A SIMULATOR FOR DISTRIBUTED AMBIENT INTELLIGENCE SENSING

Julien Pansiot, Benny Lo, Guang-Zhong Yang

Royal Society/Wolfson Medical Image Computing Laboratory
Imperial College London, United Kingdom

In pervasive healthcare, extensive research is being directed towards the use of sensor networks for promoting healthy behaviours, early disease detection, improved treatment compliance, and support for informal care giving. For the elderly, home-based healthcare encourages the maintenance of physical fitness, social activity and cognitive engagement to function independently in their own homes. Existing research has shown that when privacy and security issues are properly addressed, video based sensor networks provide an effective means of monitoring behaviour changes. The purpose of the UbiSense project is to develop a pervasive homecare system based on distributed vision sensors.

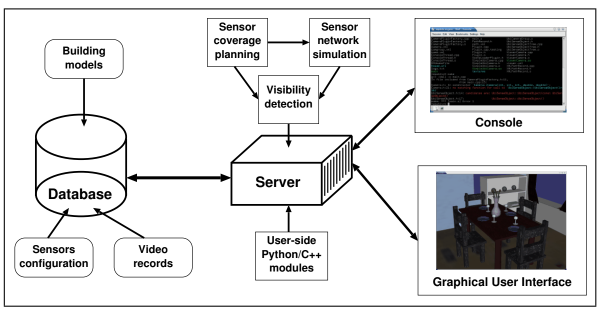

Figure 1: Overview of the simulator

To enable the development of novel vision-based health monitoring and distributed processing techniques for different home settings, a simulator for distributed ambient intelligent sensing is designed which can simulate different home layouts, camera views and activities. This allows integrated design and implementation of algorithms, activity tracking and distributed processing. For behaviour modelling, an activity database linked to segmented video frames is used to represent the scenario of how the movement of the subject is captured by different sensors. The database contains XML data files of different building models, floor plans, static and dynamic human models, real/simulated video clips and both intrinsic and extrinsic parameters of the sensors including camera configurations and computational pipeline based on DSP or FPGA.

Figure 2: A view of the former version of the simulator

Video processing workflow management is implemented where the module server provides the functionalities for database access and feature tracking library binding. The system allows the primitives to access and manipulate the scene objects. High-level tools are also provided, such as a basic visibility detection algorithm to decide whether or not a person is visible through a given camera. The simulator can also be extended with user algorithms by dynamically loaded plug-ins where algorithm prototyping will be achieved through the Python programming language. This will be useful to test camera coverage planning and to simulate in real time the general behaviour of the sensor networks. In particular, parameters such as computational resource sharing and message transfer between sensors can be tracked.