Mobile Robot Cooperation with Infrastructure For Surveillance: Towards Cloud Robotics

Hadi Aliakbarpour, Paulo Freitas, Joao Quintas, Christiana Tsiourti, and Jorge Dias

Institute of Systems and Robotics, University of Coimbra.
{hadi,pfreitas, jquintas}@isr.uc.pt, ctsiourti@ipn.pt, jorge@isr.uc.pt

Abstract. This paper proposes a cooperative framework among a mobile robot and an infrastructure for the sake of surveillance and identification. The infrastructure is comprised of a network of cameras, inertial sensors, a facial identification system and a GPU-enabled server. The cooperation among the mobile robot and infrastructure is service-based. The scene is observed by camera sensors. There is a mobile robot freely moving within the scene. For some security reasons the system gets interested in identification of a person inside the scene. The infrastructure provides three services to the mobile robot including positions of the person and robot as well as 3D information of the person. In the other sides, the mobile robot provides the facial (portrait) image of a requested person, as a service. Using these services the infrastructure can identify the person.
The cameras in the infrastructure and also the camera on the mobile robot are rigidly coupled with an inertial sensor (IS). The inertial data provided by IS is used in two ways: definition of a fusion-based virtual camera and also virtual planes to register 3D data. Taking advantage of the defined virtual camera, a localization method is proposed to obtain the position of the robot with respect to the infrastructure. Experimental results are provided to demonstrate the feasibility and effectiveness of the proposed framework for the purpose of being used in *cloud robotics*.

Keywords: Mobile robot, infrastructure, cloud robotics, inertial sensor (IS), camera network, sensor fusion, localization.

1 Introduction

In this paper a cooperative framework among a mobile agent and an infrastructure is proposed which is towards the *cloud robotics*. Several research groups are exploring the idea of robots that rely on cloud computing infrastructure to access vast amounts of processing power and data. This approach, which some are calling "cloud robotics," would allow robots to off-load compute-intensive tasks like image processing and voice recognition and even download new skills instantly. For conventional robots, every task — moving a foot, grasping things, recognizing a face—requires a significant amount of processing and pre-programmed information. As a result, sophisticated systems such as humanoid robots need to carry powerful computers and large batteries to power them. According to James Kuffner, from Carnegie Mellon University, cloud-enabled robots

could offload CPU-heavy tasks to remote servers, relying on smaller and less power-hungry on-board computers. Even more promising, the robots could turn to cloud based services to improve such capabilities as recognizing people and objects, navigating environments, and operating tools.

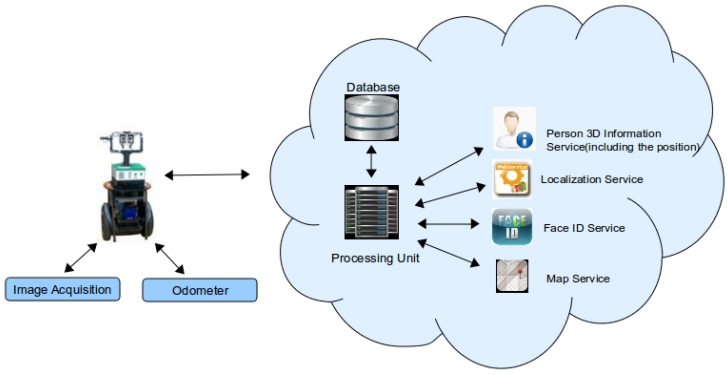

Fig. 1. Schematic diagram of the cloud robotics concept.

Recent literature as in [10], established a road-map for the next 15 years for the action and language integration in robotics, and proposed some challenges for cognitive robotics architectural approaches and implementation. In order to implement a cognitive robotic systems capable of the more advanced skills, we just can image the amount of processing power involved to do it. However these capabilities will definitely change the way social robots interact with their environment, being persons, other robots or objects in the environment.

Cloud computing concept is specially linked to Service-Oriented Architecture (SOA). The technical foundations of the Service-Oriented Architecture (SOA) [11] vision and web services are well-known and widely recognized and accepted as a suitable architectural style for developing modern applications. Service-Oriented Architecture (SOA) establishes an architectural model that aims to enhance the efficiency, agility, and productivity of an enterprise by positioning services as the primary means through which solution logic is represented in support of the realization of strategic goals associated with service-oriented computing. On a fundamental basis, a service oriented computing platform revolves around the service-orientation design paradigm and its relationship with service-oriented architecture. A SOA implementation can consist of a combination of technologies, products, Application Programming Interfaces (APIs), supporting infrastructure extensions, and various other parts. The most popular approach to implement SOA is by using web services. Their initial use was primarily within traditional distributed solutions wherein they were most commonly used to facilitate point-to-point integration channels. However, as the maturity and adoption of web services standards increased, so did the scope of their utilization. The service-oriented computing concept becomes a distinct architectural model that has been positioned by the vendor commu-

nity as one that can fully leverage the open interoperability potential of web services, especially when individual services are consistently shaped by service-orientation. For example, when exposing reusable logic as web services, the reuse potential is significantly increased. Because service logic can now be accessed via a vendor-neutral communications framework, it becomes available to a wider range of service consumer programs. Additionally, the fact that web services provide a communications framework based on physically decoupled contracts allows each service contract to be fully standardized independently from its implementation. This facilitates a potentially high level of service abstraction while providing the opportunity to fully decouple the service from any proprietary implementation details. All of these characteristics are desirable when pursuing key principles, such as Standardized Service Contracts, Service Re-usability, Service Loose Coupling, Service Abstraction, and Service Composability. Cloud computing [9] refers to the provision of computational resources on demand via a computer network. Users or clients can submit a task, such as word processing, to the service provider, such as Google, without actually possessing the software or hardware. The consumer's computer may contain very little software or data (perhaps a minimal operating system and web browser only), serving as little more than a display terminal connected to the Internet. Since the cloud is the underlying delivery mechanism, cloud based applications and services may support any type of software application or service in use today.

The topics referring to Cloud Robotics and similar subjects (e.g. Internet Robots, Robots as Web Services, etc.) are assisting to a boost of interest by the scientific community. In one hand the basic concepts related with these topics are very attractive for the future developments in robotics, which will require increased computational power, in order to execute more and more demanding and complex tasks, which is particularly expected for smart social robots. In the other hand, although these concepts are not new, with some works dated back in the 1990s, we are now in better conditions to give these approaches a renewed try. Various works describe their efforts to create an infrastructure to enable web services for robotics. In [13], the authors discuss a ubiquitous control platform for an autonomous robot that can access distributed application logic based on recent network technologies like XML, SOAP, WSDL, UDDI. To solve the ad hoc problem of how the distributed application logic can be invoked by the robot "Web Services" are presented as the best solution. Web services can speed development with a more flexible infrastructure where multiple services can work together to provide data and services for the application. The paper of [19] describes how semantic web and web services can be applied on robotics in order to facilitate cooperation between robots for joint tasks execution. By implementing Semantic Web Services on top of isolated robots within a network perspective they can be regarded as distributed web services that communicate between them semantically, allowing atomic operations to be described, parameters communicated and readjustments done in real time.

This paper is organized as following: Cloud infrastructure including its two main services which are our contributions are introduced in Sec. 2. Some experimental results are provided in Sec. 3 and. Eventually Sec. 4 is dedicated to the conclusion and future work.

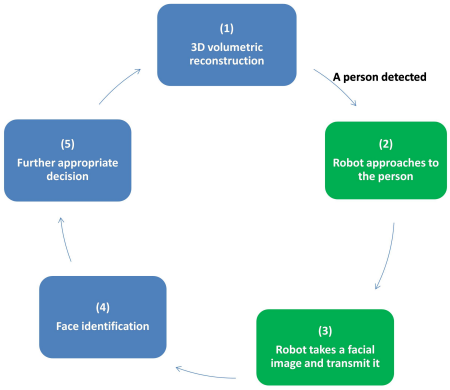

Fig. 2. Schematic cycle of the cloud robotics in the context of the proposed scenario.

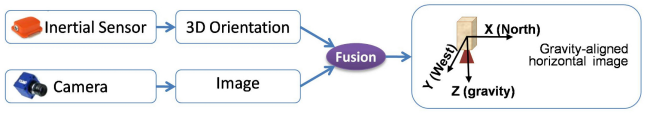

Fig. 3. Fusion-based virtual camera.

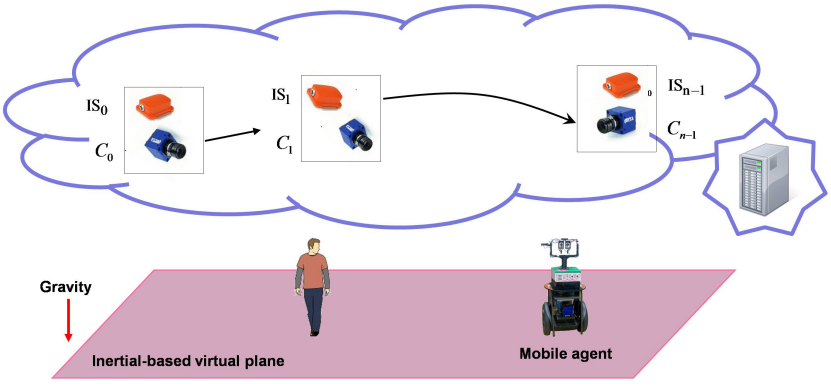

Fig. 4. Network of IS-camera couples and a mobile robot.

2 Cloud infrastructure services

The cloud infrastructure includes a network of IS-camera couples, a GPU-enabled server for performing parallel processes and an identification system. In this section the idea is to describe the services which are provided by the cloud infrastructure. As depicted in Fig. 1, these services include: (1) 3D information of the person in the scene including its position, (2) localization of the mobile agent, and (3) providing the identification of the person. We continue to introduce some appropriate methods for each service.

Since we want to divide the robot's work load according to different infrastructure nodes, the routing mechanism should consist in a set of brokers, according the type of data to be processed, encapsulating a group of messaging queues each to distribute the load among different nodes. The architecture proposed in this paper aims to integrate the mobile robot and associated functional capabilities as services. The mobile robot will act as service provider and consumer. Services will be published into a common service repository, thus making them discoverable by other remote services. A list of the services that can be provided by the infrastructure is presented in table 1.

Cloud Infrastructure Services	Robot Services
Human 3D Reconstruction [7]	Controllable Mobile Platform
Human Detection	On-Board Ethernet Camera
Human Localization	Patrolling Indoor Areas
Robot Detection [15]	Patrolling Outdoor Areas
Robot Localization	Sound Acquisition
Facial Identification	Human Interaction Interface

Table 1. List of available services in the cloud infrastructure and in the mobile robot

2.1 3D information of the person

A data registration method using IS-camera network, which provides 3D information of the person, is introduced here. The scene is observed by a camera network. Each camera within the network is rigidly coupled with an IS (see Fig. 4). Using fusion of inertial and visual information it becomes possible to consider a virtual camera instead of each couple (see Fig. 3). Such a virtual camera has a horizontal image plane and its optical axis is parallel to the gravity and is downward-looking. As a result, the image plane is aligned to the earth fixed reference frame. Fig. 6 shows a network of such virtual cameras. In order to obtain image plane of virtual camera the concept of infinite homography is used [17,6]. Since there is just a rotation among the centers of real camera and its corresponding virtual camera (see Fig. 5), the image plane of the virtual camera can be obtained by applying the following *infinite homography* transformation [17,12]:

$$^{V}H_{C} = K\,{}^{V}R_{C}\,K^{-1} \tag{1}$$

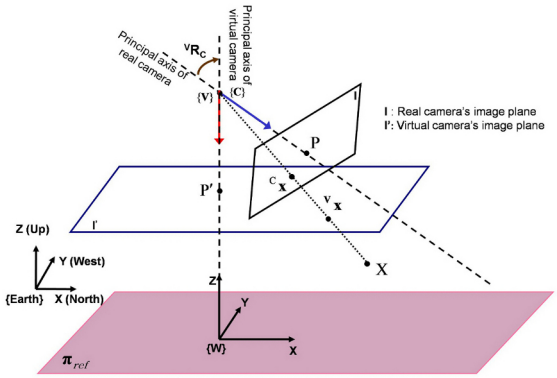

Fig. 5. Virtual camera using IS-aided homography.

where ${}^{V}R_{C}$ is the rotation matrix among the real and virtual cameras[6]. The way of obtaining ${}^{V}R_{C}$ is explained in [6].

By taking the advantage of inertial data (3D orientation), a horizontal word plane π_{ref} , which is common between all virtual cameras, has been defined in the world reference frame$\{W\}$ (see Fig. 6). The idea is to reproject the virtual image plane to the reference plane π_{ref} , for the sake of registration. The reference 3D plane π_{ref} is defined such a way that it spans the X and Y axis of $\{W\}$ and it has a normal parallel to the Z. In this proposed method the idea is to not using any real 3D plane inside the scene for estimating homography. Hence we assume there is not any real (and horizontal) 3D plane available in the scene so our $\{W\}$ becomes a virtual reference frame and consequently π_{ref} is a horizontal virtual plane on the fly. Although $\{W\}$ is a virtual reference frame however it needs to be somehow defined and fixed in the 3D space. Therefore here we start to define $\{W\}$ and as a result π_{ref}. With no loss of generality we place O_W, the center of $\{W\}$, in the 3D space such a way that O_W has a height d w.r.t the first virtual camera, V_0. Again with no loss of generality we specify its orientation as same as the earth fixed reference. Then as a result we can describe the reference frame of a virtual camera $\{V\}$ w.r.t $\{W\}$ via the following homogeneous transformation matrix

$$ {}^{W}T_{V} = \begin{bmatrix} {}^{W}R_{V} & \mathbf{t} \\ \mathbf{0_{1\times 3}} & 1 \end{bmatrix} \tag{2} $$

where ${}^{W}R_{V}$ is a rotation matrix defined as (see Fig. 6):

$$ {}^{W}R_{V} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix} \tag{3} $$

and $\mathbf{t}$ is a translation vector of the V 's center w.r.t $\{W\}$. Obviously using the preceding definitions and conventions, for the first virtual camera we have $\mathbf{t} = [\,0\;0\;d\,]^{T}$.

The reprojection of the virtual camera's image points onto the π_{ref} can be performed by applying a homography matrix, namely ${}^{\pi_{ref}}H_{v}$, since the operation is plane to plane. Here we continue to formally define such a homography matrix using the rotation and

translation between these two planes ($I^{'}$ and π_{ref}). A 3D point $\mathbf{X} = [X\ Y\ Z\ 1]^T$ lying on π_{ref} can be projected onto virtual image plane as

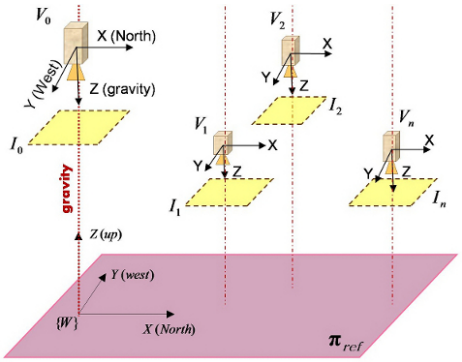

Fig. 6. A network of virtual cameras: The coordinate frames of all virtual cameras are aligned to the world reference frame.

$$\mathbf{x} = {}^{\pi_{ref}}H_v\,\mathbf{X} \tag{4}$$

where ${}^{\pi_{ref}}H_v$ is a homography matrix which maps the π_{ref} to the virtual image plane and is defined by

$${}^{\pi_{ref}}H_v = K\,[\mathbf{r}1\ \mathbf{r}2\ \mathbf{t}] \tag{5}$$

in which $\mathbf{r}1$, $\mathbf{r}2$ and $\mathbf{r}3$ are the columns of the 3×3 rotation matrix and t is the translation vector between the π_{ref} and camera center [12]. We recall that all virtual cameras have the same rotation w.r.t world reference frame $\{W\}$. In other words it can be thought there is no rotation among the virtual cameras. ${}^{W}R_V$ or the rotation matrix between a virtual camera and $\{W\}$ was described through Eq. (3). Considering ${}^{W}R_V$ from Eq. (3), π_{ref} as the interesting world plane and $\mathbf{t} = [t_1\ t_2\ t_3]^T$ as the translation vector (among $I^{'}$ and π_{ref}) and eventually K as camera calibration matrix, the Eq. (5) can be rewritten as :

$${}^{\pi}H_V^{-1} = \begin{bmatrix} f_x & 0 & f_x t_1 + u_0 t_3 \\ 0 & -f_y & f_y t_2 + v_0 t_3 \\ 0 & 0 & t_3 \end{bmatrix} \tag{6}$$

The extention of the projection from virtual camera to other virtual planes parallel to π_{ref} is described in [7] where the detail of the 3D reconstruction is deeply explained.

2.2 Mobile Agent localization

Localization of the mobile agent is one of the necessary services which the cloud infrastructure can provide. In cloud robotics, one of the ideas is to reduce as much as

possible the on board processes of the mobile robot. In this fashion, we propose a localization method in which the position of the robot is estimated using the infrastructure. As mentioned in the state-of-the-art, it is already proved that the orientation obtained from inertial sensor can accelerate and improve the matching process between wide baseline images (to estimate the transformation aiming two cameras)[14,18]. Here we take the advantage and propose a method to localize the mobile robot in the scene. As mentioned before, the rotation among all virtual cameras is equal to the identity matrix ($I_{3\times3}$). Considering the virtual camera (IS-camera couple) of the mobile agent as a part of the virtual camera network, then localization problem for the robot will become just to estimate the translation. In order to compute $\mathbf{t}$, an appropriate approach is proposed here. Our approach is based on having the heights of two arbitrary 3D points such $\mathbf{X}_1 = [X_1\ Y_1\ Z_1]^T$ and $\mathbf{X}_2 = [X_2\ Y_2\ Z_2]^T$ (see Fig. 7) w.r.t one camera (namely V_0) within the network and then to have just their correspondence points in the image plane of the virtual camera mounted on the robot (this approach is compatible with the proposed 3D reconstruction framework since we are keeping the assumption of not using any real ground plane). Suppose ${}^0\mathbf{X}_1 = [{}^0X_1\ {}^0Y_1\ {}^0Z_1]^T$ and ${}^0\mathbf{X}_2 = [{}^0X_2\ {}^0Y_2\ {}^0Z_2]^T$ are coordinations of the two 3D points $\mathbf{X}_1$ and $\mathbf{X}_2$ expressed in the first virtual camera center, respectively. Based on the assumption, the parameters ${}^0Z_1$ and ${}^0Z_2$ which indicate the heights of $\mathbf{X}_1$ and $\mathbf{X}_2$ in $\{V_0\}$ are known. Recalling that V_0 is downward and have its optical axis parallel to the gravity. Therefore the term "*height*" here is equal to the Z component of the 3D point. Then using projective property of a camera we can have all three components of ${}^0\mathbf{X}_1$ and ${}^0\mathbf{X}_2$ numerically computed in a metric scale using the Eq. (7):

$$\begin{cases} {}^0\mathbf{X}_1 = {}^0Z_1\ (K_1^{-1}\ {}^0\mathbf{x}_1) \\ {}^0\mathbf{X}_2 = {}^0Z_2\ (K_1^{-1}\ {}^0\mathbf{x}_2) \end{cases} \tag{7}$$

where ${}^0\mathbf{x}_1$ and ${}^0\mathbf{x}_2$ are respectively the imaged points of $\mathbf{X}_1$ and $\mathbf{X}_2$ in the first virtual camera image plane. The same can be considered for the second virtual camera. Suppose ${}^1\mathbf{X}_1 = [{}^1X_1\ {}^1Y_1\ {}^1Z_1]^T$ and ${}^1\mathbf{X}_2 = [{}^1X_2\ {}^1Y_2\ {}^1Z_2]^T$ are respectively coordinations of the 3D points $\mathbf{X}_1$ and $\mathbf{X}_2$ expressed in the mobile virtual camera center ($\{V_1\}$). Then likewise using projective property of a camera we can have the following equation:

$$\begin{cases} {}^1\mathbf{X}_1 = {}^1Z_1\ (K_2^{-1}\ {}^1\mathbf{x}_1) \\ {}^1\mathbf{X}_2 = {}^1Z_2\ (K_2^{-1}\ {}^1\mathbf{x}_2) \end{cases} \tag{8}$$

In contrary to the Eq. (7), Eq. (8) can not be numerically computed yet, since it has two unknown values for ${}^1Z_1$ and ${}^1Z_2$ (the heights of the 3D points w.r.t $\{V_1\}$). The terms $(K_2^{-1}\ {}^1\mathbf{x}_1)$ and $(K_2^{-1}\ {}^1\mathbf{x}_2)$ in Eq. (8) as well express the 3D position of the points ${}^1\mathbf{X}_1$ and ${}^1\mathbf{X}_2$ however up to scale factors ${}^1Z_1$ and ${}^1Z_2$. Here it is desirable to rewrite the Eq. (8) as the following:

$$\begin{cases} {}^1\mathbf{X}_1 = {}^1Z_1\ {}^1\hat{\mathbf{X}}_1 \\ {}^1\mathbf{X}_2 = {}^1Z_2\ {}^1\hat{\mathbf{X}}_2 \end{cases} \tag{9}$$

where ${}^1\hat{\mathbf{X}}_1 = (K_2^{-1}\ {}^1\mathbf{x}_1)$ and ${}^1\hat{\mathbf{X}}_2 = (K_2^{-1}\ {}^1\mathbf{x}_2)$. Then the Eq. (7) and Eq. (9) can be related through the translation vector between $\{V_0\}$ and $\{V_1\}$ as:

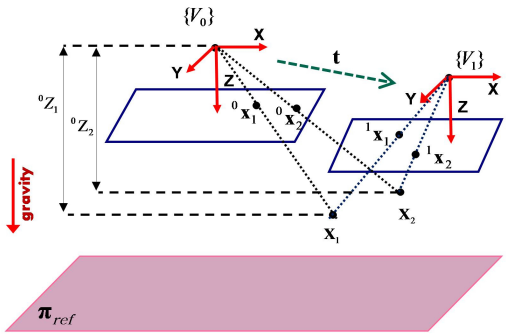

Fig. 7. Translation between two virtual cameras.

$$\begin{cases} {}^0\mathbf{X}_1 = R\,{}^1\mathbf{X}_1 + \mathbf{t} & = R\,{}^1Z_1\,{}^1\hat{\mathbf{X}}_1 + \mathbf{t} \\ {}^0\mathbf{X}_2 = R\,{}^1\mathbf{X}_2 + \mathbf{t} & = R\,{}^1Z_2\,{}^1\hat{\mathbf{X}}_2 + \mathbf{t} \end{cases} \tag{10}$$

where R is the rotation matrix between two cameras and$\mathbf{t} = (\,t_1\ t_2\ t_3\,)^T$. Since we are considering the virtual cameras and there is not rotation among them then we can simply consider R as an 3×3 identity matrix. In Eq. (10) there are five unknown parameters including ${}^1Z_1$, ${}^1Z_2$, t_1, t_2, t_3. Nevertheless there are also six linear equations which are adequate to obtain the unknowns. In order to estimate the five unknowns Eq. (10) can be arranged in the form of

$$A\,\mathbf{x} = B \tag{11}$$

where

$$A = \begin{bmatrix} {}^1\hat{\mathbf{X}}_1 & \mathbf{0}_{3\times1} & I_{3\times3} \\ \mathbf{0}_{3\times1} & {}^1\hat{\mathbf{X}}_2 & I_{3\times3} \end{bmatrix},\ \mathbf{x} = \begin{bmatrix} {}^1Z_1 & {}^1Z_2\ t_1\ t_2\ t_3 \end{bmatrix}^T \quad and \quad B = \begin{bmatrix} {}^0\mathbf{X}_1 \\ {}^0\mathbf{X}_2 \end{bmatrix}$$

Therefore $\mathbf{x}$ in Eq. (11) can be obtained using the least square approach as follows:

$$\mathbf{x} = (A^T A)^{-1} A^T B \tag{12}$$

and consequently the translation vector between one virtual camera from the structure (V_0) and the virtual camera of the mobile robot (V_1) is estimated.

3 Experiments

Some experiments have been done in the smart-room of the mobile robotic laboratory at the University of Coimbra [1], shown in Fig. 8. The superimposed area in this figure is observed by a camera network. The cameras are *AVT Prosilica GC650C GigE Color [4]*, synchronized by hardware. Each camera is rigidly coupled with an IS (see Fig. 9). Xsens MTx [5] is used as the IS. The purpose of using IS is to have 3D orientation with respect to earth, obtain virtual camera and define virtual horizontal planes. First the

Algorithm 1 Mobile agent localization algorithm (a service of cloud infrastructure).

Step 1- Take an image and an inertial data from the IS-camera couple mounted on the robot.

Step 2- Transmit the taken data to the infrastructure.

Step 3- Extract the two landmarks' positions in the image taken by mobile robot (using a mean-shift algorithm).

Step 4- Localize the position of the robot (mounted camera).

Step 5- If there is any movement by the mobile robot then repeat the steps 1 to 4.

Step 6- End.

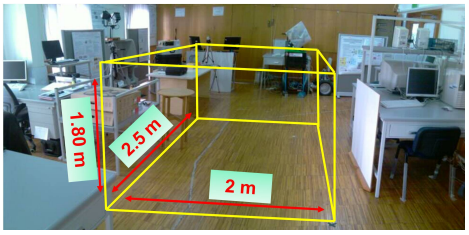

Fig. 8. Smart-room scene: The supper-imposed area depicts the area which is observed by the camera network.

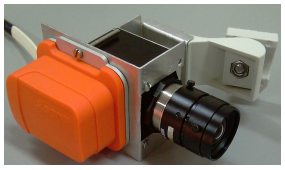

Fig. 9. The IS-camera couple used in the smart-room experiment.

Fig. 10. Left: Robot used in the experiments as mobile agent. Right: Two yellow balls used as landmarks for the mobile robot. They can be tracked using a simple colour-based tracking.

Fig. 11. Left: The robot when he approached to the person. Right: Detected face by the infrastructure using the image transmitted by the mobile robot.

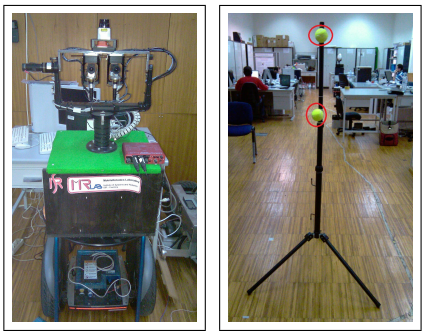

Fig. 12. Left: Robot used in the experiments as mobile agent. Right: Two yellow balls used as landmarks for the mobile robot. They can be tracked using a simple colour-based tracking.

intrinsic parameters of the cameras are estimated using *Bouguet Camera Calibration Toolbox* [8] and then *Camera Inertial Calibration Toolbox [16]* is used for the sake of extrinsic calibration between the camera and IS (to estimate ${}^{C}R_{IS}$). Fig. 12-Left shows the robot which is used in the experiment as the mobile agent. Just one of its cameras is used. Two yellow balls which are used as the landmarks are shown in Fig. 12-right. The relative heights of these two landmarks w.r.t. the one of the static cameras from the infrastructure is manually measured (it can also be done using some appropriate devices such as altimeters). Note that these two landmarks do not need to necessarily lie on a vertical line, but since we did not have altimeter available, then we used two landmarks from a vertical pole in order to minimize the measuring error.

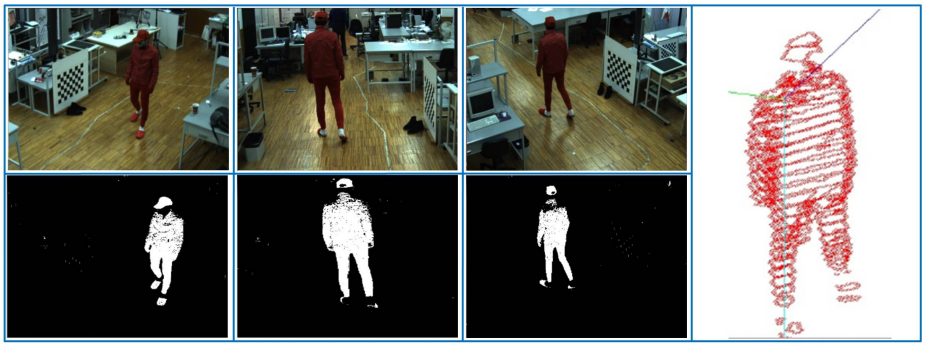

Fig. 13. Results of the 3D volumetric reconstruction of the person in the scene. The space is observed by a IS-camera network. Left pictures show the images of the person before and after background subtraction. The right image depict the results of reconstruction. 35 inertial-based virtual planes, with an interval distance of 50 *mm,* are used for purpose of reconstruction.

The general schematic cycle of the cloud robotics in the context of the proposed scenario is depicted in Fig. 2. Among the five cycles, two of them are done by the

mobile agent (green ones) and the rest by the cloud infrastructure (blue ones). This schema shows our scenario for using cloud robotics. The 3D volumetric reconstruction is running as a service (Fig. 13). Once a person gets detected, his/her 3D information (including the position) and the position of the robot is given to the mobile agent. Given these information, the mobile agent approaches to the person and take a facial picture. Fig. 11-left an image of the robot and a person in front. It is supposed that the robot can always see the landmarks in order to be localized. Then the taken portrait image by the mobile robot is sent to the infrastructure in order to identify the person. Fig. 11-right depicts the detected face which is the input for the identification system. Here the availability of a facial identification system is assumed. After having the identification result and based on the scenario policy, a further decision can be made.

The reconstruction algorithm was developed using the C++ language, OpenCV library [3] and NVIDIA's CUDA software [2] for Ubuntu Linux v10.10. The processing unit responsible for all the sensory and vision algorithm (including CUDA processing) is composed by a PC (Intel Dual Core Pentium D 950 3.40GHz processor, Cache L1 (32KB) and L2 (2048KB), 1 GB RAM, 80GB HDD and a PCI-Express NVIDIA GeForce 9800 GTX+). Fig. 13 shows the result of 3D volumetric reconstruction using the proposed approach. The person is dressed in red and we used a mean-shift based segmentation algorithm. In this experiment 35 inertial-based virtual planes, with an interval distance of 5 *cm,* are used on order to register data for the sake of reconstruction. Then the 35 layers are stacked and visualized as can be seen in Fig. 13-right.

4 Conclusion and future work

In this paper a cooperative framework among a mobile agent and an infrastructure is proposed which is towards the cloud robotics concept. The infrastructure is dominant to the scene and has computationally powerfull processing resources in order to provide services such as person 3D information, the mobile robot localization, facial identification etc.. In the other hand the mobile agent can freely move within the scene and explore it. A network of IS-camera couples are used to observe the scene. Using the fusion among camera and IS in each couple, a virtual camera is defined based on the concept of infinite homography. Such a definition made it possible to have 3D reconstruction of the scene with no planar ground assumption since a virtual ground plane is defined using inertial data. The framework is wrapped in a security scenario and some experimental results have been provided. As future work, the idea is to improve the mobile localization approach without using the landmarks. The idea is to use SIFT feature and use the 3D orientation provided by the IS in order to estimate translation part. Moreover the used heuristic path planning and navigation algorithm are intended to be improved. Another big advantage of having a mobile agent would be to have ability to make a vocal dialog with the person in the scene. This last is considered as well for the future work.

Acknowledgment

Hadi Ali Akbarpour is supported by the FCT (Portuguese Fundation for Science and Technology). The authors would like to thank Luis Almeida for his contribution in the smart-room experiments implementation, Kamrad Khoshhal and Amilcar Ferreira for their helps in the preparation of the smart-room and data acquisition.

References

1. Mrl, http://paloma.isr.uc.pt/mrl/.
2. Nvidia. http://www.nvidia.com/.
3. Opencv. http://opencv.willowgarage.com/.
4. Prosilica, http://www.1stvision.com/cameras/prosilica/gc650-gc650c.html.
5. Xsens motion technologies. http://www.xsens.com.
6. Hadi Aliakbarpour and Jorge Dias. Human silhouette volume reconstruction using a gravity-based virtual camera network. In *Proceedings of the 13th International Conference on Information Fusion, 26-29 July 2010 EICC Edinburgh, UK*, 2010.
7. Hadi Aliakbarpour and Jorge Dias. Imu-aided 3d reconstruction based on multiple virtual planes. In *DICTA'10 (the Australian Pattern Recognition and Computer Vision Society Conference), IEEE Pr., 1-3 December 2010, Sydney, Australia.*, 2010.
8. Jean-Yves Bouguet. Camera calibration toolbox for matlab. In *www.vision.caltech.edu/bouguetj*, 2003.
9. R. Buyya, Chee Shin Yeo, and S. Venugopal. Market-oriented cloud computing: Vision, hype, and reality for delivering it services as computing utilities. In *High Performance Computing and Communications, 2008. HPCC '08. 10th IEEE International Conference on*, pages 5 –13, sept. 2008.
10. A. Cangelosi, G. Metta, G. Sagerer, S. Nolfi, C. Nehaniv, K. Fischer, J. Tani, T. Belpaeme, G. Sandini, F. Nori, L. Fadiga, B. Wrede, K. Rohlfing, E. Tuci, K. Dautenhahn, J. Saunders, and A. Zeschel. Integration of action and language knowledge: A roadmap for developmental robotics. *Autonomous Mental Development, IEEE Transactions on*, 2(3):167 –195, sept. 2010.
11. T. Erl. *Service-Oriented Architecture: Concepts, Technology, and Design*. Prentice Hall, 2005.
12. Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. CAMBRIDGE UNIVERSITY PRESS, 2003.
13. Bong Keun Kim, M. Miyazaki, K. Ohba, S. Hirai, and K. Tanie. Web services based robot control platform for ubiquitous functions. In *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, pages 691 – 696, april 2005.
14. M. Labrie and P. Hebert. Efficient camera motion and 3d recovery using an inertial sensor. In *Computer and Robot Vision, 2007. CRV '07. Fourth Canadian Conference on*, pages 55 –62, May 2007.
15. J. Lobo, L. Almeida, J. Alves, and J. Dias. Registration and segmentation for 3d map building - a solution based on stereo vision and inertial sensors. In *Robotics and Automation, 2003. Proceedings. ICRA '03. IEEE International Conference on*, volume 1, pages 139 – 144 vol.1, 2003.
16. Jorge Lobo and Jorge Dias. Relative pose calibration between visual and inertial sensors. *International Journal of Robotics Research, Special Issue 2nd Workshop on Integration of Vision and Inertial Sensors*, 26:561–575, 2007.

17. Luiz Gustavo Bizarro Mirisola. *Exploiting attitude sensing in vision-based navigation, mapping and tracking including results from an airship*. PhD thesis, 2009.
18. T. Okatani and K. Deguchi. Robust estimation of camera translation between two images using a camera with a 3d orientation sensor. In *Pattern Recognition, 2002. Proceedings. 16th International Conference on*, volume 1, pages 275 – 278 vol.1, 2002.
19. L. Vasiliu, B. Sakpota, and Hong-Gee Kim. A semantic web services driven application on humanoid robots. In *Software Technologies for Future Embedded and Ubiquitous Systems, 2006 and the 2006 Second International Workshop on Collaborative Computing, Integration, and Assurance. SEUS 2006/WCCIA 2006. The Fourth IEEE Workshop on*, page 6 pp., april 2006.