iVision based Context-Aware Smart Home System

Rustam Rakhimov Igorevich, Pusik Park, Jongchan Choi
SoC Platform Research Center
Korea Electronics Technology Institute
Seongnam, South Korea
{rustam, parksik, choijc}@keti.re.kr,

Dugki Min
Computer Science Department,
Konkuk University
Seoul, South Korea
dkmin@konkuk.ac.kr

Abstract—**Ultimate goal of ubiquitous computing is to move computers away from the user's focus, by embedding them into the environment. The user will be surrounded by smart and internet powered devices. In this work we describe our research about the Smart Home system. Our solution contains all sorts of smart entities necessary at home. For the interaction with the user we defined two outputs: Smart Wall and Smart TV, and one input: iVision. iVision system is designed and implemented on SoC (System on Chip), to get higher performance, and less latency. Smart home appliances are implemented based on DPWS (Device Profile for Web Services). Additionally Context-Awareness driven architecture was suggested, to improve the quality of the service in a Smart Home system.**

Keywords- ubiquitous computing; iVision; context-awareness; DPWS; Smart TV(key words)

I. INTRODUCTION

The popularity of ubiquitous computing increases availability of heterogeneous network infrastructures. Ubiquitous computing creates smart environments, with smart, tiny embedded devices. The most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it [1]. Smart Home is one of the sub areas of Ubiquitous computing. These days' smart home systems gathered huge popularity in a society. The popularity is mainly related to the ultimate goal of the Smart Home system. Smart Home system one of the main user centric system.

In this work we designed and implemented smart home system. While designing we try to keep the architecture of home system as simple as possible. The sorts of important and necessary devices such as air-conditioner, refrigerator and oven were selected to embed with web services. These devices were not actually manufactured, since we don't have the technology and resources to produce them. Home appliances were prototyped in embedded boards. Some of these devices were equipped with Touch LCD screen. Additionally user friendly interface were programmed especially for the LCD screen devices. Each smart device implemented based on DPWS stack. It means smart devices contain main basic Smart device operations, such as discovery, function request and event handling and generation. Another novelty part of this work is related to the application of vision based solution to detect context in a real-time. There are many types of vision-based systems for surveillance and context detection in smart homes. We suggested iVision binocular camera solution that designed based on SoC technologies.

The remainder of this paper is structured as follows: in a section II, we introduce the process how our smart home system has been built and interacts. iVision is another important and key feature of this work that related to user input. iVision system is described in section III. In Section IV we describe context-awareness architecture based on our Smart Home system. Section V contains some of the improvement notes and experimental results from suggested solution. Finally we conclude this paper in section VI.

II. THE NOTION OF SMART HOME SYSTEM

There are a lot of solutions for designing and implementation of the Smart Home system. All of them have partially different solutions and different approaches in designing. One of the simplest approaches of designing is considered user as a part of the home environment. That way the architecture becomes not user centric, but it becomes an environment oriented.

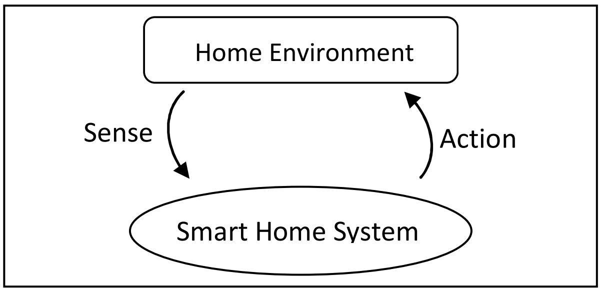

Figure 1. Smart Home System from the CPS point of view

Illustration from Figure 1, reminds CPS (Cyber Physical Space) notion. Our experience has shown the designing home system from the CPS perspective makes it more flexible, since it is actually CPS system. The user should be considered as a part of the home environment, and context-aware service as a behavior description of the user.

Any action from the environment will cause a proper reaction from the smart home system. Actions and changes happened in the home environment can be sensed using proper sensors. For example when early in the morning sun rises and huge amount of sunshine comes into the room, home system gets light information from the light detection sensor. When the change in the environment delivered to the central block the decision making process gets started.

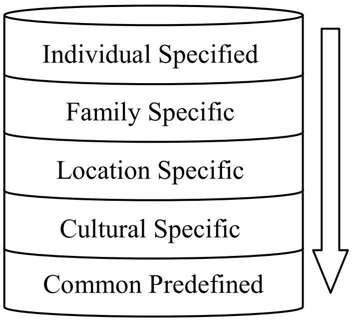

Figure 2. Context-Logic Stack for Making Decision

Smart home system contains context-logic information that will help to make the right decision. Context-logic is always evolving and mainly contains individualized data for the specific home. Context-logic can be demonstrated in a stack form as shown in Figure 2. Each layer can be served as an exceptional case for the upper context-logic. At the stage of making a decision first of all individual specified context-logic will be observed. To continue with the previous sunrise example, different individuals prefer a different amount of sunlight. That means the context-logic should collect or derive the individual specified solutions. Derive here refers to the intelligence that can conclude decision based on previously collected data. When the smart home system faces difficulties in concluding solution on one layer it downgrades to one layer. Common Predefined layer is the most bottom layer of the Context-Logic Stack, and it serves as an exceptional layer for all other layers. By other words Common Predefined layer can be considered as a default layer (factory settings).

III. EYES OF CONTEXT-AWARENESS SYSTEM: IVISION

To detect user's context we need to use different sensors all over the house. For example automatic doors which uses a sensor to detect the presence of a person in a front of it, to switch between open and close states.

Alex Pentland [7] is the one of the first researchers who designed smart room. His smart rooms used video-based techniques to identify the location, identity, facial expression and hand gestures of the people in the room. Detecting humans and their actions from multiple stationary cameras were performed by Utsumi [8] et al. The image sequences are indexed and key frames are extracted from the image sequences. Visualization of motion paths for humans in the scene are created faster way of tracking human movements.

With our solution we come up with our own solution for the sensor. iVision binocular camera is our contribution, which calculates depth map of the upfront scene image, by comparing two shifted images from two cameras. As it is known the stereoscopic image processing is not an easy process, and software solution cannot be applied in a real-time system. Assessing the expected latency on software, we implement it in a hardware level. Suggested hardware architecture performs few more operations over the extracted depth map image, such as filtering by threshold, masking, searching for blob and listing its properties. Additionally it classifies extracted blobs and rearranges them in order. Rearranged blob information stored in a fixed register of the hardware. A developer can access to these registers from the software by using few different interfaces (i2c, SRAM, UART).

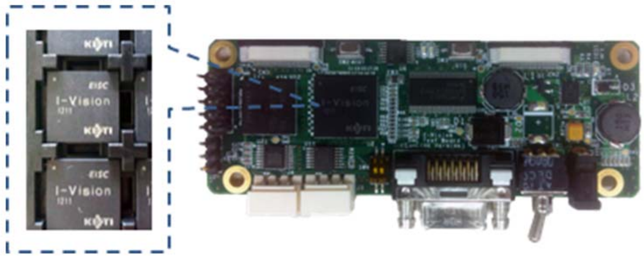

Figure 3. iVision Chip and PCB(Printed Circuit Board)

One of the main application usages of the iVision hardware solution (shown in Figure 3) is for hand tracking. Using the depth map generated by stereoscopic camera, we can cut off the outstretched hand of the user. By performing this operation over the sequential frames the user's hand can be tracked. The dynamics of the hand motions can be easily derived, and some predefined gesture can be detected. iVision system is not bounded by only hand tracking and gesture recognition application, it can be used in many other fields.

Various applications were implemented to improve the quality of the iVision system. Since it works based on stereoscopic view camera, it has specific problems, such as horizontal noise in some environments. Two hand gesture recognition [2], distant adaptive algorithm [3] and the user distance advisor [4] kind of applications were built on top of this iVision system. Most of ubiquitous homes designed to use many cameras and microphones in each room. iVision system can be best solution to reduce the number of camera's per room. Using stereo camera solution the 3D location of the user's and other objects can be derived.

To know the performance and suitability of iVision solution to smart home systems comparison with Kinect sensor solution were performed. The processing time of rectification and block matching for depth image takes almost 67%. It means the main portion of latency appeared by rectification and block matching processes. The removal of this amount of processing consumption is significant outcome and therefore should be implemented on the chip. For that reason, the Microsoft Kinect's PrimeSensor™ as well as the suggested iVision stereoscopic system supports hardwired depth map generation.

The remaining parts of algorithm such as grabbing of images, low path filtering, depth filtering, and hand tracking processes consumes big computation power. If the main system doesn't include a powerful processor like the recent multi-core CPUs the result will not be sufficient. The rest, 34% can be executed by the processor after all.

Brightness	Gesture	Kinect Solution	iVision Solution
Bright Space	Right	0	94
	Left	0	82
	Up	0	94
	Down	0	94
	Click	0	100
Shade Space	Right	98	94
	Left	98	82
	Up	92	94
	Down	94	94
	Click	82	100

Table I. Gesture Recognition Performance Comparison between Microsoft Kinect and iVision System.

As it can be seen from Table I, the Kinect sensor doesn't work when the gestures are performed in the bright environment with sunshine.

	Software Solution [Cycles]	Hardware Solution [Cycles]
Total	123,042K	10,022K
Grab images	4,016K	10,022K
Rectification	40,883K	-
Generation of Depth Map	41,256K	-
Low Path Filter	12,862K	-
Depth Filter	21,177K	-
Hands Pose Tracking	2,845K	-

Table II. Distribution and Standard Deviation of the CPU Cycles

The experimental results show the distribution and deviation for each process. In this section, the table II shows the actual number of CPU cycles for each process and compares them between different types of the stereoscopic system. Two columns contain software and hardware solution results which are received using experimental methods. The huge difference between two columns can be seen easily, because except image grabbing operation all other operations were carried out by hardware solution.

Two different systems have different grabbing methods but, nevertheless, almost all codes such as a low path filter, depth filter, and hands pose tracking are same. In spite of the same code, the numbers of measured cycles are a little bit different. Although the code for depth filtering is same, 21,177K cycles and 31,604K cycles in the table V, and 20,556K cycles and 21,998K cycles in the table VI, show different numbers of cycles.

This is why the test environment, the Microsoft Windows XP Professional, is the multicasting operating system; indeed, it is not the real-time operating system. In addition, the background processes are working during the experiments.

IV. Context-Awareness Service Architecture

The main contribution of this work is adding context-awareness service for the controlling home appliances. When web service driven smart home devices subscribes directly to the iVision system, they can receive proper events. If there is only one device is subscribed to iVision event, then the resulting action of the event is clear. If the number of subscriber devices more than one then triggering some action by certain gesture become trickier. Certain user motions might activate multiple appliances. The main reason for that is the small range of available and natural gestures. Especially natural, since it will not be easy to teach users to perform some complicated gestures to run specific services. If so then users should remember not only complicated gestures but also what these gestures are doing. When the gestures are not natural, it will be difficult to perform them, and properly the detection rate goes down. That is why the natural gestures are really important in ubiquitous technologies and in its context.

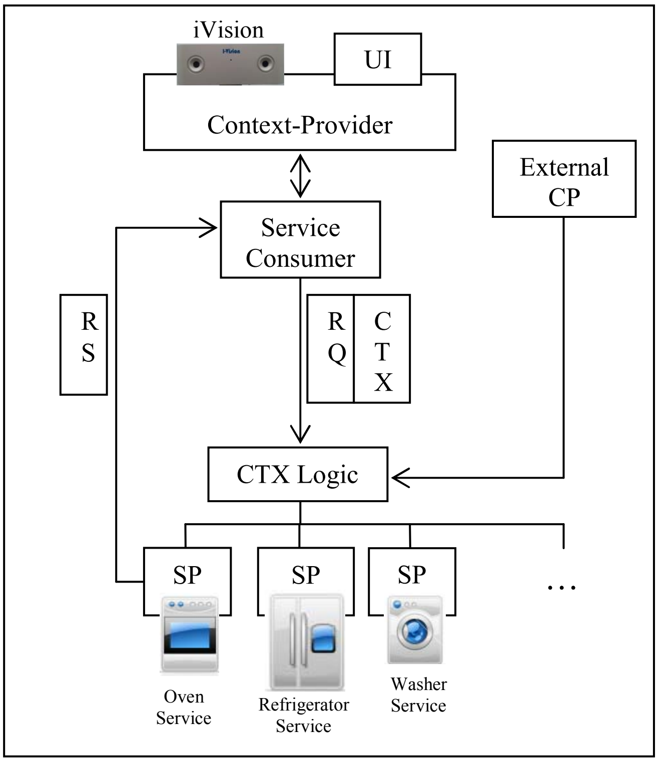

Figure 4. Motion and Gesture Recognition Hardware Architecture of iVision System

For instance, if the user performs hand up gesture, it could mean sound up or get up the air-conditioner temperature. To avoid this confusion, intuitively context should be studied. Context might be mentioned by user speech little before the actual gesture, or the user's gaze. Since in our solution we already have iVision system, the gaze detection can be properly solved. If the camera does not support that kind of service, context-aware system might just subscribe to the video streaming service from the camera. That means actual gaze recognition algorithm should be implemented in the Context-Awareness Service.

As it is shown in Figure 3, the Context-Awareness Service locates in between User Input interface and smart home appliances. Context-Awareness has a complex architecture with the huge database, to save semantics defined by a smart home ontology. Context-Awareness has three basic architectures [5], during this research work we define that dynamic context-awareness architecture is the most proper for smart home systems. iVision system serves as a physical entity of the context-provider. Using iVision a lot of context information can be extracted. All extracted context information would be packed into the request message (RQ) generated by service consumer. Context Consumer Logic (CTX Logic) can extract the context information from the encapsulated message. Depend on the context of the user the CTX Logic will trigger proper Service Provider. Since this is dynamic context-awareness architecture, External Context (EC) not generated from the iVision can also activate services. When the request generated by the user is processed last service will generate a response message to the user. SP refers to the service provider process the context information, along with request details, by selecting local logic that best meets the user's needs in that context. The response message will be generated based on the received request. The response message (RS) will contain the result value about performing service. For example, if a user decides to switch on the microwave oven and put it into preheating mode, then completion information will be packed into the response message from the service provider.

V. SOME IMPROVEMENTS FOR WEB SERVICE COMMUNICATION PART

As it was mentioned the smart devices collaborated in a smart home were DPWS driven devices. For the dpws implementation ws4d-gsoap [9] stack developed in Rostock University was used.

DPWS specifies a set of built-in services[10]:

• Discovery services: used by a device to advertise itself and/or to discover other devices.

• Metadata exchange services: provide dynamic access to the metadata of a device's hosted services.

• Eventing services: allowing other devices to subscribe to asynchronous event messages produced by a given service.

The experiment was performed on S3C6410 embedded boards. To get more realistic results the grid of ten embedded boards was deployed (Figure 5). All the experiments and experimental results refer to the real web services that can run on real devices such as air conditioner, refrigerator and microwave oven and so on.

Figure 5. Grid of S3C6410 Embedded Boards, with Ws4d-Gsoap Stack Driven Web Services

Out of experiment some are the useful points were made, about applying ws4d-gsoap stack for realizing smart home:

- When the ws4d-gsoap stack was deployed on embedded devices, probing multiple devices required long time, compared to probe only one device in local network.
- If two different devices with same web service name appeared in the network, probe function of DPWS requires longer time or sometimes even fails to detect any devices in the network.
- Network start behaving wrong, when a lot of probe messages were sent to the network
- If the network has some problems (such as few devices with the same web service name are involved) then total detection of N dpws-devices in the local network will require approximately t=N*2 seconds.
- Resolving detected drivers takes 3 more seconds per device.
- In the network without external noises the to probe 10 devices it will require 1300 ms

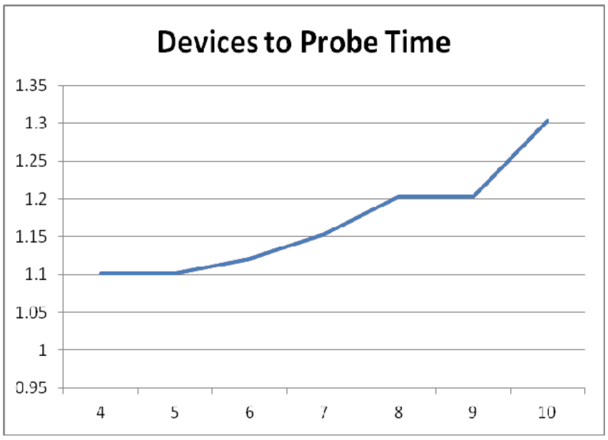

Figure 6. Number of Devices, Accordingly Necessary Time to Probe these Devices in the Network

Final experiment gave result shown in Figure 6. It shows how the probing time increases according to a number of devices. The main reason of increasing the probing process according to the number of devices is mainly related to the multicasting method. When the client sends a probe signal by multicast all web-service devices start responding, and kind of jam of response signals appearing. This conclusion was made by trying client web-service on different targets. When it was implemented on embedded board, it works faster and gives better results in performance. When it runs on common PCs with WindowsXP operating system, its performance goes down, it is mainly related to the scheduling system of WindowsXP operating system.

Originally a ws4d-gSoap library can deliver single events less than 1ms, but it doesn't support sequential streams of events. We changed the common flow of Eventing in ws4d-gSoap so it can stream the events. To do that:

- Some declarations were done globally
- Initialization and Namespace definition was done only once
- SOAP_DONE step was skipped, for the sake of providing high speed sequential events.

After changes in a source of eventing process, the performance of eventing got improved 50 times. In non-optimized Eventing procedure the delivery of the event was not stable. It was included in a range approximately 300~400ms. After performing optimizations it takes 7ms between two sequential events.

VI. CONCLUSION

In this paper we developed a Smart Home system which interacts with the smart home, made by web-service driven smart appliances. We also introduced iVision, it is another necessary web-service driven user input device. Because of iVision system and its useful features, the context-awareness service becomes available to be implemented. Since context-awareness is the key feature of smart home systems. CPS like approach was suggested to design context-aware system.

The work is funded by the Ministry of Knowledge Economy (MKE) of Korea

ACKNOWLEDGMENT (HEADING 5)

This work was supported by the IT R&D program of MKE/KEIT. [Project No.: 1004125, C&D Infotainment System Development and Software Eco-System for the Future IT-Convergence Automobile].

REFERENCES

[1] M. Weieser "The computer for the 21st century", Scientific American, pp. 94-98, 1991.

[2] Igorevich, R. R., Park, P., Choi, J., Min, D.: Two hand gesture recognition using stereo camera. In: IEEE 3rd Signal Acquisition and Processing, vol. 1, pp. 525--529 (2011)

[3] Igorevich, R.R., Pusik Park, Dugki Min, in: Application of Information and Communication Technologies (AICT), 2010 4th International Conference on, "Hand gesture recognition algorithm based on grayscale histogram of the image". 28 October 2010, Tashkent

[4] Rustam Rakhimov Igorevich, Pusik Park, Jongchan Choi and Dugki Min. "Knowledge Based Interactive Smart Camera", Proceedings of the 2012 International Conference on Effective Computing and Intelligent Interaction (ICACII 2012)", Lecture Notes in Information Technology, pp. 388-393, Taipei, Feb, 27-28, 2012

[5] http://www.ibm.com/developerworks/webservices/library/ar-conawserv/index.html

[6] Pentland, A., 1996. Smart Rooms, Scientific American: 54-62

[7] Utsumi, A., More, H., et al., 1998b. Multiple view-based tracking of multiple humans. Proceedings of the 14th International Conference on Pattern Recognition. 1: 597-601.

[8] Pusik Park[1,2], Rustam Rakhimov Igorevich[1], Jongchan Choi[1], Jongho Yoon, "Performance Evaluation of Hand Tracking with Stereoscopic Vision" International Journal of Ubiquitous Computing and Internationalization. Vol.3, No. 1, April 2011, pp 37-43

[9] http://ws4d.e-technik.uni-rostock.de/

[10] Jammes, F.; Mensch, A.; Smit, H.; , "Service-Oriented Device Communications Using the Devices Profile for Web services," Advanced Information Networking and Applications Workshops, 2007, AINAW '07. 21st International Conference on , vol.1, no., pp.947-955, 21-23 May 2007