# Multimodal object recognition module for social robots

Alejandro Cruces, Alberto Tudela, Adrián Romero-Garcés and Juan Pedro Bandera*

University of Málaga http://www.uma.es
{acruces, ajtudela, argarces, jpbandera}@uma.es

**Abstract.** Sensor fusion techniques are able to increase robustness and accuracy over data provided by isolated sensors. Fusion can be performed at a low level, creating shared data representations from multiple sensory inputs, or at a high level, checking consistency and similarity of objects provided by different sources. These last techniques are more prone to discard perceived objects due to overlapping or partial occlusions, but they are usually simpler, and more scalable. Hence, they are more adequate when data gathering is the key requirement, while safety is not compromised, computational resources may be limited and it is important to easily incorporate new sensors (e.g. monitorization in smart environments or object recognition for social robots). This paper proposes a novel perception integrator module that uses low complexity algorithms to implement fusion, tracking and forgetting mechanisms. Its main characteristics are simplicity, adaptability and scalability. The system has been integrated in a social robot and employed to achieve multimodal object and person recognition. Experimental results show the adequacy of the solution in terms of detection and recognition rates, integrability into the constrained resources of a robot, and adaptability to different sensors, detection priorities and scenarios.

## 1 Introduction

European society faces important changes in the near future. The upcoming *Silver society* [8], the energy and resources recent crisis, the increasing demand for teleworking and reducing the carbon footprint, the new Industry 4.0 and business models focused on data mining and processing, pave the way for a new social model. This model implies a strong presence of technology in daily life settings, as a tool to address many of the challenges produced by these changes, which have been recently augmented (sometimes, drastically) by the COVID-19 pandemic [14]. Smart environments able to monitor, assist and communicate with

* This work has been funded by the by the European Union's Horizon 2020 research and innovation programme under grant agreement No 825003 (DIH-HERO SUSTAIN) and by the Spanish National Research Project DEHESA RTI2018-099522-B-C41.

people are experiencing an exponential development, supported by advances in electronics, Artificial Intelligence, robotics, data mining and ubiquitous computing.

Among the devices found in these technological environments, robots may be one of the most versatile ones. The current advances in robotics point towards using these agents to offer proactivity, ubiquity and adaptability, and they have the potential to act as a natural interface between the human user-Mesh Bed Compensation and the environment [6]. However, adherence problems are a common issue for these robots working in daily life settings, specially if they cannot adapt to the particular tasks, contexts and needs and preferences of the different users [8]. Hence, it is important to provide these robotic agents, or better, the complete technological ecosystems they belong to, with the capability to robustly detect and recognize objects and people in the environment, and modify their behaviour accordingly.

While isolated sensors can be used to detect and recognize objects and people in an environment, sensor fusion techniques promotes a robustness increase on the perception process, allowing to overcome each sensor's weaknesess using the strenghts of the remainder [7]. Multimodal perception and sensor fusion are no new techniques, but in recent years this research field has received a growing attention due to the expansion of smart environments and sensor networks, such as the ones mounted in autonomous or semi-autonomous vehicles [15]. Among the numerous approaches to multi-sensor data fusion, it is possible to set the following classification: high-level fusion (HLF), low-level fusion (LLF), and mid-level fusion (MLF) [15]. LLF and MLF approaches deal with data fusion among different sensors, allowing for cooperative detection and tracking to construct shared perceptual maps, such as the one proposed in [10] to track objects in 3D space fusing optical flow, scene flow, stereo-depth, and 2D object detections, or the proposal in [9] that relies only in 2D and 3D bounding box detection to produce a more scalable fusion system. Ferreira et al. [7] also rely on low level audiovisual fusion, performed using a Bayesian framework, to drive the attention of a robot, while in [4] detected legs and faces are integrated using an Unscented Kalman Filter. Recognition is addressed in a posterior step through histogram analysis using the Bhattacharyya coefficient.

HLF approaches, on the other hand, carry out object detection or a tracking algorithm independently for each sensor, and subsequently performs fusion. They are usually easier to implement and scale, but they may discard objects with low confidence values in situations such as overlapping [15]. Among the different HLF proposed approaches, there are many based on fuzzy methods such as the one Ban et al. [2] implement to fuse speaker and face recognition. Particle filters are also frequently employed to fuse perceived objects, as in [1] where a multirobot visual tracking particle filter is proposed to track a certain object identified by its color among multiple robots. There are other approaches, finally, that employ computationally lighter approaches. Hence, Reily et al. [13] propose to control the fusion of multiple sensors via a weighted combination of observations. These

weights also control the display of observations to a human operator, providing enhanced situational awareness.

This paper presents a new HLF multimodal recognition system, based on a perception integrator module relying on confidence and pose criteria to provide a fused set of objects. The integrator uses low complexity algorithms to implement fusion, tracking and forgetting mechanisms. Its main characteristics are simplicity, adaptability and scalability. While the system can be integrated in any technological ecosystem including any kind of sensors, its HLF nature makes it a better solution for non-critical object detection (as pointed out in [15], more complex LLF or MLF approaches may be better candidates when avoiding false negatives is a must). Hence, we describe and test in the paper the application of the system to a certain use case: object and person recognition for social robots.

## 2 Software and hardware assets

### 2.1 Cognitive architecture

This paper proposes a multimodal recognition system that is integrated in the CORTEX cognitive architecture [5]. The main characteristic of CORTEX is the existence of a unified, dynamic working memory that represents information at different abstraction levels, from raw perceptual data to high-level symbols and action plans. This so-called Deep State Representation (DSR) becomes a short-term dynamic representation of the context, both internal (inner state, proprioception) and external (environmental data, objects and people in the environment). Around this DSR, a set of *agents* are deployed that interact through this shared representation to achieve different goals. This structure allows decomposing the functionality of the system into a set of software components, that can be placed anywhere in the reactive-deliberative spectrum, and that are in charge of different functionalities. The CORTEX implementation employed in this paper implements the DSR and the agents using the robocomp framework [11], while processing modules (i.e. software components) connected to these agents may be implemented in different frameworks and middlewares, such as Robot Operating System (ROS) [12].

### 2.2 Robotic platform

In this paper, the recognition system considers input data coming from different sources, all of them related to a particular Socially Assistive Robot (SAR), that integrates an instance of the CORTEX architecture. Figure 1(left) shows the robot employed in the tests performed in the paper. The base of this robot is the CLARA robot [3], developed in the CLARC EU Project (ECHORD++ FP7-ICT-601116) to perform Comprehensive Geriatric Assessment (CGA) procedures, and currently deployed in a retirement house to implement different user defined tasks [8]. This robot is already using CORTEX to: (i) navigate autonomously in dynamic daily life environments; (ii) interact with people in

constrained scenarios using speech, a tactile screen and other *ad-hoc* devices; (iii) allow healthcare professionals to set an agenda for the different use cases the robot can perform; (iv) detect people in its surroundings using two different mechanisms (LIDAR-based and vision-based) that will be detailed in Section 4.

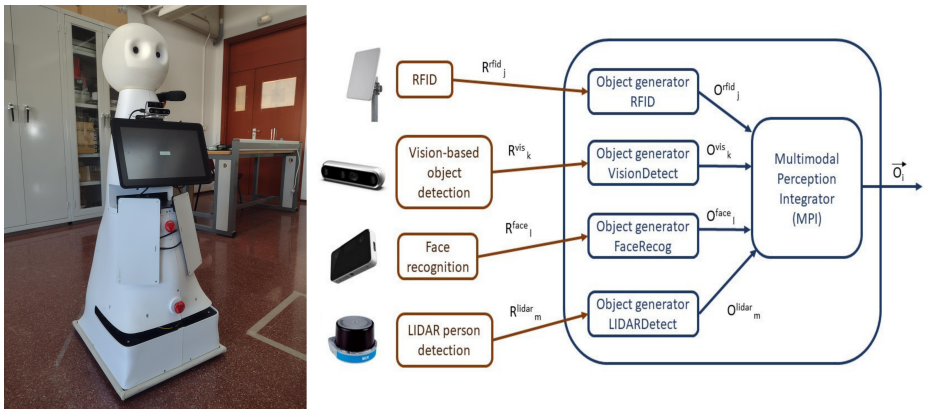


**Fig. 1.** (Left) Socially Assistive Robot employed in this paper; and (Right) Multimodal recognition system overview

The perceptual capabilities of the robot are extended in the paper including two additional modules devoted to recognition: one of them is an Intel Real Sense F455 camera, specifically designed for face recognition applications, and the other is a Keonn AdvanReader-10 RFID reader, connected to two Advantenna-p12 antennas, allowing the robot to locate RFID tags. These two both systems are also deeply detailed in Section 4.

## 3 Multimodal recognition system

Figure 1 (right) shows an overview of the proposed multimodal recognition system. All these software components are programmed as ROS nodes. The system is basically composed by a set of object generator modules, that receive data from sensors and generate detected objects ${O^m}_k$, and a Multimodal Perception Integrator (MPI) that, upon receiving any new object ${O^m}_k$ from an object generator module, processes it and produces the output of the system, $\overrightarrow{O_i}$. This output is the set of detected objects, that is sent to a robocomp agent to be inserted into the DSR. Hence, $\overrightarrow{O_i}$ is a vector containing the set of detected objects for iteration $i$, where a new iteration is performed every time any sensor produces a new detected object. Each of the objects in the vector $\overrightarrow{O_i}$, and each of the objects detected by object generator modules ${O^m}_k$, contain the fields detailed in (1).

$$O = (id, C_{id}, type, C_{type}, \overrightarrow{p}, \overrightarrow{p_{err}}, t) \tag{1}$$

where $O$ stands for a certain detected object, *id* is its identifier (i.e. an unambiguous identifier for that object, such as the passport number of a person), $C_{id}$ is the confidence value associated to *id* in the range [0...1], *type* is the type of object (e.g. "chair" or "person"), $C_{type}$ is the confidence value associated to *type* in the range [0...1], $\overrightarrow{p}$ is the pose (position and orientation) of object $O$ in the format $(x, y, z, yaw, pitch, roll)$, $\overrightarrow{p_{err}}$ codes the error of each pose data, so that the value of each pose coordinate $j$ of the object should be considered as $p(j) \pm p_{err}(j)$, and $t$ is a timestamp recording the time in which that object was detected.

It is interesting to highlight that vector $\overrightarrow{O_i}$ may contain several objects sharing the same object *type* (e.g. several "chair" objects), but it is also possible to have several objects sharing the same *id* (e.g. several detections of a certain person), due to the probabilistic nature of the data provided by the sensors. Another relevant characteristic of the selected notation is that it is possible to have unknown fields for *id* and *type* (the label *unknown* is used in these situations, and the corresponding confidence $C_{id}$ or $C_{type}$ is set to 0.0), and also for the pose data (in that case, the corresponding $\overrightarrow{p_{err}}$ coordinate will be -1).

### 3.1 Object generators

Different sensors may provide different data in different formats. In order to keep the Multimodal Perception Integrator (MPI) module independent from the particular data sources and make the system easily scalable, a set of *object generator* modules appears which process the sensor data ($R_b^a$ in Figure 1 (Right)) and converts them to a standard object representation $O_b^a$ able to be used by the MPI. These object generators check consistency of generated object data (e.g. by assigning a certain *type* to the data provided by a certain sensor), and fill empty fields if required (e.g. by inserting a timestamp in case it is not available in the sensor).

### 3.2 Multimodal Perception Integrator (MPI)

This module is in charge of producing the set of detected objects to be inserted in the DSR. As the flow diagram in Figure 2 depicts, upon receiving a new object ${O^m}_k$ from any of the active object generator modules (see Fig. 1 (Right)), the MPI triggers a fusing process that will generate a new output vector $\overrightarrow{O_i}$ by adding or fusing ${O^m}_k$ to the previous instance of the output vector $\overrightarrow{O_{i-1}}$, and deleting (*forgetting*) objects which data can no longer be considered reliable.

The fusing process will try to fuse the input object with the ones already detected, following a certain order that is determined by rearranging $\overrightarrow{O_{i-1}}$ to $\overrightarrow{O'_{i-1}}$. Rearranging is performed using a certain sorting function $f_{sort}(\overrightarrow{O}, R)$ that, given a certain input object $R$, sort objects in $\overrightarrow{O}$. Then, object ${O^m}_k$ is tested against the objects in $\overrightarrow{O'_{i-1}}$ (see Figure 2).

The selected sorting function $f_{sort}$ strongly influences the outcome of the system. For the tests conducted in this paper, $f_{sort}$ is implemented as depicted in

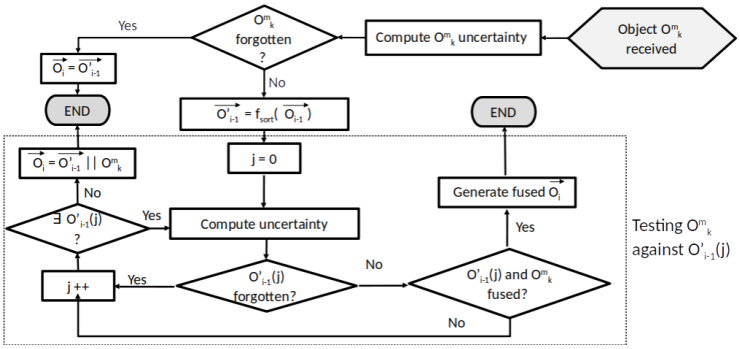


**Fig. 2.** Flow diagram of the forgetting and fusing process in the MPI

(2), where $\overrightarrow{O(t^X; i^Y)}$ stands for a subset of $\overrightarrow{O}$ including all objects with $type = X$ and $id = Y$. The values that $X$ and $Y$ indexes can have are the following: (i) $R$ when the value is equal to the corresponding value of input object $R$ (e.g. $\overrightarrow{O(t^R; i^R)}$ is the subset of $\overrightarrow{O}$ including all objects that have the same *type* and the same *id* than $R$); (ii) $U$ when the value is *unknown*; and (iii) $O$ when the value is not $R$ nor *unknown* (i.e. the object has a recognized *type* or *id* value, that differs from the one of the input object).

$$f_{sort}(\overrightarrow{O}, R) = ( \begin{array}{l} \overrightarrow{O(t^R; i^R)}||\overrightarrow{O(t^R; i^U)}||\overrightarrow{O(t^R; i^O)}|| \\ \overrightarrow{O(t^U; i^R)}||\overrightarrow{O(t^U; i^U)}||\overrightarrow{O(t^U; i^O)}|| \\ \overrightarrow{O(t^O; i^R)}||\overrightarrow{O(t^O; i^U)}||\overrightarrow{O(t^O; i^O)}) \end{array} \quad (2)$$

Equation 2 shows that the proposed implementation firstly looks for a possible fusion in the subset of objects that share the *type* and *id* of $R$. If no fusion is performed, the function looks in the subset of objects that have the same *type* than $R$, but have an *unknown id*, and so on. The use of this particular function prioritizes the fusion with *a priori* similar objects, but it is also possible to fuse objects having different *id* or even different types, to deal with noisy or imprecise detectors. It is also interesting to point out that, in the common situation in which setting an *id* for an object always implies setting a *type*, the only effect in the previous function will be that $\overrightarrow{O(t^U; i^R)} = \overrightarrow{O(t^U; i^O)} = \overrightarrow{O(t^O; i^R)} = \oslash$.

### 3.3 Forgetting obsolete objects

Pose uncertainty can grow over time, depending on the nature of the perceived object. If this uncertainty grows over a certain threshold $\theta$, the object information should be considered obsolete and the object removed from the system. Figure 2 shows the flow diagram followed by the forgetting and fusion algorithm employed in the MPI. As depicted, there are two steps in which object's obsolescence is checked. Checking is based on the time difference between two detected objects, $\Delta t$. This difference is processed by a function $f_u$ that defines how uncertainty

grows in time. The implementation of $f_u$ may depend on different factors, such as the quality of the sensor or the object type. In this paper, the algorithm has been set to a worst-case error increment: 2 meters per second for position coordinates, and $2\pi$ radians per second for orientation coordinates.

The first check is performed to the incoming object ${O^m}_k$, considering a $\Delta t$ that is the maximum time difference between the timestamp of ${O^m}_k$ and the timestamps of objects in $\overrightarrow{O_{i-1}}$. ${O^m}_k$ will be forgotten if $f_u(\Delta t) > \theta$ and the input object was perceived *before* the object in $\overrightarrow{O_{i-1}}$ which timestamp was used to compute $\Delta t$.

Objects in $\overrightarrow{O_{i-1}}$, on the other hand, will be checked for removal before they are compared against ${O^m}_k$. In this case, $\Delta t$ is computed as the time difference between that particular object and ${O^m}_k$. Again, the object will be discarded if $f_u(\Delta t) > \theta$.

### 3.4 Fusion algorithm

Providing none of them has been forgotten, in order to fuse two objects $O_1$ and $O_2$, it is firstly necessary to update the pose uncertainty ($\overrightarrow{p_{err}}$) of the first object detected. Being $\Delta t$ the difference between the timestamps of both objects, the object with an older timestamp will have its $\overrightarrow{p_{err}}$ values modified before the fusion as $p_{err}(k) = f_u(\Delta t) * p_{err}(k)$. Then, the two objects will fuse if and only if all the following criteria are met: (i) their *id* can fuse; (ii) their types can fuse; and (iii) their poses can fuse.

Algorithm 1 shows these steps for the possible *id* fusion of two objects $O_1$ and $O_2$. The confidence-based fusion procedure is designed to: (i) fuse objects that are similar (with the same *id*); (ii) fuse *unknown* objects with known objects; and (iii) fuse objects with a low recognition probability with objects more robustly detected. The *type* fusion algorithm follows the same steps as *id* fusion.

In order to understand how the complete MPI works it is important to remember that a fusion will only be performed if the pose fuse criterion, and not only the *id* and *type* fuse criteria, is met. Being $voxel_1$ and $voxel_2$ the 3D volumes in which $O_1$ and $O_2$ may be located, pose fusion is achieved as detailed by Algorithm 2.

To summarize, two certain objects will fuse if their possible poses overlap, and: (i) they are *similar* (regarding *id* and/or *type*); (ii) one of them is not classified (i.e. *id* and/or *type* are *unknown*) while the other is; (iii) they are not similar but the confidence in the classification was low for both of them.

## 4 Experimental setup: employed sensors

While the proposed system may incorporate any different data source, this section details the ones integrated in the robot used in this paper.

**RFID identification**. CLARA robot is equipped with an RFID reader and two antennas. The antennas are located at the robot's front with an horizontal angle between them of 60° (30° between each one and the robot's frontal plane),

**Algorithm 1** Confidence fusion algorithm ($id$)

**if** $id^1 \neq unknown$ **and** $id^1 = id^2$ **then**
  $id^{fused}$ set to the value of the object with higher $C_{id}$
  $C_{id}^{fused}$ set to the value of the object with higher $C_{id}$
**end if**
**if** $id_1 = unknown$ **then**
  $id^{fused}$ set to $id^2$
  $C_{id}^{fused}$ set to $C_{id}^2$
**end if**
**if** $id^1 \neq unknown$ **and** $id^1 \neq id^2$ **then**
  **if** $(C_{id}^1 + C_{id}^2) \leq 1.0$ **then**
    $id^{fused}$ set to the value of the object with higher $C_{id}$
    $C_{id}^{fused}$ set to the value of the object with higher $C_{id}$
  **else**
    no fusion
  **end if**
**end if**

**Algorithm 2** Pose fusion algorithm

**if** $voxel_1$ **overlaps** $voxel_2$ **then**
  **for** $k = 1, 2, \ldots$ **do**
    **if** $(p_{err}(k)^1 \geq 0.0)$ **and** $(p_{err}(k)^2 \geq 0.0)$ **then**
      $e = (p_{err}(k)^1 + p_{err}(k)^2)$ **or** $e =$ minimum arch containing overlapping circle sections
      $d = p(k)^2 - p(k)^1$
      $p(k)^{fused} = p(k)^1 + d * p_{err}(k)^1/e$
      $\alpha_1 = (p_{err}(k)^1 - |d * p_{err}(k)^1/e|)$
      $\alpha_2 = (p_{err}(k)^2 - |d * p_{err}(k)^2/e|)$
      $p_{err}(k)^{fused} = min(\alpha_1, \alpha_2)$
    **else**
      **if** $(p_{err}(k)^1 \geq 0.0)$ **then**
        $p(k)^{fused} = p(k)^1$
        $p_{err}(k)^{fused} = p_{err}(k)^1$
      **else**
        **if** $(p_{err}(k)^2 \geq 0.0)$ **then**
          $p(k)^{fused} = p(k)^2$
          $p_{err}(k)^{fused} = p_{err}(k)^2$
        **else**
          $p_{err}(k)^{fused} = -1$
        **end if**
      **end if**
    **end if**
  **end for**
**else**
  No fusion
**end if**

so radiation patterns of both antennas, characterized by a 90° beam width at horizontal plane, overlap. This setup allows to locate RFID tags by comparing the RSSI levels received by each antenna with the estimated attenuation due to path loss and antenna's radiation pattern, modeled as a cardioid.

**Vision-based person detection**. Our main object generator is a vision based module using an Intel RealSense RGBD camera and an Intel Neural Compute Stick (NCS2). Within this module we load a MobileNet-SSD neural network model inside the NCS2 to compute the bounding box and category of an object from the color camera. Then, depth analysis is performed to extract the bounding box surrounding the object in 3D coordinates.

**LIDAR-based person detection**. The LIDAR is the sensor that covers a wider area for the robot, with respect to the other sensors mounted. Thus, this module can detect people in ranges where the other object generator modules can not. The LIDAR detector pipeline has three steps. First, the laser scan is splitted in different segments by a Jump Distance Segmentation algorithm which divide two consecutive points in different segments if the euclidean distance between them is larger than a threshold. Secondly, a set of features are extracted from every segment. Thirdly, these features are forwarded to a boosting model previously learned. Once a segment is classified as a person, then, its pose is calculated as the centroid of the segment. Finally, this data is sent to the MPI.

**Vision-based face recognition**. The last sensor that gets into the mix is the Intel RealSense ID for facial authentication, face camera from go on. This device has the capacity to extract faceprints from the person's face in front of the robot and store it in a secure database. As the bounding box of the face and the faceprint score are the only two outputs we receive from the device, we extract the face pose by matching the face camera image with the RGBD camera image and point cloud described in section 4.

## 5 Experiments

The system has been tested in a real unconstrained environment, where people move around the robot and different objects are randomly distributed. Detection confidence values for the different sensors were empirically set according to prior evaluation of their isolated performance, and remained fixed for all performed experiments. In order to provide a quantitative evaluation of the solution, an experimental setup was prepared in which a chair is located in a fixed, known position of the map, and a person moves two meters, from one known position to another (see Figures 3(a) and (b)). Other objects (an additional chair, a table, etc.) are scattered through the environment. Figures 3(b) and (d) show the associated output of the MPI module during one of the experiments. The box drawn in the virtual map for each object measures its position uncertainty. In this sample, the person (Ale) is correctly identified thanks to the combined use of the RFID detector and the face recognition system. Once labelled, the person is still identified even when the two previous modules are not perceiving him, thanks to the leg and object detection modules keeping an updated pose of

the person. Note that the RFID and face recognition modules only can detect the person while he is into the nearest position.

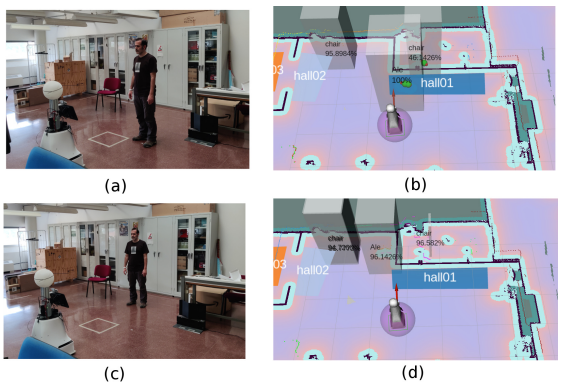


**Fig. 3.** Experiments. (a): Scene 1 perceived in t; (b) Objects provided by MPI for scene 1; (c) Scene 2 perceived in t+$\delta$; (d) Objects provided by MPI for scene 2.

Tables 1 and 2 show the results offered by the different sensors and the MPI for two objects (the person and the chair on the left) during two tests lasting for one minute each. Each table includes the number of detected *id* and *type* for each object, along with the confidence values average ($\bar{C}$), the position mean error ($\bar{e}$) and its standard deviation ($\sigma_e$) with respect to the ground-truth, and average detections per second (fps). Table 1 shows results when the person starts the test close to the robot and moves to a further position, while 2 shows the opposite case (the person moves closer). More than 4000 incoming inputs from the object generator modules where fed to the MPI, coming from different sources, in each test. These tables show how the fusing algorithm takes the best characteristics for each input module (frequency, confidence and accuracy) improving the ones collected by isolated sensors, even when only one sensor is detecting an object, thanks to the fusion mechanism. It can also be seen in Table 1 how recognized objects are tracked and kept on recognized as long as any sensor is still detecting them: in this test, the RFID recognized the person but only while he was close to the robot, at the beginning of the test. However, even when RFID no longer detected the person, the leg and object detection modules kept on doing it. These new detected objects fuse with previous ones and the *id* of the person propagates during this *de facto* tracking process. Moreover, once a sensor is able to provide a high confidence value for an object, even if that sensor no longer detects the object, that confidence value persists thanks to the followed fusion approach (see Algorithm 1). Table 2, on the other hand, shows how the confidence associated to each sensor affects obtained results. Hence, the relatively low *type* confidence included in Table 2 can be explained because initially only LIDAR (with low confidence associated) was detecting the legs of the person.

| | | faceRecog | LIDAR | objDetect | RFID | Fused |
|---|---|---|---|---|---|---|
| **Person** | $id$ ($\bar{C}_{id}$) | 4 (89.3%) | 0 | 0 | 132 (100%) | 1649 (100%) |
| | $type$ ($\bar{C}_{type}$) | 4 (89.3%) | 1304 (18.0%) | 209 (99.0%) | 132 (100%) | 1649 (100%) |
| | $\bar{e}$ | 0.12 | 0.09 | 0.18 | 0.44 | 0.08 |
| | $\sigma_e$ | 0.01 | 0.04 | 0.11 | 0.11 | 0.05 |
| | fps | 0.06 | 20.17 | 3.23 | 2.04 | 25.51 |
| **Chair** | $id$ ($\bar{C}_{id}$) | 0 | 0 | 0 | 0 | 0 |
| | $type$ ($\bar{C}_{type}$) | 0 | 0 | 437 (94.2%) | 0 | 437 (95.9%) |
| | $\bar{e}$ | | | 0.54 | | 0.54 |
| | $\sigma_e$ | | | 0.36 | | 0.46 |
| | fps | 0 | 0 | 6.76 | 0 | 6.76 |

**Table 1.** Object detection and recognition results. Person moves away from the robot

## 6 Conclusions and future work

The qualitative results show that the proposed fusion technique can overcome the limitations of individual sensors, offering improved results over isolated devices. It can also extend the results of a correct detection or recognition over time thanks to the fusing algorithm acting as a *de facto* tracking system. The solution is lightweight, able to run in autonomous agents, and easily scalable to include new sensors, mobile or fixed. Future work will firstly focus on looking for better matches in the fusing algorithm, as the current proposal just fuses the new detection with the first match. Even if the sorting function is adequate, this criterion is too constrained. More complex settings, including several people and moving devices, will also be addressed. Moreover, fusion with data provided by sensors installed in several robots and/or as part of smart environments will also be evaluated. Ablation studies considering different sensors in the fusion process will be conducted to analyze the influence of each sensor. Context information and learned cues will also be used to fed high-level knowledge to the MPI just like any other sensor (e.g. the daily agenda of a person can help filtering the possible identifications offered by perceptors). Finally, a comparison between the presented HLF approach with other MLF and LLF approaches that use DL techniques will also be addressed.

| | | faceRecog | LIDAR | objDetect | RFID | Fused |
|---|---|---|---|---|---|---|
| **Person** | $id$ ($\bar{C}_{id}$) | 2 (88.1%) | 0 | 0 | 96 (100%) | 844 (100%) |
| | $type$ ($\bar{C}_{type}$) | 2 (88.1%) | 1257 (16.8%) | 152 (97.3%) | 96 (100%) | 1507 (67.6%) |
| | $\bar{e}$ | 0.06 | 0.10 | 0.14 | 0.65 | 0.10 |
| | $\sigma_e$ | 0.01 | 0.01 | 0.04 | 0.09 | 0.03 |
| | fps | 0.03 | 18.27 | 2.21 | 1.40 | 21.91 |
| **Chair** | $id$ ($\bar{C}_{id}$) | 0 | 0 | 0 | 0 | 0 |
| | $type$ ($\bar{C}_{type}$) | 0 | 0 | 497 (94.4%) | 0 | 497 (96.3%) |
| | $\bar{e}$ | | | 0.53 | | 0.53 |
| | $\sigma_e$ | | | 0.39 | | 0.43 |
| | fps | 0 | 0 | 7.23 | 0 | 7.23 |

**Table 2.** Object detection and recognition results. Person moves closer to the robot

## References


1. Amorim, T.G.S., Souto, L.A., P. Do Nascimento, T., Saska, M.: Multi-robot sensor fusion target tracking with observation constraints. IEEE Access 9, 52557–52568 (2021)
2. Ban, K.D., Kwak, K.C., Yoon, H.S., Chung, Y.: Multimodal user identification in a intelligent robot environment. In: Proc. of the Int. Conf. on Control, Automation and Systems, 2007 (ICCAS '07). pp. 1172–1177 (11 2007)
3. Bandera, A., Bandera, J.P., Bustos, P., Calderita, L.V., Duenas, A., Fernández Rebollo, F., Fuentetaja Pizán, R., García Olaya, Á., García Polo, F.J., González Dorado, J.C., et al.: Clarc: a robotic architecture for comprehensive geriatric assessment (2016)
4. Bellotto, N., Hu, H.: Multimodal people tracking and identification for service robots. I. J. Information Acquisition 5, 209–221 (09 2008)
5. Bustos, P., Manso, L., Bandera, A., Bandera, J.P., García-Varea, I., Martínez-Gómez, J.: The cortex cognitive robotics architecture: Use cases. Cognitive Systems Research 55, 107–123 (2019)
6. Feil-Seifer, D., Mataric, M.: Defining socially assistive robotics. In: 2005 IEEE C9th Int. Conf. on Rehabilitation Robotics. pp. 465–468 (July 2005)
7. Ferreira, J.F., Castelo-Branco, M., Dias, J.: A hierarchical bayesian framework for multimodal active perception. Adaptive Behavior 20(3), 172–190 (2012)
8. Iglesias, A., Viciana-Abad, R., Pérez-Lorenzo, J., Lan Hing Ting, K., Tudela, A., Marfil, R., Dueñas, A., Bandera, J.P.: Towards long term acceptance of socially assistive robots in retirement houses: use case definition. In: Proceedings of the International Conference on Autonomous Robot Systems and Competitions (ICARSC). pp. 134–139. IEEE (April 2020)
9. Kim, A., Ošep, A., Leal-Taixé, L.: Eagermot: 3d multi-object tracking via sensor fusion. In: 2021 IEEE International Conference on Robotics and Automation (ICRA). pp. 11315–11321 (2021)
10. Luiten, J., Fischer, T., Leibe, B.: Track to reconstruct and reconstruct to track. IEEE Robotics and Automation Letters 5(2), 1803–1810 (2020)
11. Manso, L.J., Bachiller, P., Bustos, P., Núñez, P., Cintas, R., Calderita, L.V.: Robocomp: A tool-based robotics framework. In: Ando, N., Balakirsky, S., Hemker, T., Reggiani, M., von Stryk, O. (eds.) Simulation, Modeling, and Programming for Autonomous Robots. SIMPAR 2010. Lecture Notes in Artificial Intelligence. vol. 6472, pp. 251–262. Springer Berlin Heidelberg, Darmstadt, Germany (2010)
12. Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., Wheeler, R., Ng, A.Y., et al.: Ros: an open-source robot operating system. In: ICRA workshop on open source software. vol. 3, p. 5. Kobe, Japan (2009)
13. Reily, B., Reardon, C., Zhang, H.: Multi-modal sensor fusion and selection for enhanced situational awareness. In: Dennison Jr., M.S., Krum, D.M., Sanders-Reed, J.N., Arthur III, J.J. (eds.) Proc. of Virtual, Augmented, and Mixed Reality (XR) Technology for Multi-Domain Operations II. vol. 11759. SPIE, Florida, USA (april 2021)
14. Vimarlund, V., Borycki, E.M., Kushniruk, A.W., Avenberg, K.: Ambient assisted living: Identifying new challenges and needs for digital technologies and service innovation. Yearb Med Inform 30(1), 141–149 (august 2021)
15. Yeong, D.J., Velasco-Hernandez, G., Barry, J., Walsh, J.: Sensor and sensor fusion technology in autonomous vehicles: A review. Sensors 21(6) (2021)