella, F. Zorzi, M. Zorzi: "Experimental comparison of RSSI-based localization algorithms for indoor wireless sensor networks"*, In Proceedings of REALWSN'08, pp. 1-5. Glasgow, Scotland, UK. April 1. In conjunction with ACM EuroSys 2008.

[2] *Mathias Broxvall, Marco Gritti, Alessandro Saffiotti, Beom-Su Seo, Young-Jo Cho : PEIS Ecology: Integrating Robots into Smart Environments.* Proceedings 2006 IEEE International Conference on Robotics and Automation, ICRA, 2006, pp. 212-218 .

[3] *Alberto Sanfeliu, Norihiro Hagita, Alessandro Saffiotti, Network robot systems,* Robotics and Autonomous Systems, Volume 56, Issue 10, Network Robot Systems, 31 October 2008, Pages 793-797, ISSN 0921-8890

[4] *OpenCV Library:* http://opencvlibrary.sourceforge.net/

[5] *GNU Scientific Library: a numerical library for C and C++ programmers*

[6] *Hess, R., Sift Feature Detector A C implementation of a SIFT image feature detector:* http://web.engr.oregonstate.edu/~hess/index.html.

[7] *API and Tutorial NesC:* http://www.tinyos.net/tinyos-2.x/doc/

[8] *Lowe, D. G., "Object recognition from local scale-invariant features":* Proceedings of International Conference on Computer Vision, 1999, pp. 1150-1157.

[9] *Lowe, D. G., "Distinctive Image Features from Scale-Invariant Keypoints":* International Journal of Computer Vision, 2004 , pp 91-110.

[10] *Lowe, D. G., "Local Feature View Clustering for 3D Object Recognition."* Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2001. CVPR 2001, pp 682-688 .

[11] *David Nistér and Henrik Stewénius, "Scalable Recognition with a Vocabulary Tree."* IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2006, pp 2161-2168.

[12] *Iryna Gordon and David G. Lowe, "What and Where: 3D Object Recognition with Accurate Pose."* Toward Category-Level Object Recognition, eds. J. Ponce, M. Hebert, C. Schmid, and A. Zisserman, (Springer-Verlag, 2006), pp. 67-82.

[13] *Columbia University Image Library COIL-100: www1.cs.columbia.edu/CAVE/software/softlib/coil-100.php*

[14] *OpenCV Library: http://sourceforge.net/projects/opencvlibrary*

[15] *Shouwen Lai, Bo Zhang, Binoy Ravindran and Hyeonjoong Cho:CQS-Pair: Cyclic Quorum System Pair fot Wakeup Scheduling in Wireless Sensor Networks.* Lecture Notes in Computer Science, 2008, pp 295-310.

[16] *C.H.J.R. Jiang, Y.C. Tseng and T. Lai: Quorum-based asynchronous power-saving protocols for IEE 802.11 ad-hoc networks.* Proceedings International Conference on Parallel Processing, 2003, pp 257-264.

[17] *W. Luk and T. Huang: Two new quorum based algorithms for distributed mutual exclusion.* Proceedings of the 17th International Conference on, Distributed Computing Systems, 1997, pp 100-106.

[18] *Menegatti Emanuele, Zanella Andrea, Zilli Stefano, Zorzi Francesco, Pagello Enrico:Range-Only SLAM with a Mobile Robot and a Wireless Sensor Networks.* 2009 IEEE International Conference on Robotics and Automation, May 12 - 17, 2009, Kobe, Japan.

[19] *Azad, P. and Asfour, T. and Dillmann, R. Combining Appearance-based and Model-based Methods for Real-Time Object Recognition and 6D Localization*IROS 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, Oct. 2006, Beijing, China, pp.5339-5344

[20] *T. Asfour, P. Azad, N. Vahrenkamp, K. Regenstein, A. Bierbaum, K. Welke, J. Schroder, R. Dillmann, Toward humanoid manipulation in human-centred environments,* Robotics and Autonomous Systems, Volume 56, Issue 1, Human Technologies: 'Know-how', 31 January 2008, Pages 54-65, ISSN 0921-8890

[21] *Vincze, M.; Schlemmer, M.; Gemeiner, P.; Ayromlou, M., "Vision for Robotics: a tool for model-based object tracking,"*Robotics & Automation Magazine, IEEE , vol.12, no.4, pp. 53-64, Dec. 2005

[22] *Hess SIFT implementation:* http://web.engr.oregonstate.edu/~hess/