A Distributed Perception Infrastructure for Robot Assisted Living

Stefano Ghidoni\textsuperscript{a}, Salvatore M. Anzalone\textsuperscript{b}, Matteo Munaro\textsuperscript{a}, Stefano Michieletto\textsuperscript{a}, Emanuele Menegatti\textsuperscript{a}

\textsuperscript{a} Intelligent Autonomous Systems Laboratory, University of Padova, Padova, Italy
\textsuperscript{b} Institut des Systèmes Intelligents et de Robotique, University Pierre et Marie Curie, Paris, France

Abstract

This paper presents an ambient intelligence system designed for assisted living. The system processes audio and video data acquired from multiple sensors spread in the environment to automatically detect dangerous events and generate automatic warning messages. The paper presents the distributed perception infrastructure that has been implemented by means of an open-source software middleware called NMM. Different processing nodes have been developed which can cooperate to extract high level information about the environment. Examples of implemented nodes running algorithms for people detection or face recognition are presented. Experiments on novel algorithms for people fall detection and sound classification and localization are discussed. Eventually, we present successful experiments in two test bed scenarios.

Keywords: Ambient intelligence, Intelligent autonomous systems, Assisted living, Camera network, Distributed sensing, Autonomous robots.

1 Introduction

Distributing intelligence among agents and objects that share the same environment is the main idea of Ambient Intelligence (AmI) \cite{1, 2, 3}. Such systems are based on sensors and actuators spread across the environment to capture relevant information and to act on the environment accordingly.

Research in this area focused on the extraction of relevant information from the raw data streams collected by sensors \cite{4, 5, 6} and on user profiling \cite{7} (i.e. the intelligent environment can adapt its behaviors according to the people that are inside it). In recent years, particular attention has been given to ambient intelligence applications aimed at helping elderly people living alone and maintaining their independence, while assuring a high level of assistance when needed \cite{8, 9}.

In this paper, an ambient intelligence system meant for assisted living and surveillance applications is described. Our system is composed of several audio and video sensors spread in the environment that let the system process images and sounds from multiple sources, and acquire also panoramic views of the room. The system presented in this paper offers two main advantages over the works presented in the literature: first, it includes both audio and video event detection, that is identified as being highly beneficial also by previous works \cite{10}. Second, unlike most of the works in this field \cite{11}, it does not rely on accelerometers or any other technology that requires the cooperation from the people living in the intelligent environment. This is the first time a “passive” system handling both audio and video data is presented in the literature.

The capability of an AmI system to work without needing wearable sensors is particularly important when assisting elderly people \cite{12}. In fact, it is well known from clinical studies \cite{12} and by companies in the business of assisted and monitored living for elderly people that any system based on the cooperation of the monitored persons is doomed to fail, even if this “cooperation” means only to wear a small intelligent device (e.g. a bracelet with an accelerometer or a more intrusive necklace with an emergency button on it). Therefore, our system is designed to be non intrusive and to work with non-cooperating people.

An additional feature of our system is the use of the innovative Omnidome\textsuperscript{®} \cite{13} dual-camera sensor together with standard perspective and Pan-
Tilt-Zoom (PTZ) cameras. The Omnidome sensor is particularly suited for video-surveillance applications, as it couples the 360° panoramic view of an omnidirectional camera with the high resolution offered by a PTZ unit; the cross-calibration of the two sensors represents a strong plus, that makes it possible to consider it as a single, high-resolution omnidirectional sensor. This paper presents the first experiments employing this sensor in an assisted living application, and shows which are the benefits in such context.

The system focuses on the detection of audio and video salient events in order to guarantee a safe and secure environment for elderly people living at home. Several papers in the literature describe which are the main tasks to be accomplished for assisting elderly people living at home [1, 14]. The scenario outlined in these papers is rather complex, because the events to be detected can belong to the audio, video or inertial sensory field. Systems for assisted living are therefore equipped with very heterogeneous sensors, in order to gather a high amount of data about the environment, and better knowledge about the context. The ultimate goal of assisted living is to provide a set of user-centered services [1, 15].

The structure of an assisted living system is usually based on a set of applications working at the same time [16, 17]. Such applications are quite independent, because they analyze data provided by diverse sensors, and aim at detecting very different events: for this reason a data fusion stage is often missing, since observations made by different sensors are unrelated.

Our system aims at detecting high level knowledge about the environment, and exploits several processing algorithms: some of them, like fall detection, are specific for assisted living, and were expressly developed for this system, while others (like face detection) are more general, therefore state-of-the-art solutions have been employed. A Pioneer P3-AT robot is also part of the system to extend the range of functionalities of a typical AmI setup: the integration of a mobile platform represents a strong advantage, since it enables the system to become active on the environment [18].

A key advantage of the system we propose is that it is built upon a framework that handles communications among different nodes that process the video streams acquired by the cameras. In other words, we exploited an existing framework meant for multi-media applications to implement the distributed architectures described in the literature [19]. This is the only way to ensure fast realization and scalability of this kind of systems that are spread across the environment. Since the system is multi-modal i.e. it analyzes audio and video data, we chose a multi-media framework called NMM (Network-integrated Multimedia Middleware) [19].

2 System infrastructure

Our system is part of the “Safe Home” project, and was designed to provide increased safety to elderly people living at home. This includes a wide variety of services:

- a home security system,
- a home safety system,
- a domotic system for automating doors, windows and air conditioning,
- a video communication framework for connecting the assisted person with friends and relatives,
- a telemedicine system,
- a data collection and recording infrastructure,
- a front-end for handling the interface from and to external systems (e.g. for controlling the domotic system from outside the assisted living environment).

Regarding the first two points, it should be noted that security means the protection of the assisted people against possible intrusions from outside (e.g. a thief), while safety refers to the protection of the people against unwanted dangers caused inside the apartment (e.g. a fall).

The whole structure of the Safe Home system is shown in Figure 1. As it can be seen, the system includes sensors, actuators and an interface to a remote control room that is used by human operators to react to the alarms issued by the system. Such system has been developed in the context of a cooperation project among university and industry. Our task was to develop computer vision and audio processing algorithms to assist elderly people: we therefore decided to focus on fall detection, that is the most dangerous event in this context [12], and on sound classification, focused on events related to security.
As in the case of several AmI systems proposed in the literature [16], our system relies on multiple sensors (like microphones and cameras). A distributed processing is therefore desirable for scalability purposes. In our approach, each sensor is connected to a separate computer that processes its data stream to extract high-level information. This model lets the system communicate effectively, because only compact high-level information is exchanged, while heavy data streams are processed locally. A centralized approach would impose strong limitations on the total number of sensors, because only few of them can be connected to the same computer, and cables have a maximum length in the orders of a few meters.

To realize the distributed perception infrastructure described above, we exploited the open-source Network-integrated Multimedia Middleware (NMM) by Motama [19], that provides communication facilities over the network. In NMM, the basic processing unit is a node, and every processing algorithm should be placed inside a node in order to interact with other entities in the middleware.

Nodes are organized by means of graphs, that describe how nodes are connected and exchange data. One of the main advantages of using NMM is that a single graph can connect nodes that reside in different machines on the same network, without any overhead for handling the communication over the network. A complex perception system can be synthesized by implementing the system functionalities inside nodes in the appropriate machines, and connecting them using a distributed graph.

NMM is not designed to control active devices, thus for controlling mobile robots that can act in the environment a different system has to be used. The control of the mobile robot in the environment is achieved through the Robot Operating System (ROS) framework [20], that provides an easy access to both high level behavior and low level control of the robot. We created a communication link between NMM and ROS to let the robot and the camera network cooperate effectively. In the following sections, we present the different processing modules and how we have implemented the cooperation among them, in order to achieve the desired system functionalities.

**An example: the click & move functionality**

To show the typical structure of a NMM graph and the set of processing nodes involved, the simple “click & move” application is considered. This was
developed for sending commands from the camera network to the robot. It presents a graphical user interface in which all video flows collected by the different cameras are shown (like in Figure 9): the user can click on any view, and the application will mark the same point on the other views, sending the robot to that location.

The click & move application can be run on any combination of sensors, and a possible NMM graph is shown in Figure 2. In this example, four cameras available in the test room are involved:

- “door”, installed over the room door;
- “PTZ”, a PTZ camera fixed to the ceiling roughly at the center of the room;
- “ceiling”, a perspective camera placed so that its optical axis is perpendicular to the ground and, again, installed roughly at the center of the room;
- “omnicam”, an omnidirectional camera attached to the wall, at the center of one side of the room.

A set of nodes are activated on each machine connected to a sensor. The distributed system structure can be observed: the four blue columns in Figure 2 represent four different machines that are connected to the network. Each machine is directly connected to a sensor, which is managed by a set of NMM nodes that handle acquisition, processing and data transmission to other machines over the network. All MouseDetectionNodes send their output to one VideoEventReceiverNode; this last node manages the interface of the application. The nodes working on the stream generated by the door camera, for example, are the following:

- TVCardReadNode2: manages the acquisition from an analogic source;
- YUVtoRGBConverterNode: converts the image format from YUV, provided by the acquisition node, to RGB;
- VideoScalingNode: rescales the video;
- VideoUndistortingNode: undistorts the images in the video, according to the results of the intrinsic parameter calibration;
- MouseDetectionNode: catches the mouse events on the window.

Figure 2: Graph of the click & move application. The distributed structure is visible, since all nodes (white dashed boxes) but the VideoEventReceiverNode are distributed among the different machines (blue dashed boxes) directly connected to the video sources.

All the nodes in the structure discussed above are included in NMM basic functionalities but the VideoUndistortingNode and the MouseDetectionNode, that were expressly developed for this project.

3 Distributed perception modules

In our system, cameras of different types and microphones cooperate to recognize events using images and sounds. The system structure is shown in Figure 3, where it is possible to identify a block for multimodal sensor integration that collects the output of several sensing blocks. This is not a sensor fusion module, because data provided by the different sensing systems are unrelated, as explained in section 1. In the sensor integration module two activities take place: on one side, observations made by computer vision modules from different viewpoints are related to each other: for example, if an event is detected by one camera, but needs further investigation, the sensor integration module can provide information about sensor geometry to help further sensors focus on the same portion of the environment. The second task performed in this module is the creation of links between events coming from different sensory fields. For example, a fall and a scream can be related: if they occur
with short difference in time, they can be linked to each other.

In the left part of the scheme in Figure 3, the video processing is shown: images acquired by the cameras are processed using motion detection and fall detection, which is based on the motion patterns found in the first block. The fall detector provides a high-level output that is suitable for sensor integration. Sound is processed in two different blocks: one is dedicated to sound localization, while the other analyzes the sound content exploiting a feature extractor and a classifier. The sensor integration block is meant to collect the high-level data provided by all sensing blocks, fuse them together, and trigger further analyzes of the environment, as it will be explained in section 3.1.

3.1 Distributed pipeline and cameras cooperation

Some of the events detected by the system can involve one or more of its sensing modalities: for instance, a scream, detected by the sound subsystem, could be caused by a person falling. On the other hand, a scream and a fall can occur at the same time, but happening to different people at different positions. To understand whether two events are related or not, cooperation algorithms between audio and video sensors have been developed, as shown in Figure 3. Such cooperation is based on spatial information, that is available for both modalities (audio and video). When events occur close in time, the system verifies whether they occurred in the same location, and if they are compatible: for example, the video event “person falling” is compatible with the audio events “scream” and “broken glass”, but not with “barking dog” (see paragraph for the discussion about audio and video events that are detected by the system). This association mechanism provides a more detailed description of the events that aims at increasing the confidence of the detected events.

Cooperation between audio and video is also used to further investigate events: if only a single modality is able to detect an event, it can trigger the other to focus on it. For example, the localization by the audio processing system of a broken glass or of a barking dog can trigger and guide a PTZ camera towards the sound source. Such kind of geometrical information is particularly useful while dealing with PTZ cameras: they are a natural choice for acquiring highly detailed views of the environment, but it is impossible to get an overall understanding of the environment using their images. Cooperation can overcome this problem by letting PTZ cameras be controlled by other modules: in this case the geometrical information of all the sensors present in the environment is used to point the PTZ cameras to the direction of the perceived event, focusing on it while keeping under control the other parts of the environment.

The camera network is composed of PTZ, static and omnidirectional cameras that cooperate. The system exploits these sensors in different ways in order to achieve the maximum coverage and performance. For example, omnidirectional cameras offer low resolution, but extremely wide images, that are suitable for keeping under control large spaces. On the other hand, only PTZ cameras can acquire high resolution images that can be used for face recognition. The fields of view of the sensors should overlap, since a certain level of redundancy can be useful for overcoming the problem of occlusions and to increase performance. As an example of this, we will present a distributed people recognition approach.

Most people detection and face recognition algorithms work better when observing a person from a frontal view, but they drop in performance when the person or the face is seen from the side.

In the following, all system modules will be described in detail.

3.2 People detection

Recognizing people is one of the basic capabilities of an ambient intelligence system, because many events that should be observed, especially for elderly care, are related to human activities. In Figure 4 we propose a pipeline suitable for perform-
ing distributed face recognition in an heterogeneous camera network composed of fixed and PTZ cameras and a central server. Fixed cameras are used for running the people detection function based on a simple Viola-Jones algorithm trained with full bodies (obviously more modern and better performing approaches to people detection could be used, like [21, 22, 23], but the focus of this work is on distributed perception, and the performance of the Viola-Jones algorithm were satisfactory in the current scenario).

A people detection module runs on each fixed camera of the system. Once people are detected, an important security task is to recognize their identities, which is done by observing their faces; images provided by fixed cameras, however, have too low resolution to perform an accurate face recognition. To overcome this issue the coordinates of the detected people are sent over the network to the nodes connected to the PTZ camera, that is able to zoom on the faces to be recognized.

3.3 Distributed people recognition

Since the location of each person has already been measured by other cameras, the expected position for the face to be detected is then calculated to drive the PTZ unit to acquire a close-up image of the face. The coordinates of the person position on the ground plane are estimated by intersecting the 3D ground plane equation computed in the calibration phase with the ray passing by the camera optical center and the lowest point of the person in the image. The 3D position of the face is estimated as the intersection of two 3D lines: one connects the image face position and the camera optical center, and the second is perpendicular to the ground plane and passes by the person 3D position. If a PTZ camera receives the person position only, it zooms at a standard height of 170 cm from the ground plane. A face detection algorithm is exploited to segment the face from the image; the selected region of interest is then sent over the network to the face recognition node, that stores the face database and runs the face recognition algorithm in the next section. Only one PTZ unit is employed in our case, but multiple units could be added and exploit the same server.

Faces found by the Viola-Jones detector are processed to calculate a vector of features that best describe them, using the technique called Eigenfaces [24]. A small set of sample pictures is used to generate a space capable to better encode the variations between faces, called Eigenspace. In this project, this Eigenspace has been built using the samples of the free face database “YaleFaces” [25]; a generalized features space for frontal faces has been obtained. As a final step, features are fed into a SVM properly trained to classify frontal faces of human users, giving for them a face claim identifier. A statistical refinement of the output can be performed in this stage, forcing the system to return as a result of the current face identity the mode of the last 5 identities claimed.

3.4 Fall detection

One event that deserves special care with elderly people is the fall: this is a dangerous situation, because after a fall people could be unable to move, even for pushing an alarm button. Several techniques based on diverse sensors are described in the literature [10], but two are the main approaches to the problem: computer vision and wearable inertial sensors. The former choice turns out to be more expensive, but the latter requires the assisted people to cooperate and always wear the sensors, which is never the case, because inertial sensors are felt as intrusive by the elderly or they just forget to wear them. For this reason, computer vision is the most favorable choice for assisting the elderly [10].

The topic of fall detection using computer vision has already been addressed: the main approach is based on motion analysis, since a fall is a movement that generates a specific motion pattern, as pointed out in [26, 27]. Fall detectors can be based on either motion analysis of the whole body, or fo-
crcing on some parts only, as it is the case of the head in [28], or, finally, exploiting features like gradients [29]. Fall detectors operating on omnidirectional images [30] and range data [31] have also been described. In all cases, the aim of the system is to distinguish between a fall and a normal motion, like sitting on a chair or on a sofa, or going to bed. For this reason, motion direction and speed should be analyzed.

The developed fall detection module is aimed at detecting the particular motion cues of a falling person. The module exploits a mixed technique relying on both background subtraction and frame differencing. Such combination is used to compensate the drawbacks they both have; in particular, frame differencing turns out to be very efficient when coping with illumination changes, as discussed in [13]. This is a very important feature to be considered in real-world scenarios, and a strong advantage over other methods like [27], whose performance has been assessed on sequences with a flat background.

Unlike the algorithms presented in the literature, our fall detector can be executed indifferently on images provided by both perspective cameras or omnidirectional ones, after the unwarping process. Even though an unwarped image is in theory similar to a perspective image, in practice a strong difference is represented by the non-uniformity of the image resolution. The upper part of the unwarped image has a much higher resolution than the lower part, because the former comes from the rectification of the outer circle in the omnidirectional image. The non-uniform resolution causes a certain degree of deformation that may invalidate some assumptions that can normally be made in fall detection systems [27]. On the other hand, algorithms focused on omnidirectional images [30] are not general, and cannot be applied on all video sources of the system, since just one omnidirectional camera is present in the system.

To keep our approach general and obtain a good performance on unwarped omnidirectional images, we developed a fall detection algorithm that considers all blobs found by the motion detector, and performs a further processing based on three factors: aspect ratio, gradient and movements during and after a possible fall. The first parameter, aspect ratio, is the ratio $W/H$ between width and height of the bounding box containing a person. This parameter is very commonly used in people detection and tracking, but for detecting falls it is used in a slightly different way here: it is tracked over time in order to detect possible changes in orientation. The change in the bounding box aspect ratio is measured by means of the $\beta$-parameter, evaluated as:

$$\beta = \sqrt{(W - W')^2 - (H - H')^2},$$  \hspace{1cm} (1)

where $W$ and $H$ are the bounding box width and height at k-th frame, while $W'$ and $H'$ have the same meaning referred to frame k-1.

The second parameter considered by the fall detector is the gradient. For performing gradient analysis, both horizontal and vertical Sobel edge detectors are exploited for evaluating corresponding edge images $I_x$ and $I_y$. Two features, called horizontal ($G_x$) and vertical ($G_y$) gradients are then evaluated by summing all pixel values inside the considered motion blob in the two images:

$$G_{\text{dir}} = \sum_{i=0}^{W} \sum_{j=0}^{H} I_{\text{dir}}(i, j),$$  \hspace{1cm} (2)

where dir indicates the direction (either $x$ or $y$). This could seem similar to aspect ratio, but it represents a different approach, because gradient analysis considers the content of the bounding box, neglecting the ratio between width and height: an object can present a strong horizontal gradient even though it is vertically placed, and vice-versa. Horizontal and vertical gradients provide information because people that are walking or standing have a vertical component that is sensibly stronger with respect to the horizontal one, while fallen people have their main component in the horizontal direction. The fall detector then compares the two gradients with their evolution over time, for understanding if the moving blob corresponds to a person that is falling.

The third factor considered by the fall detection algorithm is the movement during and after the fall. As mentioned above, moving objects in the scene are detected using a mixed technique based on background subtraction with active background and frame differencing [13]. The output of this algorithm is used to validate those bounding boxes whose $\beta$-parameter and $G_{\text{dir}}$ follow an evolution that is compatible with a fall, which happens when they are both above a threshold for a number of consecutive frames corresponding to 1s. When such pattern is found, the system verifies that no motion (or limited motion) is found inside the bounding box, in order to signal a fall has happened.
Figure 5: Examples of fall detection: in (a) the green bounding box indicates a tracked person that did not experience any fall. Fallen people have been detected in (b) and (c), and marked by a red bounding box.

Some results of the fall detection algorithm are presented in Figure 5 in (a) a person is surrounded by a green bounding box, indicating that no fall occurred. In (b) and (c) two images in which falls have been correctly detected (indicated by the red bounding box) are shown. These results have been obtained on unwarped omnidirectional images, which represent a difficult test for the algorithm: it needs to process images with non uniform resolution, as mentioned above, and bounding boxes of limited size (as it can be seen in the images).

3.5 Sound identification and localization

Sound information is important to reveal dangerous events. The classification of audio events adds information to detections made on video data. Moreover, when objects are occluded or the illumination is too low, sound could be the only way to localize an event.

3.5.1 Sound identification

Focusing on the problem of sound identification inside an intelligent environment, some assumptions can be made: first of all a closed set of “dangerous” sounds is identified, in our case: screams, shots, bangs and breaking glasses. Moreover, the system needs to work in real-time. According to these constraints, a novel sound classification system has been developed and implemented in a NMM node. The node performs a short time analysis of the perceived sound, whose spectrum is processed to extract relevant features that can be used as a “signature” of the sound class. Mel-Frequency Cepstral Coefficients (MFCC) are used as fast, real-time and reliable sound features.

During the feature extraction step, short frames of 30 ms of the audio signal are captured from the microphone, collected and filtered through a raised cosine Hamming window to prevent aliasing. Each window is overlapped with the following one by 10 ms. The frequency spectrum is then calculated, applying the FFT algorithm. Finally, MFCC are calculated: the module in decibel of the frequency spectrum is multiplied by a Mel filter bank and then, finally, the Discrete Cosine Transform is applied. The Mel filter bank is a set of band-pass filters localized according to the Mel scale, an auditory scale capable of modeling the response of the human ear. The Discrete Cosine Transform is applied to the results of the Mel filter bank, to obtain the Mel-Cepstral vectors, that represent the short-term power spectrum of a sound, that can in turn be used as a feature of the current sound.

Features are classified in a second stage: during an offline training step, cepstral vectors are col-
lected and used to create a model for each sound; in the identification phase the cepstral vectors are compared to these models in order to find an identity claim. The classification process of cepstral vectors is carried out by a Support Vector Machine.

In order to prevent fluctuations and to obtain more stable performances, a final refinement stage has been applied: through a statistical analysis of the last 50 identity claims, the most frequent identity is claimed as the identity of the current sound perceived.

3.5.2 Sound localization

Standard methodologies use two or more microphones spread over the environment in order to estimate the time delay of arrival of the sound wave. According to this delay and to the geometrical position of the microphones, it is possible to accurately calculate not only the direction but also the three-dimensional position of the sound source. Commonly used techniques to calculate the time delay are based on the Generalized Cross-Correlation (GCC) function between couples of signals. In particular, in order to take advantage of redundant information provided by multiple sensors, the Multichannel Cross-Correlation Coefficient (MCCC) method can be a good choice to calculate the sound source position [33].

Sound source localization has been implemented in a NMM node connected to two microphones placed in the environment. Each couple of microphones will be able to perceive a single sound source. Knowing the distance between the two microphones we estimate the position of the audio source through the estimation of the time delay between the waveforms perceived by each microphone (TDOA):

$$\theta = \arcsin \left( \frac{\tau c}{d} \right),$$

(3)

where $\theta$ is the incoming direction of the sound, $d$ is the inter-aural distance, $c$ is the sound speed and $\tau$ is the TDOA. The TDOA is evaluated using the Generalized Cross-Correlation. The GCC in the frequency domain is defined as:

$$t_{GCC}_{X_1X_2}[p] = \sum_{f=0}^{N-1} \Psi[f]S_{X_1X_2}[f]e^{j2\pi pf},$$

(4)

where $N$ is the number of samples, $\Psi[f]$ is the frequency domain weighting function. The cross-spectrum of the two input signals is defined as:

$$S_{X_1X_2}(f) = E\{X_1(f)X_2^*(f)\},$$

(5)

were $X_1(f)$ and $X_2(f)$ are the Discrete Fourier Transform of the input signals and $(\cdot)^*$ denotes the complex conjugate. GCC shows a peak in correspondence of the time delay, and minimizes the influences of uncorrelated noise. In order to exploit the phase information during the computation of the GCC, the PHAT weighting function (Phase Transform Filter) has been used to normalize the amplitude of the spectral density of the two signals:

$$\Psi_{PHAT}[f] = \frac{1}{|S_{X_1X_2}[f]|}.$$  

(6)

The GCC can show consistent information when the sound signal changes over time without any pseudo-periodic feature; otherwise the performance is sensibly reduced.

4 Robot control

Elderly assistance sometimes requires small interventions by an external agent. Service robotics could be the solution in situations in which the human presence is not necessary, but an active device can further investigate or even solve dangerous events. A teleoperated robot can be guided by a human to a specific place in the intelligent environment in order to look at elderly conditions, help in a specific task, or simply give a better view of a scene through a mobile camera.

Our system integrates a tool to manage the movement of a mobile robotic platform inside the...
environment monitored by the camera network. The robot we use (Figure 6) is a Pioneer P3-AT equipped with several sensors. A Microsoft Kinect provides RGB-D data with 640×480 pixel resolution at 30 frames per second. A SICK LMS100 Laser Range Finder mounted on a pan-tilt unit scan an angle of 270° in the environment at a maximum frequency of 50 Hz. Front and rear sonars sense obstacles from 15 cm to 7 m paying particular attention to transparent or moving objects. All these sensors are interfaced with the robot by using ROS. The Robot Operating System (ROS) [20] by Willow Garage is an open-source, meta-operating system providing services like hardware abstraction, low-level device control, message-passing between processes, and package management. ROS contains tools and libraries useful for robotics applications, such as navigation, motion planning and image and 3D data processing. The system runs on a laptop equipped with an integrated Wi-Fi wireless LAN adapter in order to communicate with the whole intelligent environment. Among the drivers provided with ROS, the ones that are used in this system are:

- OpenNI driver that allows to interface with common RGB-D devices;
- $p2os$ driver that provides the controller for P3-AT mobile platform and sonar;
- $LMS1xx$ driver that obtains raw data from several types of Laser Range Finders.

The use of ROS gives us the possibility of achieving a full integration between robot and sensors.

Thanks to some open-source software, we could integrate a lot of useful functionalities to our robot:

- $gmapping$ [34] or $vslam$ packages implement methods of localization and mapping using depth information;
- $navigation$ [35] package processes information from odometry and sensor streams and provides velocity commands to be sent to the mobile robot;
- $peopleTracking$ [36] package is multi-people tracking and following algorithm based on RGB-D data, specifically designed for mobile platforms;

In the intelligent environment two different systems have to coexist: NMM, to manage the communication between different video and audio devices, and ROS, to make the robot able to safely move in an indoor environment. The cooperation between these systems is necessary to let the robot properly react to dangerous events.

As we said before, the communication between the robot and the intelligent environment is realized by message passing between NMM and ROS. The interface consists of a wrapper which takes information from NMM nodes and republishes it using the proper topic in the ROS framework. The only data to be exported consists of the action the robot has to accomplish and the position it has to reach in 3D coordinates. Robot actions are decided by the sensing system based on the events detected in the environment; commands coming from the sensory system are then converted and sent to the ROS framework. It is important to note that all issues related to robot navigation and sensing are solved inside the ROS environment, which is best suited for this task.

5 Experimental results

Tests of the system have been performed in two different intelligent environments. The first one was set up in our laboratory, the second one was installed in a real domotic apartment designed for elderly people. Tests conducted in the laboratory of Figure 7(a) used a PTZ camera, an Omnidome camera [13], two perspective cameras, two microphones and a Pioneer robot. More extensive tests have been performed also in the assisted living apartment for elderly people, set up jointly with industrial partners in a complex infrastructure, in the context of the project “Safe Home”. Industrial partners provided several domotic nodes to control shutters, air conditioning, lights, door and windows. A sketch of the system is shown in Figure 7(b): two perspective Axis M1103 IP cameras have been placed in the kitchen and in the bedroom. In the Safe Home environment the system has been tested during 10 days of uninterrupted running.

To show how the intelligent environment works, consider the test bed of Figure 7(a). When a human enters the room, the omnidirectional and perspective cameras perceive him and the tracking starts. Each detected person is associated to a numerical identifier and to a bounding box, and the track of each person is kept until the person leaves the room.
The system also monitors the bounding box of each track, and considers the variation on the bounding box dimension in order to detect falls, as described in section 3.4. If a fall occurs, the PTZ camera will receive a command to point at the direction of the falling person to get a more detailed image of the dangerous event – this is achieved by means of the intrinsic calibration of the two cameras constituting the Omnidome sensor, rather than exploiting the sensor integration module. At the same time, a warning message is issued to a human operator in a remote control room, who can simultaneously monitor a high number of smart environments.

The behavior of the system in response to auditory stimuli is similar. The sound detection system isolates sounds from silence; then, if a sound has been revealed, it is identified to recognize dangerous events, such as a scream, a shoot or a broken glass, beside natural human voice.

When events are detected by the system, a human operator has the possibility to control the robot and drive it to verify what happened in the environment.

5.1 Multi camera calibration and robot action

The cameras placed in the environment need to be calibrated to cooperate effectively. Thus, observations made from each viewpoint can be referred to a common reference system. The common reference has been placed at one corner of the room and all cameras have been calibrated with respect to that reference. Two NMM nodes for each camera transform coordinates in the image reference from and to the world coordinates.

The standard calibration process [37] implemented in the OpenCV library\footnote{\url{http://www.opencv.org}} has been exploited. The process is based on multiple observations of a checkerboard for inferring the intrinsic parameters, and on the observation of the same object in a known position for extrinsic parameters. The omnidirectional camera, which is part of the Omnidome sensor\footnote{\url{http://www.opencv.org}}, was calibrated using the OCamCalib toolbox\footnote{\url{http://www.opencv.org}}, which is again based on the observation of a checkerboard.

The calibration accuracy that was achieved is sufficient for the modules of the system, that do not suffer from errors in the order of 20-30 mm. Simplicity was preferred to more complex calibration techniques because it would be unfeasible to require an accurate calibration when the system will be installed in real-world scenarios.

Calibration average errors when projecting image points to the 3D ground were measured for all cameras. They are 3.96 cm for the PTZ camera, 7.7 cm for the door camera, 2.49 cm for the central camera and 4.37 cm for the omnidirectional camera. Since the uncertainties related to the detection algorithms composing the system are in the same order of magnitude, a more accurate calibration would not provide significant benefits. This calibration procedure that aims at simplicity rather than precision is therefore suitable for the system being considered.
Integration with the robot has also been tested by means of a simple application, similar to the click & move described in section 2. At http://youtu.be/gYqBdVLBqPQ, a video shows how just by clicking in one of the image of the distributed camera network, the operator can move the robot or the PTZ camera. Tests involved both communication and calibration, since the robot shares the same reference system used by the camera network. The uncertainties related to robot positioning are comparable with reprojection errors.

5.2 People detection and recognition evaluation

For testing the pipeline described in Figure 4, we used a fixed and a PTZ camera placed as reported in Figure 7(a). At http://www.youtube.com/watch?v=j7SV0h18wOs, a video shows our system at work: two people enter the room and are detected by the fixed camera. Then, their positions are sent over the network and read by the PTZ camera, which takes a close-up of one person at a time, finds the face and sends the corresponding part of the image to the network again. Finally, the recognition server analyzes the face image and recognizes the person. Some screenshots of this video are reported in Figure 10.

The face recognition training set was composed of videos containing the faces of three people acquired in different positions, orientations and lighting conditions. In Figure 11, a graphic representation of the two first PCA components of the examples contained in the training set is shown. Different colors highlight points belonging to different people. As it can be noticed, people faces are well separated in this space. In Table 1, execution times are reported for every part of the algorithm. The most computationally expensive part is given by the face detection performed by the PTZ camera while zooming.

The performance of the system has also been evaluated by varying the number of subjects that should be recognized during each experimental session. As shown in Figure 11, in case of a small number of identities to be distinguished, the per-
Figure 10: Screenshots from the video showing our complete system at work. Colored arrows show the information flow through the network components.

Table 1: Computational times for every algorithm (ms).

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Time (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>People detection</td>
<td>83.3</td>
</tr>
<tr>
<td>Face detection in a whole image</td>
<td>181.8</td>
</tr>
<tr>
<td>Face detection in a person bbox</td>
<td>9.7</td>
</tr>
<tr>
<td>Face recognition</td>
<td>0.1</td>
</tr>
</tbody>
</table>

Performance of the system is very good; but while incrementing the number of subjects, the discrimination becomes more difficult. In order to avoid this problem it is possible to increase the dimension of the feature vectors extracted by the eigenfaces algorithm: in this way it will be possible to rely on a more detailed descriptor of each face. Figure 11 shows also a comparison of the results obtained while using different dimensions of the feature vectors. Recognition problems might also be overcome by using more adaptive techniques of clustering instead of SVM or by using a more reliable multimodal approach, as using voice claims.

5.3 Audio and video events evaluation

Some experiments focused on situations with just one person in the scene, while other were performed with several people. Event-based performance has been measured: test sequences have been divided into events, that are a set of consecutive frames in which one or more people are either walking or falling. In 21 events a person falls, while in 35 no falls happen. The algorithm was tested on sequences acquired using both perspective and omnidirectional cameras, and the performance level is the same in both cases.

For measuring performance, true positives (TP) and negatives (TN) have been measured, together with false positives (FP) and negatives (FN). TP is the number of sequences in which the event (i.e. fall in this case) occurs, and is correctly detected by the system, while TN is the number of sequences in which the event does not occur, and is correctly not detected by the system. Misclassifications are
Figure 11: First two PCA components of the faces used as training set. Different colors mean different people (a). Performance of the people identification by varying the number of faces to be recognized, and among different lengths of the features vector (b).

Table 2: Types of falls considered for testing the system, and performance indicators.

<table>
<thead>
<tr>
<th>Fall type</th>
<th>TP</th>
<th>FP</th>
<th>FN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Front fall</td>
<td>8</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>Lateral fall</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Backward fall</td>
<td>3</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Crowded environment</td>
<td>6</td>
<td>3</td>
<td>1</td>
</tr>
</tbody>
</table>

measured by FP and FN: FP is the number of occurrences in which the event does not occur, but is detected by the system, while FN is the vice-versa.

Several different types of falls exist, that generate different motion cues. Test sequences considered a variety of situations, that are grouped in Table 2 together with the performance indicators mentioned above. Additional sequences not containing any fall were also used.

All the above statistics are event-based: videos exploited for performance measurements have been divided into sub-sequences, each one representing a single event to be recognized. During the experiments, 90% true positives and 80% true negatives were measured, leading to an overall accuracy of 85%, and a precision of 90% (see Table 3). Errors were mainly due to instabilities in the patterns found by the motion detector, and can therefore be corrected by smoothing such motion information.

The sound detection system has been trained using a total of 60 seconds pre-recorded sound data; samples belonged to four classes: screams, bangs, broken glass, dog barking, that are the sounds the system is able to recognize. A further “background” class of human voices in unharmful situations is also used. Experiments on the sound recognition system reported a performance of 83% of true positives and 94% true negatives with 16% of false negatives and 5% of false positives, showing a correct perception of sound events in the 85% of the cases. The system gathered an overall accuracy of 88% and a precision of 83% on its measurements (see Table 3).

However, thanks to the multimodal approach, the complete system has been able to outperform the performances of the single modalities, giving a correct evaluation of true positives and negatives of 98%; when the system is not able to recognize an event through one channel, a second one will recover the situation, improving the total performances of the whole system. At last, the tracking bounding boxes and the reliable sound localization estimation assured in the 83% of the cases a correct framing of the PTZ cameras to the event sources.

6 Conclusions

In this paper, an intelligent perception system has been presented. It is capable of performing monitoring tasks based on multimodal sensors, that makes it possible to perceive different kinds of features of the environment. The system has several strong points: it is equipped with omnidirectional and perspective cameras, and makes use of
Table 3: The confusion matrices of (a) the people falling detection system and of (b) the sound identification system.

(a) Fall Detection

<table>
<thead>
<tr>
<th>Predicted classes</th>
<th>Falling</th>
<th>Not Falling</th>
</tr>
</thead>
<tbody>
<tr>
<td>Falling</td>
<td>0.90</td>
<td>0.09</td>
</tr>
<tr>
<td>Not falling</td>
<td>0.2</td>
<td>0.8</td>
</tr>
</tbody>
</table>

(b) Sound Identification

<table>
<thead>
<tr>
<th>Predicted classes</th>
<th>Scream</th>
<th>Gunshot</th>
<th>Glass</th>
<th>Bark</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scream</td>
<td>0.85</td>
<td>0.11</td>
<td>0.04</td>
<td>0.00</td>
</tr>
<tr>
<td>Gunshot</td>
<td>0.08</td>
<td>0.85</td>
<td>0.07</td>
<td>0.00</td>
</tr>
<tr>
<td>Glass</td>
<td>0.02</td>
<td>0.02</td>
<td>0.96</td>
<td>0.00</td>
</tr>
<tr>
<td>Bark</td>
<td>0.06</td>
<td>0.27</td>
<td>0.00</td>
<td>0.67</td>
</tr>
</tbody>
</table>

the novel Omnidome dual-camera sensor. Software modules have been developed to be general and work on both kinds of images: for example, the fall-detection algorithm is capable of working on both unwarped omnidirectional images and common perspective images. The system is multimodal, and is capable of offering several user-centered services that are the goal of an assisted living environment.

The system was successfully used during the industrial project “Safe Home” funded by the Veneto Region, Italy, in a real environment meant to provide assistance to elderly people, by detecting people falls and classifying sounds connected with dangerous situations. The use of the visual and auditive modalities assured a performance improvement with respect to single channels. The system has been also capable of coordinating active sensors (PTZ cameras) and a robot in order to react to perceived stimuli, for instance by letting the active cameras framing a location where a sound triggering an alarm has been perceived, or by moving the robot to that location.

Acknowledgments

The authors wish to thank Alberto Salamone, Riccardo Levorato, Andrea Cervellin and Carlo Giuliani. This project was partially funded by Regione Veneto with Cod. progetto: 2105/201/5/2215/2009 approved with DGR n. 2215, 21/07/2009 and with Cod. progetto 2105/201/4/1268/2008 approved with DDR n. 112, 15/10/2008, and with DGR.1966, 15/7/2008 “Sicurezza Integrata” and with DGR. 2549, 4/8/2009 “Safe Home.”

References
