BeaStreamer-v0.1: a new platform for Multi-Sensors Data Acquisition in Wearable Computing Applications

Pierluigi Casale
TomTom

Oriol Pujol
University of Barcelona

Petia Radeva
Autonomous University of Barcelona

BeaStreamer-v0.1 : a new platform for Multi-Sensors Data Acquisition in Wearable Computing Applications.

Pierluigi Casale*, Oriol Pujol* and Petia Radeva*

* *Computer Vision Center, Campus UAB, Edifici O , Bellaterra, Spain*
Dept. of Applied Mathematics and Analysis, University of Barcelona, Barcelona, Spain
E-mail:pierluigi@cvc.uab.es

Abstract

In this paper, we present *BeaStreamer-v0.1*, a new wearable computing platform designed fusing the Beagleboard hardware platform and the GStreamer software platform. The device has been designed for monitoring a variety of day-to-day activities and to be used as a 24/24h digital personal assistant.*BeaStreamer-v0.1* can acquire data collected from multiple sensors in controlled and uncontrolled environments. The benefits of using BeaStreamer-v0.1 are multiple. First, the small size of the Beagleboard allows to use a really portable computer device. In addition, Beagleboard ensures laptop-like performances despite its dimentions. Using GStreamer makes managing the parameters in the adquisition of many different media types simple and allows to joint the acquisition of different types of data under a unique and compact framework. We demonstrate how the adquisition of audio, video and motion data can be easily performed by BeaStreamer-v0.1 and we point some highlights in the computational power of the system, some of them to be exploited as future lines of the work.

Keywords: Wearable Sensors, BeagleBoard, Social Sensors, Multimodal Data Fusion, Pattern Recognition Applications.

1 Introduction

In a recent interview to CNN, Gordon Bell (from *Microsoft Research*) tells how and why he had been recording every single event in his life over the last decade. He carried around video equipment, cameras and audio recorders to capture conversations, trips and any kind of experiences. Bell says that this huge amount of data (more than 350 GigaBytes, not including the streaming audio and video) is a replica of his biological memory. This digitized *eMemory* is never forgotten. *Microsoft* is working on a SenseCam [1], shown in Figure 1. *SenseCam* is a camera that can be worn around the person's neck and automatically captures every detail of daily life with photos.

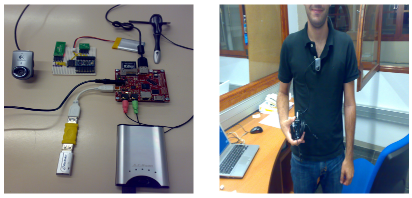
Figure 1: Images of the SenseCam, from [1].

In the same direction, *Intel* has been working on the *"Everyday Sensing and Perception"* project[2]. A team of twelve researchers has been working for three years on the *90/90 Challenge*, i.e. in building a real-time system for egocentric recognition of handled objects that is accurate at 90% over 90% of our days.

These examples clearly show an increasing interest in developing perception-based systems capable of monitoring a variety of day-to-day activities, both in the research community and in the industry as well. A system being aware of both context and activities during daily life not just would be able to give assistance in memory-retrieval tasks, but also for real-time assistance to not completely self-sufficient people.

In this paper, we present the first version of *BeaStreamer*, a wearable system for multi-sensors data acquisition and analysis that, in experiences similar to [1] and [2], could be successfully used as a 24/24h digital personal assistant. Using wearable devices such BeaStreamer-v0.1 opens the opportunity to define new use cases, such as healthcare monitoring in patients rehabilitation or studying of social behaviour of people.

The article is organized as follows. In Section 2, we will describe the system and its parts, both hardware and software. In Section 3, we will show some examples of data acquisition and analysis with BeaStreamer-v0.1. Finally, we will discuss conclusions and future works.



(a) (b)

Figure 2: (a) The *BeaStreamer-v0.1* system;
(b) The *BeaStreamer-v0.1* system worn by a tester.

2 BeaStreamer-v0.1

Beastreamer-v0.1 is a wearable system designed for real-time multi-sensors data acquisition. In this work we use it for acquiring audio, video and motion signals, but its capabilities are not restricted to these data types. Any kind of data flow might be acquired from the system and stored in memory. In Figure 2(a) we show the system disassembled on a table, showing all the components. In Figure 2(b) we show a tester wearing the system.

The system can be easily brought in one hand or in a little bag around the users waist. The audio and video dataflows are acquired using a standard low-cost webcam that can be hooked to the shirt just down the neck or at chest level. An Arduino-based bluetooth accelerometer, can be put in the pant pocket or in the shirt pocket. Audio and video data are acquired via GStreamer, motion data are acquired via bluetooth. Although at the moment, the main functionality of the system is data acquisition, the system has been designed also for data analysis.

The core of the system is based on *Beagleboard*, an OMAP-based board with high computational power. The system is equipped on-board with a 4 Gigabytes SD-Card where both operating system and data acquired are stored. In the next sections, we describe the hardware components, the development environment and finally, the operating system and the application software running on the board.

2.1 The Hardware Core : BeagleBoard.

The *BeagleBoard* (BB)[3], shown in Figure 3, is a low-power, low-cost single-board computer produced by *Texas Instruments (TI)*.

With open source development in mind, BB has been developed to demonstrate the potential of *TI's OMAP3530* system-on-chip, though not all OMAP functionalitites are available on the board. The BB sizes approximately *80mmx80mm* and it provides all the functionalities of a basic computer.

The *OMAP3530* system-on-chip includes an *ARM Cortex-A8* CPU at 600 MHz which can run Windows CE or Linux, a *TMS320C64x+ DSP* for

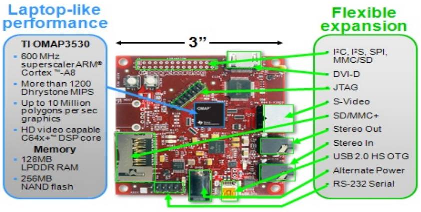


Figure 3: Beagleboard front view, from [3].

accelerated video and audio codecs, and an *Imagination Technologies PowerVR SGX530 GPU* to provide accelerated 2D and 3D rendering that supports OpenGL ES 2.0. Built-in storage and memory is provided through a *Package on Package* chip that includes 256MBytes of NAND flash memory and 256MBytes of RAM. The board carries a single SD/MMC connector, supporting a wide variety of device such as WiFi Cards, SD/MMC Memory Cards and SDIO Cards. One interesting feature of the OMAP3530 is the possibility of booting the processor from SD/MMC card.

Video output is provided through separate S-Video and HDMI connections. A *4-pin* DIN connector is provided to access the S-Video output of the BeagleBoard. This is a separate output from the OMAP processor and can contain different video output data from what is found on the DVI-D output. The BB is equipped with a DVI-D connector that uses an HDMI connector. It does not support the full HDMI interface and it is used to provide the DVI-D interface only.

Two USB ports are present on the board. Both ports can be used as host ports, using High Speed USB devices conform to USB 2.0 protocol, using a maximum of *500 mA* to power the host device. If additional power is needed or multiple devices as mouse, keyboard and USB mass storaged devices must be used, one USB port can be used as OTG (On-The-Go) port to drive an self-powered USB hub. The USB OTG port can be also used to power the board from a standard external USB port. If both USB ports need to be used, there exists an additional *5 mm* power jack to power the board. DC supply must be a regulated and clean *5 Volts* supply. The board uses up to *2 Watts* of power.

Beagleboard presents on board a populated *RS-232* serial connection where a serial terminal is present. Using the terminal, it is possible to set the boot parameters and the size of the video buffer. Furthermore, a *14-pins* JTAG connection is present onboard to facilitate the software development and debugging on-board using various JTAG emulators. Two stereo *3.5mm* jacks for audio input and output are provided. An option for a single *28 pin* header is provided on the board to allow the connection of various expansion cards. Due to multiplexing, different signals can be provided on each pin providing more that *24* actual signal accesses. This header is not populated on the BB and, depending on the usage scenario, it can be populated as needed. Because of the efficient power consumption, the board requires no additional cooling.

Typical usage scenario for the BB are shown in Figure 4. BB might be considered a laptop substitute. There are many projects using BB in

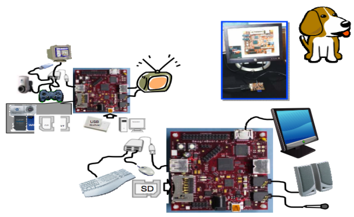


Figure 4: Tipical Usage Scenarios for Beagleboard, from [3].

robotic applications ([4], [5]). Nevertheless, up to now, there is no literature using BB as a wearable device, despite of the low dimensions of the board. The major issue of using BB in wearable applications is the need of a portable power supply source. In our applications, we use an *A.C. Ryan MobiliT* external USB battery at *3400mAh*, allowing *4* hours of autonomy for the system in complete functionality.

2.2 The Motion Sensor : Arduino

Arduino ([6]) is an open-source electronics prototyping platform based on flexible, easy-to-use hardware and software. *Arduino* can sense the environment by receiving input from a variety of sensors and can affect its surroundings by controlling lights, motors, and other actuators. The microcontroller on the board is programmed using the *Arduino* programming language and the *Arduino* development environment. The boards can be built by hand or purchased preassembled. The software can be downloaded for free. Although *Arduino* was built for artists and hobbysts, there are many people working on real electronic interactive projects, thanks to the rapid prototypization *Arduino* allows. We prototype a Bluetooth-based accelerometer using the *Arduino* board, an analogic *ADXL 345* accelerometer and a *BlueSMiRF Gold* Bluetooth modem.

2.3 The Development Side : OpenEmbedded + Ångström.

Openembedded (OE) offers a complete cross-compiler environment and allows developers to create complete Linux Distributions for embedded systems. OE offers different kernels for the BB. All kernels come with several patches and the support of the BB hardware is not perfect yet. Figure 5 summarizes the current status of hardware support, where it is possible to see that all the features of OMAP processors available in the BB are actually available for the use.

Boot	USB OTG	USB host	DVI	Audio out	Audio in	S-Video out	MMC / SD	RS232	Flash	DSP	SGX
2.6.27-r11	host only			Working on	NO	NO					
2.6.28-r12											

Figure 5: Current Status of Hardware Support for BB in OE, from [3].

The Linux Kernel *2.6.28r12* runs on our system. This particular Linux Kernel has *V4L2* (Video for Linux 2) drivers, allowing to plug in the system almost every Linux-compatible webcam. Furthermore, it contains *BlueZ*, the official Bluetooth protocol stack. Using this kernel version, several problems appear when using the DSP on the board. At the moment, all the problems related to DSP have been officially resolved in the Linux kernel version *2.6.29* but there exist concerning issues regarding the use of USB ports. For that reason, we use the "old" kernel release leaving aside, provisionally, the DSP functionalities.

The *Ångström Distribution* (AD) is the Linux Distribution running on the board. AD if a specific Linux distribution for embedded systems. A complete image of AD can be built using OE or with an online tool [9], where it is possible to choose the packages to be installed in the system. In the distribution we build, we include a toolchain for developing source codes on board. We install the *arm-gcc* compiler, *arm-g++* compiler and the *Python-Numpy* development environment. In addition, we build the GStreamer and the OpenCV packages, as we will explain in the next section.

2.4 The Software Side: GStreamer + OpenCV

GStreamer is a framework for creating streaming media applications. The GStreamer framework is designed to make easy writing applications handling audio/video streaming. Nevertheless, Gstreamer is not restricted to audio and video, and it can process any kind of data flow. One of the the most obvious uses of GStreamer consist in using it to build a media player. GStreamer already includes components for building a media player that can support a very wide variety of formats, including MP3, Ogg/Vorbis, MPEG-1/2, AVI, and more.

The main advantages of GStreamer are that the software components, called plugins, can be mixed and matched into arbitrary pipelines so that it is possible to write complete streaming data editing applications. Plugins can be linked and arranged in a pipeline. The GStreamer core function is to provide a framework for connecting plugins, for data flow management and for media type handling and negotiation. Using GStreamer, performing complex media manipulations becomes very easy and it integrates an extensive debugging and tracing mechanism. In BeaStreamer-v0.1, we use a pipeline for acquiring audio and video from webcam, with the possibility to encode the dataflow with the request quality and the resolution and the possibility to change the acquisition parameters at run time.

In addition, we compile on BeaStreamer-v0.1 the well-known *OpenCV* libraries and its Python bindings.

3 Experiments with BeaStreamer-v0.1

In this section, we show some experiment performed with BeaStreamer-v0.1 to demonstrate its capabilities. In Figure 6, we show six sequential photos taken wearing BeaStreamer-v0.1, walking in the street. Using GStreamer, we take photos with a framerate of one photo/second with a resolution of *320x240 pixels*, compressed in *jpeg* format. At the same time, in a separate thread, we record a countinous audio flow from the webcam microphone, sampled at *44100 samples/s* and compressed in *ogg* format. GStreamer allows setting online the parameters of acquisition making simple to change the resolution of photo and the encoding audio quality.

In order to get an estimate of the autonomy of the system, we record an audio/video stream compressed in *ogg* format and receive motion data from the bluetooth accelerometer. We are able to record up to *4 hours* of audio, video and motion data.

Using OpenCV, we setup a face detector running on photos adquired sequentially with GStreamer. The face detector can compute detections at a framerate of *5-10* frames/second depending of the images resolution, without using DSP. An example of faces successfully detected is shown in Figure 7. Using images with resolution of *80x60 pixels*, the face detector can scan the image in less than *100 ms* and detect faces in *200 ms*.

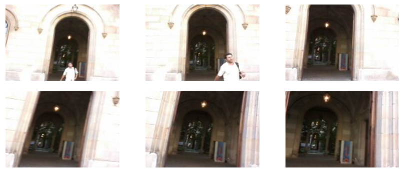


Figure 6: A sequence of photos taken with BeaStreamer-v0.1

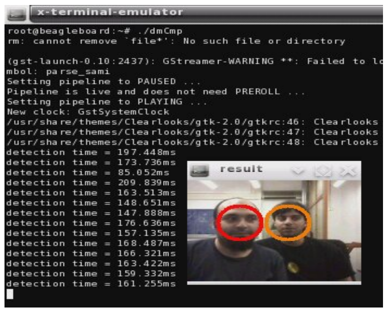


Figure 7: Face Detector running on BeaStreamer-v0.1.

Finally, in Figure 8 we show how BeaStreamer-v0.1 receives motion data. The acceleration analog values are converted in *10-bit* values from Arduino *ADC* (Analog-to-Digital Converter) at *40Hz*, stored in a buffer and sent every second via Bluetooth as UNICODE characters with a label showing the axis. BeaStreamer-v0.1 receives the data and stored them in a text file.

4 Conclusions and Future Works

In this paper we presented BeaStreamer-v0.1, a new platform for multi-sensors data acquisition. BeaStreamer-v0.1 is small and easy to bring, al-

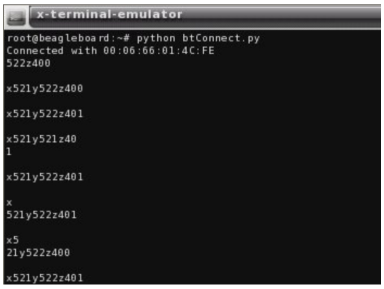


Figure 8: Acquisition of motion data.

lowing its use in wearable computing applications, for controlled and uncontrolled environments. We showed that different types of data can be easily acquired joining the potentiality of the Beagleboard and GStreamer.
The Beagleboard allows to connect different types of sensors communicating via Bluetooth or via the principal types of communication protocols implemented in the OMAP processor. GStreamer provide a framework to manage different types of data flows using a single and coherent environment.
At the moment, just a basic face detector has been developed on the system to demonstrate its capabilities. Furthermore, the computational power of the system will increase as soon as the DSP side of the OMAP will be completely operative.
Finally, we consider that unifying Bluetooth and general sensors acquisition under GStreamer will provide a powerful and complete platform for general multi-sensors data acquisition and analysis.

Acknowledge

This work is partially supported by a research grant from projects TIN2006-15308-C02, TIN2009-14404-C02, FIS-PI061290 and CONSOLIDER- INGENIO 2010 (CSD2007-00018), MI 1509/2005.

Special thanks to the “Computer Vision Group/DSPLab” of *Instituto di Fisiologia Clinica*, Consiglio Nazionale di Ricerche (CNR), Pisa, Italy for their help in the first stage of system development.

References

[1] Microsoft Research, *Introduction to SenseCam*, http://research.microsoft.com/en-us/um/cambridge/projects/sensecam/

[2] X. Ren, M. Philipose, ”Egocentric Recognition of Handled Objects: Benchmark and Analysis”, *Proc. of 1th Workshop on Egocentric Vision*, http://www.seattle.intel-research.net/egovision09/

[3] Beagleboard.org, *Beagleboard Technical Documentation*, http://beagleboard.org

[4] Home Brew Robotic Club, *A tutorial on setting up a system to do image processing with a BeagleBoard.*, http://www.hbrobotics.org/wiki

[5] Beaglebot, *Beagle powered robot* http://www.hervanta.com/stuff/Beaglebot

[6] Arduino.cc , *Arduino Technical Documentation*, http://www.arduino.cc

[7] Openembedded.org, *Open-Embedded Official Manual*, http://docs.openembedded.org/usermanual.html

[8] The Ångström Distribution, *Ångström Documentation*, http://linuxtogo.org/gowiki/Angstrom

[9] The Ångström Distribution, *Online Ångström Building*, http://www.angstrom-distribution.org/narcissus/

[10] GStreamer, *GStreamer Technical Documentation* , http://gstreamer.freedesktop.org/documentation/