Expert Systems with Applications 40 (2013) 6715–6727

Contents lists available at SciVerse ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

INT^3-Horus framework for multispectrum activity interpretation in intelligent environments

Antonio Fernández-Caballero [a,b,*], José Carlos Castillo [c], María T. López [a,b], Juan Serrano-Cuerda [a], Marina V. Sokolova [a,d]

[a] Instituto de Investigación en Informática de Albacete (I3A), 02071 Albacete, Spain
[b] Departamento de Sistemas Informáticos, Universidad de Castilla-La Mancha, 02071 Albacete, Spain
[c] Instituto Superior Técnico, Instituto de Sistemas e Robótica, 1049-001 Lisbon, Portugal
[d] South-West State University, 94, 50 let Oktyabrya, 305040 Kursk, Russia

ARTICLE INFO

Keywords:
Monitoring and activity interpretation
INT^3-Horus framework
Intelligent environments
Fall detection

ABSTRACT

The INT^3-Horus framework, dedicated to monitoring and activity interpretation in intelligent environments is introduced. Firstly, the paper introduces a general description of the INT^3-Horus approach. The following aspects of the proposal are highlighted: the framework is multisensory by nature and includes information fusion abilities; it is based on the model-view-controller paradigm; it is defined as a hybrid distributed system; it incorporates a Common Model that houses the data structures to support the exchange of information between levels of the framework. Then, the INT^3-Horus framework ontological model is introduced. The ontology is composed of a couple of classes, namely the Level Class and the DataType Class. The paper also describes the relations between both classes, as well as it introduces the notion of set of rules which determine the system functionality for a given domain. Lastly, a case of study on elderly fall detection is described to show the efficiency of the proposed framework.

1. Introduction

New generation monitoring and activity interpretation systems are characterized by a significant improvement in the possibilities of rapidly and efficiently transmitting data, voice and video to increase their performance (Gascueña and Fernández-Caballero, 2011). Some design trends may be drawn from current monitoring architectures (Castanedo et al., 2010). On one hand, the use of a large number of sensors connected to processing nodes to cover large areas aims at monitoring in real time. On the other, the attention of a human operator is attracted as soon as an event of interest is detected. The nodes independently process the information, allowing for scalability and robustness. Furthermore, the common requirements met by current architectures are: (a) information is fused from different types of sensors, (b) network constraints are taken into account, (c) security of communications is ensured between processing modules, generating standards that define the data format, (d) automatic learning is present, (e) environment and sensors are modeled, and, (f) the addition of new processing nodes is allowed without affecting the benefits.

Within the most theoretical approaches to monitoring frameworks we find a work (Wilhelm and Gokce, 2010) which proposes a programming model for designing surveillance systems in port and maritime security. There is also the IBM project called *Smart Surveillance System (S3)* (Onut et al., 2010), where a middleware is proposed for use in surveillance systems. It provides video analysis based on behaviors. Within this research the *Detec* surveillance system (Detec Home Page) is found. Its motion detection systems enable storing the events (images and their time stamps) associated to scenario objects in disk. There is also a system for the control and tracking and traffic by means of a single camera (Hsieh et al., 2006). Similarly, a system (Kamijo et al., 2000) proposes traffic monitoring for detecting accidents in crossings. On the other side, there are proposals where part of the processing is performed in a central node and the rest is distributed. These systems are denominated hybrid frameworks (e.g. (Heikkila and Silven, 1999; Pozzobon et al., 1998; Ronetti and Dambra, 2000)).

The paper introduces a case of study, namely monitoring and fall detection of elderly people (e.g. (Chernbumroong et al., 2013; Costa et al., 2012; Paoli et al., 2012; Rueangsirarak et al., 2012)). Fall detection is still a challenging and emergent problem (Anderson et al., 2006), especially for monitoring special-needed elderly people (Chu et al., 2012; Doukas et al., 2011; Khawandi et al., 2011). The problem has mainly been provoked and induced by population ageing, showing a tendency of permanent growth

* Corresponding author at: Departamento de Sistemas Informáticos, Universidad de Castilla-La Mancha, 02071 Albacete, Spain. Tel.: +34 967599200; fax: +34 967 599224.

E-mail addresses: Antonio.Fdez@uclm.es, caballer@dsi.uclm.es (A. Fernández-Caballero).

0957-4174/$ - see front matter
http://dx.doi.org/10.1016/j.eswa.2013.06.058

in accordance with recent demographical predictions. About a third part of humans aged 65 and above suffer from a fall each year, increasing up to 42% for elderly over 70 years. Moreover, every year approximately around 50% of humans who live in long-term care institutions fall and 40% of them experience recurrent falls. In addition to this, elderly people are highly vulnerable, as falls are known to be one of the leading causes of injuries and death. Another important consequence of recurrent falls is the post-fall syndrome that manifests through depression, loss of autonomy, immobilization, and may result in impairments in daily activities (World Health Organization, 2007).

2. Description of INT^3-Horus

INT^3-Horus is a new multisensory framework to carry out monitoring and activity interpretation. The framework establishes a set of levels with some clearly defined input/output interfaces to provide a hierarchy to the processing. The levels consist of a set of modules that incorporate the algorithms dedicated to processing at each level. If thinking of several sensors that provide input information, at the lowest level (the acquisition level) several modules, each one responsible for the acquisition of a type of sensor, are located. For each level, the framework provides a set of inputs and outputs to be met by the modules (see Fig. 1). The inputs and outputs are independent from each other and from the number of modules. Thus, a module is not required to implement all inputs and outputs on its level, but it may implement the subset that best fits its needs. A higher level task is in charge of selecting those modules at different levels that are compatible with each other to create a monitoring system based on the framework.

Thus, INT^3-Horus allows the coexistence of modules that are in charge of information of different nature within a single level, although they conceptually work at the same processing level. For instance, at the level of acquisition there are some modules that capture information from cameras, while others are prepared to capture data from wireless sensor networks. Although both sources of information seem incompatible a priori, upper levels house algorithms to merge and operate with them, regardless of the data capture algorithms. Following the scheme described in Fig. 1, the levels of the framework establish a hierarchy from the level of sensor information acquisition to the level of activity analysis, by connecting the inputs of the immediately upper level through the outputs of the lower level (see Fig. 2).

Next the most significant characteristics of the INT^3-Horus framework are described.

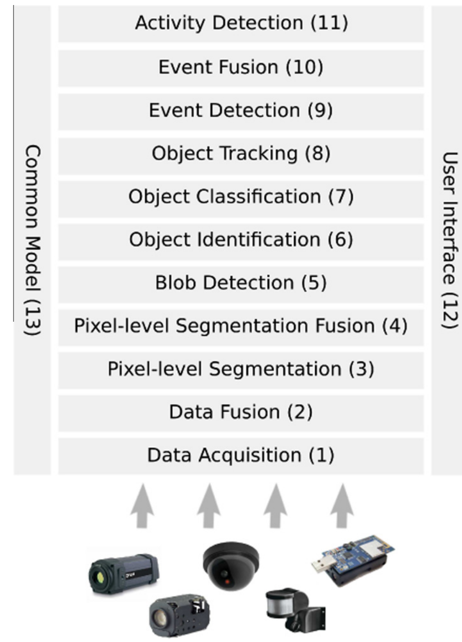

Fig. 2. Framework levels.

2.1. INT^3-Horus is multisensory

The framework is designed to work with different sources of information. The sources are mainly based on vision sensors as they are the most widely used for monitoring tasks. Nevertheless, other sensor technologies are introduced to provide the framework of greater power and flexibility. These technologies are mostly sensors used in commercial surveillance, that is, volumetric sensors, presence detection sensors, contact sensors for doors and windows, etc. The framework also includes the ability to access information from wireless sensor networks (WSNs), which allows rapid deployment of sensors in the area to be monitored, regardless of its characteristics (indoor, outdoor, and so on).

2.2. INT^3-Horus includes information fusion

As the framework operates with different data sources, INT^3-Horus includes information fusion algorithms in its design. For this reason, the JDL data fusion model has been used as basis, adapting some of its levels from military to civilian. The JDL model was developed by the *Joint Directors of Laboratories Data Fusion Group*, a committee of the United States Department of Defense (DoD). The architecture proposed by the JDL model is considered the de facto standard for implementing a surveillance system. The proposed model illustrates the main functions, relevant information and databases, as well as the interconnection required to perform data fusion. The JDL fusion model also provides a definition of the concept of data fusion (United States Department of Defense, 1991) that was later refined as a "multilevel, multifaceted process dealing with the automatic detection, association, correlation, estimation, and combination of data and information from single and multiple sources" (DSTO, 1994).

After studying the different fusion levels proposed by the JDL model, Fig. 2 shows a schematic view of our proposal for the union of traditional surveillance processing levels with the JDL fusion levels. Following this approach, processing starts at the level of acquisition (1), where several sources provide different types of information (e.g. color cameras, thermal cameras and volumetric sensors). After that, the proposal establishes an initial fusion level

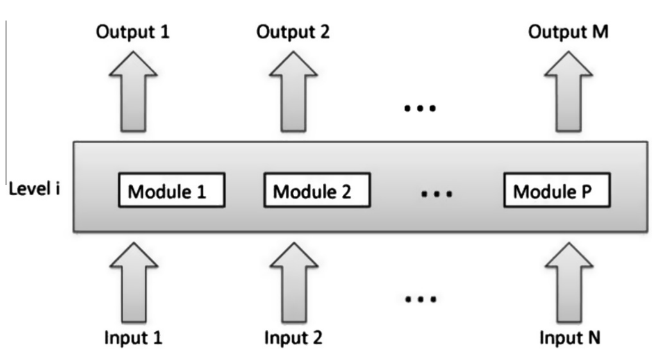

Fig. 1. Example of level with inputs, outputs and operation modules. Despite inputs are common for the whole level, it is not mandatory for each level to manage them. The same happens with outputs; they reflect the format of the level outputs, which varies depending on the implemented modules.