Towards a Network Robot System for Object Identification and Localization in RoboCup@Home

Alexei Borissov, Jakob Janecek, Federico Pecora and Alessandro Saffiotti
AASS Mobile Robotics Lab
Dpt. of Technology, Örebro University
S-70182 Örebro SWEDEN
<name>.<surname>@oru.se

***Abstract*— This paper describes a realization of a network robot system for autonomous object localization and identification. Developing a "Lost & Found" capability, the use of which can be envisaged in a wide range of applicative domains including domestic assistive scenarios, is a challenging task for current AI and robotic technology. Indeed, this task is currently one of the core challenges within the RoboCup@Home competition.**

A number of approaches for implementing a robust and general Lost & Found functionality are feasible. In this paper we present a solution which integrates state-of-the-art intelligent software, robotic and sensory components in a distributed network of cooperating modules. This article describes the design and implementation of the system, provides a preliminary experimental evaluation and discusses the applicability of our approach to the RoboCup@Home challenge.

I. Introduction

This paper deals with autonomous object localization and identification within the general setting of domestic assistance, where objects are generic household items. Our goal is to obtain a system that can provide a "Lost & Found" capability which does not rely on ad-hoc characterisitcs of the objects (e.g., special colored tags, or any other feature that can be added to facilitate object identification).

The Lost & Found functionality has important applications in a wide range of applicative domains, including domestic assistive scenarios for elderly people affected by cognitive decline (e.g., memory loss). Indeed, this task is currently one of the core challenges within one of the robotic community's principal benchmark, namely the RoboCup@Home league [10] of the RoboCup competition [9]. The aim of RoboCup@Home is to foster the development of useful robotic applications that can assist humans in everyday life. The participants of the Lost & Found challenge are required to provide a system that will find a number of objects in an environment without human interference. Each participant is given some time to teach their system the set of previously unknown household objects, after which the objects are placed randomly in a mock-up domestic environment and the robot is then required to locate them.

Implementing Lost & Found is a challenging task for current AI and robotic technology. At least two general approaches for implementing a robust and general Lost & Found functionality are feasible, from a 'super-robot' approach in which one robotic platform performs the entire task by itself leveraging its mobility to explore the environment, to a purely distributed approach relying on pervasive fixed vision within the environment. In this paper we present a solution for autonomous object localization and identification which attempts to combine the best of both approaches. The solution is realized and deployed within a network robot system and relies on the integration of several software, robotic and sensory components distributed within our smart home test environment. Our solution leverages the availability of a particular network robot system consisting in an ecology of Physically Embedded Intelligent Systems (PEIS), the PEIS-Home [11]. The infrastructure underlying this approach to networked robotics essentially consists in a middleware for facilitating the exchange of information and cooperation between heterogeneous PEIS. A PEIS can be any software, robotic or sensory component offering a partiular functionality or service.

With the availability and affordability of robots such as the Pioneer [1] and the Roomba vacuum cleaner [4] and with the wide use of cameras for surveillance and baby monitoring, the idea of combining these technologies towards the aim of developing a Lost & Found functionality is particularly attractive: they are affordable, easy to setup and use, and require little maintenance. If, for example, the robot was specifically designed for vacuuming the floor but has access to surveillance cameras in the house, it is now able to find misplaced objects that the user instructs it to find. If the camera happens to break, the robot can still perform its designated task of strictly vacuuming.

In this article we describe our implementation of the Lost & Found task using the PEIS middleware and a number of state-of-the-art vision algorithms. The fundamental intuition underlying our approach is to break the task down into a global, fixed-vision based object localization stage followed by a local verification of candidate objects on board a mobile robot. The former phase provides a coarse but extremely fast estimate of the location of the object(s) which are to be found, while the latter refines this information to assess whether the sought after object(s) match the initial hypothesis. Also in this article we present a preliminary experimental evaluation of the system, in which we measure both the accuracy of the global estimation and the performance of the entire system. Finally, we provide a discussion on the advantages and disadvantages of the approach as well as on the adecquacy of the implemented approach to the

In: Proc. of the IROS-08 Workshop on Network Robot Systems. Nice, France, 2008.

RoboCup@Home challenge.

II. System Overview

As mentioned, there are a number of approaches to tackle the Lost & Found problem. These solutions range from informing the robot on a semantic level of the approximate position of the object (e.g., in the kitchen) and having the robot go there and confirm the fact that it is there. This of course would require prior knowledge about the object and where it was placed. Another approach would be to simply have a robot wander around and analyze everything it sees and attempt to match it to an object that it is looking for. This approach is not only inefficient but also very computationally expensive as feature-based comparison to match each scene against a known image can be very memory intensive. Random navigation in the environment could take a very long time, even if a map of the environment is known and the robot is able to keep track of where it has been. A third alternative consists in using very good fixed cameras that provide a high quality image covering the whole environment. These cameras would potentially have to be able to zoom on an object in order to verify features at a local level. This camera feature would come at a high cost, would still require a robot to fetch the object.

In a domestic scenario in which robots potentially are already deployed to address other needs, it makes sense to use affordable, off-the-shelf technology which can complement the robot's functionality. The implementation of the Lost & Found functionality described herein relies on the use of inexpensive fixed video cameras found in any electronic store and a mobile robot used for domestic cognitive assistance. This, together with the fact that we use the PEIS-Ecology middleware, makes the system configurable as well as expandable (e.g., by adding more cameras to extend the reach of the Lost & Found functionality).

Given the task of locating objects in an environment, it is first necessary to teach the robot an object and then of course have it go locate it. A clean line may be drawn between these two phases and as such we developed two different modules; a learning module for data acquisition and a seach module for finding objects. There is no overlap between the two except for the fact that the learning module saves the information into a knowledge base that the searching module then accesses in order to perform its task. More about each module is described in Sections III and IV respectively.

Our approach to object identification and localization effectively decomposes the problem of seraching for objects into two steps. The first step consists in finding potential points of interest (candidates) with a fixed vision system able to see the whole environment. The second step of the process is to navigate to the candidate's position and attempt to use a Pan Tilt Zoom (PTZ) camera on the robot to verify that an object is the one of interest. The process can be visualized in Figure 1. In both steps, a combination of state-of-the-art artificial vision algorithms are employed to assess the presence within the scene of known objects.

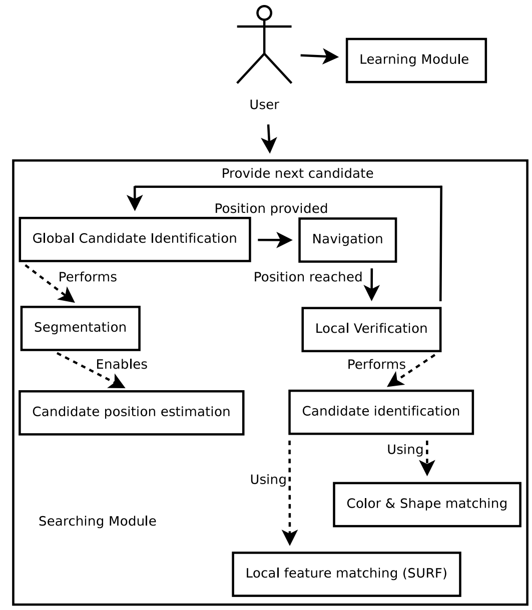


Fig. 1: General flow diagram

The advantage of using a two-step approach is that since the fixed vision system is able to see the whole environment, we are able to quickly identify possible candidates and roughly estimate their position thus eliminating the need for a random walk of the robot. The rough position estimate is acceptable for our task since we have a mobile robot platform with a PTZ camera through which it can inspect candidate objects more closely (as long as the robot is 'in the area' of the object). Furthermore, if some sort of a pre-processing algorithm is applied, false positive candidates may be eliminated thus reducing the list of possible candidates, and as a consequence the amount of physical seraching that must be performed by the robot.

III. Object Acquisition

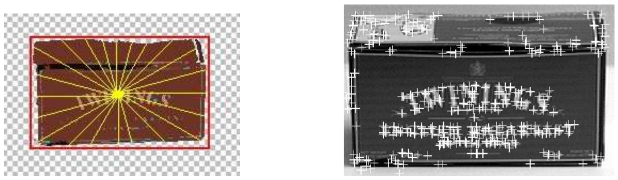
Object acquisition is a key component of the system. In compliance to the RoboCup@Home regulations, we have chosen to use the user as a teacher for our system. The system provides a Graphical User Interface (GUI) to enter the required data about each object. All information aquired through this learning module is stored in a knowledge base.

All information gathered in this phase is used during the search to identify the desired objects. In relation to the vision algorithms employed for object identification (explained in the following sections), we have chosen to capture the following traits for each object:

- **Object name:** `my_favourite_cup`, `pill_box`, `book`, ...
- **Color:** red, green, yellow, ...
- **Size:** physical object dimensions (length, width and height)
- **Shape signature:** 64 normalised distance measurements from the center of segmented object spanning 360° around the object (see Figure 2a)

In: Proc. of the IROS-08 Workshop on Network Robot Systems. Nice, France, 2008.

- **Local features:** features obtained using feature detector (see Figure 2b)



(a) Visual representation of a shape signature. (b) Example of local features. White crosses denote features.

Fig. 2: Shape signature and local features of an object.

The name of the object is used as a symbolic link within the system to connect the data acquired and therefore also as a key to represent the object. The color is of importance as it acts as the initial filter for both the global components and the mobile platform. The size, that is the physical dimensions of the object, is used to provide a position estimate for each color blob found in the fixed vision system (more on this in Section IV-A). A number of shape signatures are saved for each object describing the various viewpoints. These are utilised when searching for the object, where the signatures are used as a further detection filter for the identification of the object. The local features are obtained using SURF (Speeded Up Robust Features) [2], which builds on the same principle as the established detector SIFT (Scale-Invariant Feature Transform) [8], and is a relatively fast and robust method used in local invariant feature detection. As for the signatures, also features area detected at various viewpoints of the object, representing how the object could be seen in the environment. As this is the most computationally intensive of the algorithms, we aim to limit its use by performing the aforementientioed filtering stages. More on the local verification process is described in Section IV. Overall, the meaningful information for an object can be acquired through the GUI in three easy steps, the combined time requirement for which does not exceed three minutes.

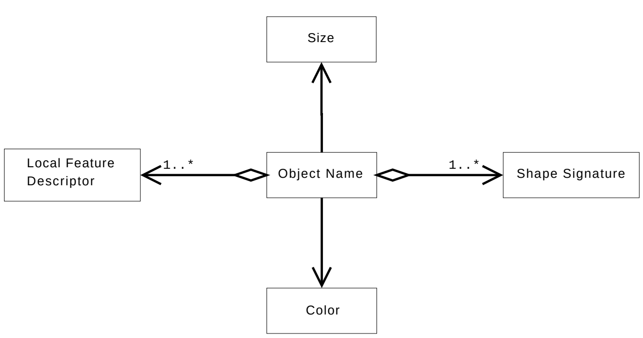


Fig. 3: Overview of object properties

The details of the learning module are outside the scope of this paper, and the interested reader is referred to [5] for further details.

IV. Co-operative Search

The search procedure may be called co-operative as it combines fixed vision and mobile vision (PTZ camera) that is located on a (mobile) robot platform. This is done, as previously mentioned, using a two-step process in which the global part of the system (fixed vision) is responsible for finding candidates and localizing them in the environment while the local part (robot) then navigates to these points and furhter investigates these candidates. An overview of this process can be seen in Figure 4.

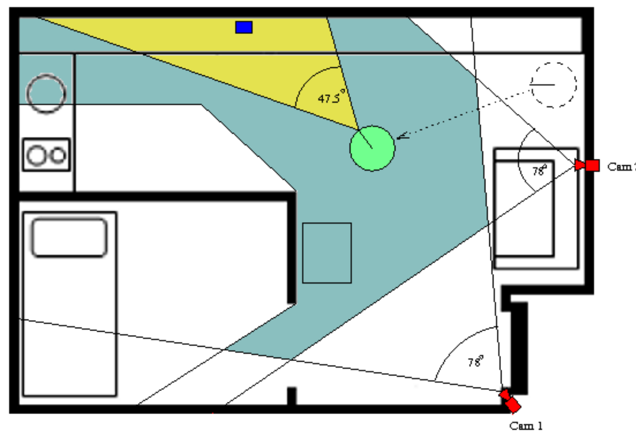


Fig. 4: Overview of the two step process

A. Global position estimation

The first step, global position estimation, is required to identify color blobs of interest and provide their position estimate to the second step of local verification. Given a number of objects to locate, this procedure analyzes the scene in order to identify possible candidate color blobs that mach the colors of the requested objects. This step is entirely based on color segmentation. In order to limit the amount potential matches, this step is provided with a *pre-processing* phase consisting of backround subtraction and noise filtering. Following this, *position estimation* occurs on the resulting image, on which fuzzy matching is performed to obtain an estimate of the position of each blob.

Pre-processing. Finding candidate blobs is a rather trivial process as there are a number of approaches that exist. The simplest and perhaps easiest is to use color segmentation. Using this approach, color blobs found in the image become points of interest. Notice though, relying on segmented blobs alone may lead to numerous false positives; if we were to look for a red box, everything red, including red TV lights, red ornaments, red crayons, everything red would be returned; see Figure 5b or Figure 5e.

Since the global part of the system provides position estimates of all blobs found to the robot to further explore, providing a list as concise as possible is desireable, as this will minimize the time needed to investigate all candidate locations. We achieve the desired minimized list by performing background subtraction. Such a technique is acceptable

In: Proc. of the IROS-08 Workshop on Network Robot Systems. Nice, France, 2008.

in the scope of the RoboCup@Home challenge as we are able to extract objects from the environment to learn them giving the system an opportunity to capture the background image. Once the data acquisition for all objects is complete, they are placed into the environment and remain static until the completion of the task. These circumstances allow us to capture foreground images (Figure 5a and Figure 5d) and trust the resulting image until the test is complete. The images in Figure 5c and Figure 5f are the results of performing background subtraction followed by a number of image processing techniques to reduce noise and enhance the regions of interest on two cameras that we use[1].

Position Estimation. Taking the background subtracted images from the previous section as input, we have developed a fast and simple position estimation for color blobs using a method based on [6], which uses fuzzy logic and fuzzy sets. As we use a mobile robot to perform local verification of the objects in question, it is enough to acquire a more general estimate for the position of said objects from the global perspective. The information gathered from these images is the width of the color blobs in pixels. Specifically, this measure is determined using the Watershed segmentation algorithm [3]. The position and orientation of the camera as well as the maximum and minimum dimensions of the objects we are looking for are known. Therefore, we can determine a distance range approximation from the camera source to the objects in question. The range of this distance approximation provides the *core* of our initial fuzzy set, which is represented by a trapezoid. The sequence in Figure 6 explains what happens at this point. Figure 6a displays the trapezoid generated by the range approximation from the camera source. A trapezoid is created, seen in Figure 6b, which is perpendicular to the range estimate trapezoid. This accounts for the uncertainty of the bearing of the object in the field of view of the camera. The *support* of the range trapezoid as well as the core and support angles of the bearing uncertainty trapezoid were chosen empirically. The two trapezoids are encased in a bounding-box, see Figure 6c, simplifying the calculations for the impending trapezoid intersections.

Following the creation of these bounding boxes for each object, the estimates are transformed into the global coordinate frame by taking into account the location and orientation of the fixed camera. Upon the completion of this step for each fixed camera, the resulting fuzzy sets can be matched, as shown in Figure 6d. The intersection of these sets signifies that both cameras perceive the same object.

As well as the global x and y position of the object, the *hue* and *saturation* components of the color blobs are used in matching the objects in view. The core and support for these are chosen to be wide enough to encapsulate all possible shades of the same color.

A significant method of increasing speed for the fuzzy intersections is in the search algorithms to find the best match. Instead of performing an exhaustive search by matching all possible targets from all fixed camera sources to find the maximal core overlap, we use a greedy search. This means that we use the first match that we find which corresponds to an intersection of all four trapezoids (range, bearing, hue and saturation) over a certain threshold. This reduces the search time significantly. This approach is acceptable for our purposes as we are not so concerned about the exact location of the object, but more of a general estimate for the global position which our autonomous robot can then navigate to.

[1]The further filtering process is described in more detail in [5].

B. *Local object verification*

The local phase is responsible for verifying whether an object corresponds to the hypothesis formulated by the global estimation. The initial step here is to navigate to the position provided by the global position estimation. Once there, it is necessary to attempt to locate the desired object within the field of view of the PTZ camera. Since the system knows the object's name and color, it proceeds to load its shape signatures. Using this information, it compares all segmented shapes seen by the PTZ camera against the shape signatures that correspond to the desired object. A candidate list is created and the candidate with the highest shape match is chosen from the list. The system then centers on the object and zooms in on it to perform local feature matching using SURF. As SURF is rather computationally expensive compared to color segmentation, we aim to reduce the number of times it is performed. This reduction is achieved by setting higher-than-required filters when matching shape signatures in the candidate identification step. These filters of course can be relaxed with ease at the sacrifice of running SURF more than necessary. After SURF is performed, if more than n features are matched against the images in the knowledge base, the object currently being examined is confirmed to be the desired one. If it is not confirmed, the list that was previously generated is accessed and the candidate with the next highest shape probability is selected and further examined.

V. Distributed Implementation

The hardware we use are two fixed Logitech QuickCam Fusion web-cams and a PeopleBot robot with a Pan Tilt Zoom camera and a SICK LMS 200 laser. These hardware components are integrated into the PEIS-Ecology middleware which enables easy and efficient information sharing with a high level of configurability. This makes the system flexible and allows other components to be easily added or removed. The cameras are placed in the corner of the room and in the middle of the room as can be seen in Figure 4. The PEIS-Ecology components we use and their purpose can be seen in Figure 7, where the numbers on the edges indicate the order of invocation of the various components for the Lost & Found task.

By using the PEIS-Ecology, we were able to easily fuse these componets together to achieve a network of modular functionalities. Indeed, the entire system provides a 'super-robot' behaviour although the actual functionality (both in terms of the necessary hardware and software components)

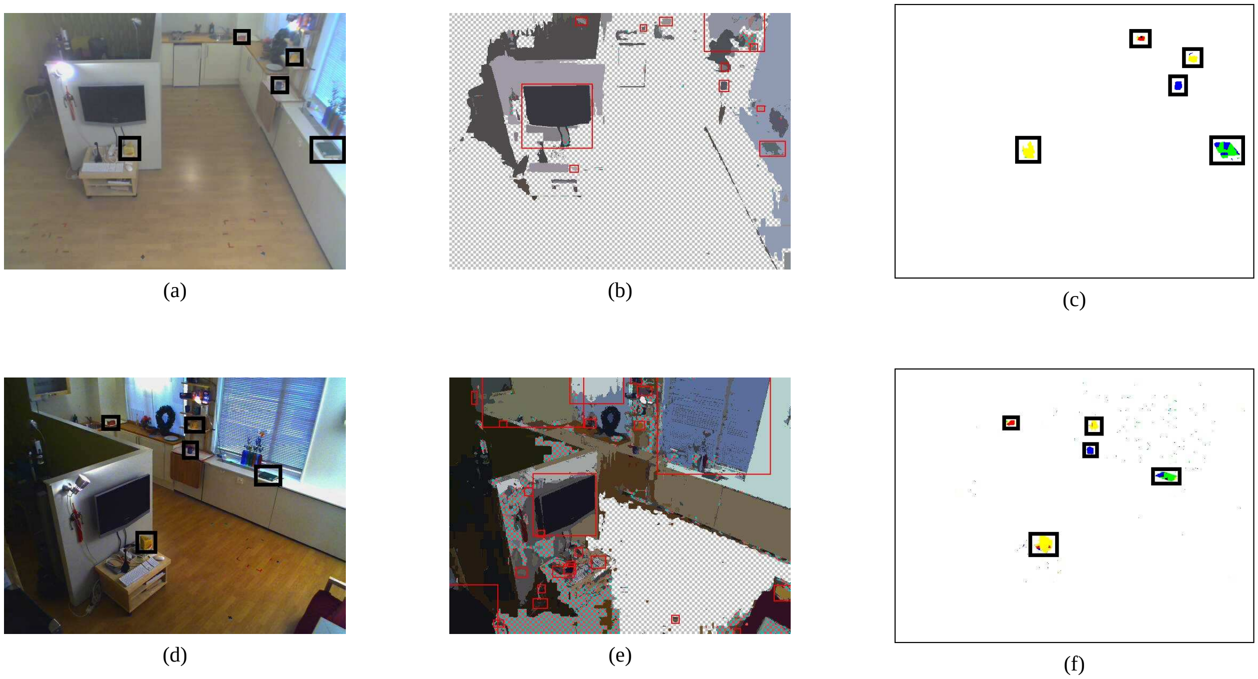


Fig. 5: Image background subtraction and filtering as it is performed by the two cameras (top and bottom rows). The first column contains foreground images, the second column the result of color segmentation. The third column shows background subtraction.

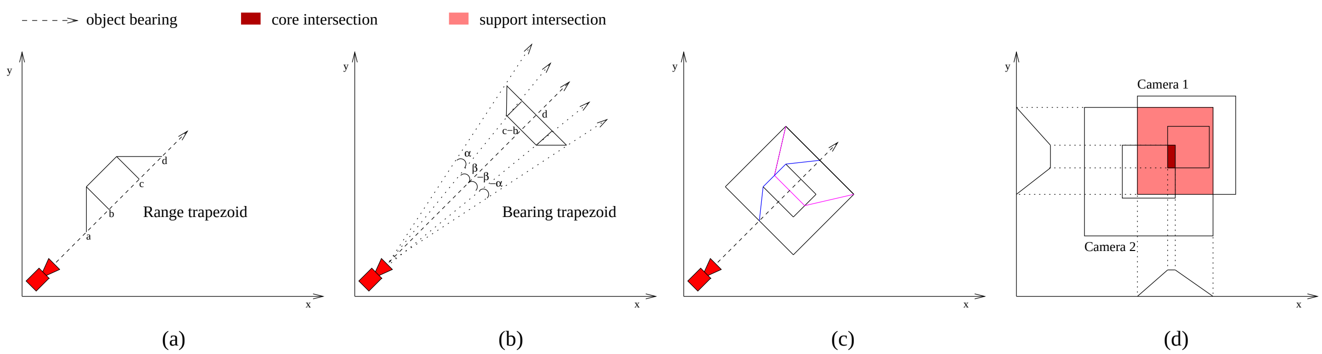


Fig. 6: Overview of the procedure for global position estimation.

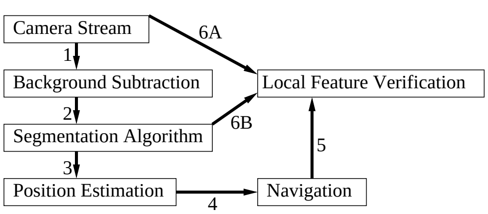


Fig. 7: Network of PEIS components providing the Lost & Found functionality (numbers indicate the sequence of activation of the components).

is provided by the network. The *Camera Stream* component provides a stream from any camera in the environment, static or PTZ, to either *Background Subtraction* or to *Local Feature Verification* components. The result is then either used by the *Segmentation* component to segment the image or by *Local Feature Verification* to verify that the object that is seen is one of the requested objects. Finally, the *Local Feature Verification* component provides SURF for local feature verification.

VI. Evaluation

For the evalutaion of the system, we chose a set of five different, everyday household objects which could be segmented using the color segmentation algorithms and which had enough features to be recognized using the SURF local feature detector. We present here a preliminary evaluation of both global position estimation and of the complete system. As well as having a set of pre-defined objects to use in the experiments, we also chose a set of five positions in the environment. The specific positions were chosen as they are all present in the field of view of both cameras as well as providing a significant representation of the possible realistic locations and heights that objects could be placed in the environment. In addition, we chose to allow for four different rotations of each object to determine whether the system would be affected by the various viewpoints of the objects, both in global position estimation and local verification. We felt that these positions and rotations would provide sufficient variety and challenging situations for the testing and evaluation of our system. We performed two types of test on the system. The first tested the accuracy of the global position estimation for each object. The second tested the overall performance of the system in a number of metrics.

A. Position Estimation

A total of 20 tests were performed on the global position estimation system on each of the five objects chosen, placing each one of them in each position and rotation mentioned earlier. Figure 8 shows the results for one of the positions. The average and standard deviation of the estimated distances from the actual position are shown.

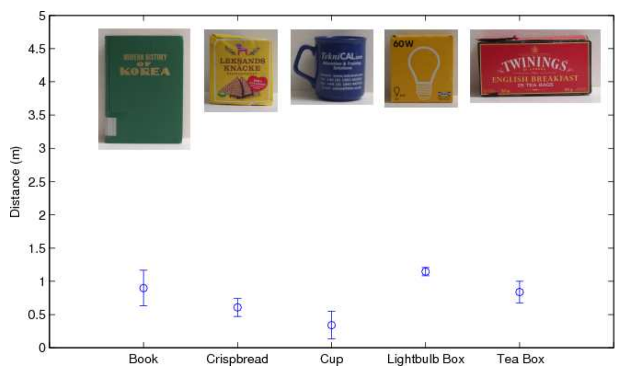


Fig. 8: Results for the five objects in one fixed position and four different orientations.

The results show that the position estimation of the objects depends strongly on the specific shape of the objects. Notice that the core of the trapezoid is calculated based on the width of the color blobs in pixels. As mentioned earlier, the position and orientation of the camera in as well as prior knowledge on the dimensions of the objects are taken into account. Nonetheless, the distance range approximation will inevitably be less accurate in the case of strongly asymmetrical objects such as a book. This explains the higher distance from the real position of the estimate. Also, asymmetry accounts for greater variability between estimations performed on different rotations. Conversely, objects with a rotation-invariant shape, such as the cup, lead to a more accurate position estimation.

Overall, the average global position estimation always lies between a few centimeters and $\sim$1.5 meters from the real position. Such estimates are acceptable in light of the following local verification step, where the mobile robot reaches the estimated position to ascertain the validity of the hypotheses.

B. Complete System

The second batch of tests involved running the system through its entire operational cycle, i.e., from the acquisition of background images to performing local feature verification. We randomly generated which object to find, at which position to place it and at what orientation to face it for each test. A breakdown of the results can be seen in Figure 9.

Here we highlight some important points about the results from this data. The success rate of the system (i.e., an object being correctly located and identified) was 70% (14 tests were successful out of 20). The average time for completion of a successful test was approximately 113 seconds. This does not include the time taken for the system to perform background subtraction, which requires an additional 10-15 seconds. Notice that the background subtraction and filtering step can be assumed to take constant time, and does not depend on the number of objects in the scene. Indeed, it is also easy to conceive a system in which this operation is constantly performed as a background process, in effect continuously updating the history of the background rather than performing it on user demand.

A test of note is test number 11. It is the only successful run which required the robot to rotate, and only one of two which found candidates other than the true object to perform local feature verification on. One should notice that by having an incorrect bearing to the position of the object can, in effect, quadruple the time taken to localize the object. Without this situation, the average time would have been below 100 seconds per run. Another important aspect to notice is that our results for false positives are perfect; 0 false positives were identified by the system over the 14 successful runs.

Finally, we note also that the robot achieves an average final distance from the object of slightly over one meter. Indeed, this is the farthest navigable distance with respect to the estimated object position as determined by the global step, and no attempt is performed by the robot to achieve a closer distance once the object has been identified. This is partially in contrast with the RoboCup@Home rules, which state that the robot should conclude the trial with a final distance from the object of less than a meter. This final step can be clearly implemented in our system by inducing the robot to proceed in the direction of the object's bearing until an obstacle is encountered.

C. RoboCup@Home Scenario

Finally, we performed a limited set of runs of the complete system in a scenario which is adherent to the RoboCup@Home challenge: three objects were placed in the environment, and the system was required to find them within five minutes.

HERE WE SHOW THAT IT WORKS.

VII. Discussion and Conclusions

In this paper we have presented a distributed approach for identifying and localizing generic household objects in a domestic setting. This specific problem is the focus of the "Lost & Found" challenge in the RoboCup@Home league, and presents a number of specific difficulties for current state-of-the art artificial vision technology. The domestic

In: Proc. of the IROS-08 Workshop on Network Robot Systems. Nice, France, 2008.

Run No.	Success / Fail	Time (s)	Distance (m)	Times SURFed	Rotations	B.S. Time Cam.1 (s)	B.S. Time Cam.2 (s)
1	S	127	0.6150	1	0	12.6233	10.8753
2	S	101	0.6400	1	0	12.8739	10.4294
3	F	–	–	–	–	12.3244	12.7810
4	F	–	–	–	–	12.2046	12.6368
5	S	86	0.5920	1	0	14.5597	13.1417
6	S	108	0.7817	1	0	14.4238	13.2527
7	S	87	1.2125	1	0	14.1417	15.4604
8	S	100	0.7683	1	0	13.4639	13.5876
9	S	65	2.1557	1	0	13.6263	12.2606
10	F	–	–	–	–	14.9618	10.5240
11	S	389	2.0258	3	2	11.6141	13.1281
12	S	74	1.0881	1	0	10.9771	10.6008
13	S	99	0.8655	1	0	11.4084	10.6812
14	F	–	–	–	–	14.3229	13.9257
15	F	–	–	–	–	11.1393	11.6145
16	S	81	1.0358	1	0	13.4117	13.3598
17	S	112	0.8474	2	0	11.7652	11.7760
18	F	–	–	–	–	13.4033	14.7546
19	S	63	2.2008	1	0	13.0227	13.8944
20	S	92	0.8274	1	0	13.8907	13.3724
Mean		113.1429	1.1183	1.2667	0.1429	13.0079	12.6528
S.D.		81.3831	0.5755	0.5936	0.5345	1.2112	1.5027

Fig. 9: Results for the tests on the complete operational cycle of the system for single, randomly placed and rotated objects in the test environment.

setting is typically cluttered and poorly structured unless invasive measures are implemented to facilitate object recognition. Household objects present attributes which are generally diverse: some objects are feature-rich while in others color and/or shape represent the most meaningful attributes. Throughout the development and testing of our system, we have employed a range of ordinary objects such as those shown in Figure 10. Moreover, a distinctive feature of the Lost & Found task as it is defined in the RoboCup@Home challenge is that the system must be easily taught the objects it needs to find. This requirement is particularly significant in light of future applicability of such systems in real domestic scenarios with real users.

The approach presented herein attempts to overcome the above difficulties by combining the strengths of multiple vision algorithms (color-, shape- and feature-based) along with the ability to dispose of coarse but fast global vision in combination with accurate local verification on board a robot. As shown in the experimental evaluation, the approach cannot safeguard againts false positives/negatives. However, thanks to the fact that our system first determines the color that best matches the object currently being searched before comparing its shape signature, the likelihood of false positives is greatly reduced (as not more than one object signature at a time is compared against all seen objects). The probability of obtaining false negatives is also rather small, since the system's knowledge base contains image signatures from a number of different view points. These features are demonstrated in the experiments, where in the vast majority of cases the object is found without exploring different candidates.

It is interesting to notice that the system can offer partial support for finding objects that are *similar* to known objects.

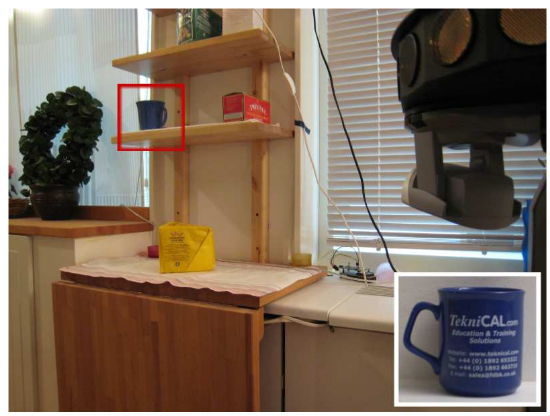


Fig. 10: An example of recognition of similar objects: a blue cup is found (shape and color matching), although it is not the specific blue cup specified by the user (feature matching fails).

In the example portrayed in Figure 10, the robot is in the process of verifying three candidates for a box of crackers, a box of tea bags and a blue cup (respectively, on the counter and on the right and left ends of the first shelf). In this case, the blue cup does not correspond to the known blue cup (in the framed inset), although the two cups are similar in color and shape. As a consequence, the local verification stage succeeds in matching color and shape, but fails due to local feature mismatch. The result is, indeed, a first step towards the possibility to find classes of objects in addition to specific objects.

Future work will evolve in two directions. First, we will continue development of the PEIS-Home towards the aim of obtaining an increased coverage area of the environment with fixed vision, thus increasing the precision of the global position estimates. This affords less reliance on the PTZ zoom feature, as well as improving the estimate of the bearing towards objects, which in turn decreases the time needed by the robot to explore the location (avoiding the need to pan, tilt and rotate). Also, more investigation can be performed on the global position estimation and perhaps the method in [6] can be fully implemented. In this context, we also intend to investigate an alternative to the "stop-and-look" shape matching algorithm currently implemented, in order to allow the robot to continue navigating past candidates which do not pass the shape criteria. All these enhancements to the system would increase the competitiveness of the system in the context of the RoboCup@Home challenge.

Lastly, we intend to explore the possibility of leveraging the partial object matching capability in the context of our current work on perceptual anchoring [7]. On account of the entire system being developed as a network of modular functionalities within the PEIS-Home (as described in Section V), the integration of the individual components of the Lost & Found feature can be employed in conjunction with knowledge representation techniques for discovering and maintaining symbolic information on household objects.

REFERENCES

[1] ActivMedia. Pioneer robot. Internet link. [online; retrieved 2008-05-11]
http://www.mobilerobots.com/research.html.

[2] H. Bay, T. Tuytelaars, and L. Van Gool. SURF: Speeded Up Robust Features. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 404–417, 2006.

[3] R. Gonzalez and R. Woods. *Digital Image Processing*. Prentice Hall, second edition, 2002.

[4] iRobot. Roomba vacuum cleaner. Internet link. [online; retrieved 2008-05-11]
http://www.irobot.com/sp.cfm?pageid=122.

[5] J. Janecek and A. Borissov. A Network Robot System for Object Identification and Localization. Master's thesis, Studies from the Department of Technology at Örebro University 33, Örebro, Sweden, June 2008.

[6] K. LeBlanc and A. Saffiotti. Cooperative Anchoring in Heterogeneous Multi-Robot Systems. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Pasadena, CA, 2008.

[7] A. Loutfi, S. Coradeschi, M. Daoutis, and J. Melchert. Using Knowledge Representation for Perceptual Anchoring in a Robotic System. In *Special issue in International Journal on Artificial Intelligence Tools. To appear (2008)*, 2008.

[8] D. Lowe. Distinctive Image Features From Scale-Invariant Keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[9] RoboCup. RoboCup. Internet link. [online; retrieved 2008-05-17]
http://robocup.org/.

[10] RoboCup@Home. RoboCup@Home. Internet link. [online; retrieved 2008-05-11]
http://www.ai.rug.nl/robocupathome/.

[11] A. Saffiotti, M. Broxvall, M. Gritti, K. LeBlanc, R. Lundh, J. Rashid, B.S. Seo, and Y.J. Cho. The PEIS-Ecology Project: Vision and Results. In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS) - to appear*, 2008.