Demo: Open-Source and Flexible Framework for Visual Sensor Networks

Luca Bondi, Luca Baroffio, Matteo Cesana, Alessandro Enrico Redondi, Marco Tagliasacchi
Dipartimento di Elettronica, Informazione e Bioingegneria
Politecnico di Milano
{name.surname}@polimi.it

ABSTRACT

We present an open-source and flexible framework for building VSN applications on top of low-cost and low-power linux-operated minicomputers. The framework comprises software modules for different types of nodes in the networks (camera, relays, cooperators and sink) in addition to a graphical user interface for controlling the network remotely. The great flexibility of the framework allows to easily implement applications scenarios characterized by different parameters, such as the wireless communication technology (e.g., 802.11, 802.15.4) or the type of data to be transmitted to the sink (image/video or feature-based data). To demonstrate the flexibility of the proposed framework, two representative applications are showcased: object recognition and parking lot monitoring.

Categories and Subject Descriptors

[**Computer systems organization**]: Embedded and cyber-physical systems—*Sensor networks*

Keywords

Visual Sensor Networks, Object Recognition, Parking Monitoring

1. INTRODUCTION

In the last few years, Visual Sensor Networks (VSNs) have gained a lot of attention within both the scientific community and the industry. Composed of many low-cost, battery-operated intelligent wireless cameras, VSNs are envisioned to play a major role in the evolution of the Internet-of-Things paradigm, supporting a vast range of applications such as object detection, recognition and tracking, video surveillance, assisted living, and many others.

VSNs are particularly challenging from the research point of view. On the one hand, they are used for multimedia applications, which are typically characterized by intense processing, high bandwidth use and energy consumption. On the other hand, being related to the more general Wireless Sensor Networks, VSNs have tight constraints in terms of computational, communication and energy resources. Thus, a huge part of the research on VSNs has been committed to finding efficient software solutions at each layer of the protocol stack, ranging from low-level communication protocols to multimedia compression schemes at the application layer. Unfortunately, many of these research works are tested and validated only through simulations, due to the lack of an easily usable, open-source and flexible hardware platform [1].

In this paper we propose a novel open-source platform tailored to easy, flexible and quick experimentation of software solutions for VSNs. On the hardware level, we base our design on top of a BeagleBone Black embedded computer operated by Linux. On the software level, we propose a flexible and efficient architecture based on widely accepted, open-source standards which provides all the necessary processing and transmission primitives needed for implementing a wide class of applications.

The framework is particularly useful to understand the differences between two operative paradigms: in the traditional compress-then-analyze (CTA) paradigm, camera nodes transmit images to the central controller, which are then analyzed through features extraction. Conversely, in the novel analyze-then-compress (ATC) paradigm, image features are extracted, compressed and transmitted by the camera nodes. This allows to perform analysis at a lower cost from both the bandwidth and energy point of view [2].

2. DESCRIPTION

2.1 Hardware Architecture

From the hardware point of view, we base our design on a BeagleBone Black Revision C embedded microcomputer, which features a 1 GHz ARM Cortex-A8 processor, 512 MB of RAM and 4 GB of eMMC flash memory and is capable of running Linux distributions based on the 3.8 kernel (e.g., Debian, Ubuntu and Android). To acquire images and videos, a BeagleBone Black can be equipped with any USB camera compatible with Video4Linux Version 2 (V4L2), or with acquisition devices created on purpose for the BeagleBone Black, such as the ultra low-power RadiumBoard HD Camera Cape. Similarly, several options are available to equip the camera node with wireless communication capabilities. Those applications which require high bandwidth may leverage wireless communication based on the IEEE 802.11 WiFi standard: this can be obtained by connecting a low-power USB WiFi adapter such as the Netgear WNA1100 or the Ed-

ICDSC 2015 Sevilla, Spain

ACM ISBN 978-1-4503-3681-9/15/09.
DOI: http://dx.doi.org/10.1145/2789116.2802650

imax EW-7811Un. For those applications where bandwidth can be traded off for energy, an IEEE 802.15.4-compliant device can be used to achieve low-power wireless communication (we use a MEMSIC TelosB node, which can be easily attached to the BeagleBone's USB port).

2.2 System Architecture

The proposed framework enables the creation of different nodes in a Visual Sensor Network. The different types of nodes that can be created are listed in the following:

- `camera`: this type of nodes play the major role in the VSN. Based on BeagleBone Black platforms equipped with cameras and 802.15.4 or 802.11 wireless communication devices, such nodes acquire image data from the environment and deliver it remotely to a central controller. Multiple cameras can be used simultaneously.

- `relay`: data delivery from camera nodes to the central controller can be performed either directly, or in a multi hop fashion. In the latter case, visual data is routed through one or more relay nodes. Depending on the technology chosen for communication (e.g., IEEE 802.15.4 or IEEE 802.11), `relay` nodes are created by using a TinyOS-operated TelosB node alone, or a BeagleBone Black with an attached WiFi dongle, respectively.

- `sink`: this node is responsible for collecting data coming from cameras and forwarding it to the central controller, where it can be displayed, stored or analysed. Depending on the particular application, the `sink` node can be constituted by either a BeagleBone Black or a standard PC/laptop if available.

- `cooperator`: the proposed framework also provides the possibility of adding a special type of nodes in the VSN, known as cooperators. Such nodes may not be equipped with a camera, but can be still added to the network to sense different kind of data (e.g., temperature, humidity), or as backup nodes. Their purpose is to help camera nodes in executing intense processing tasks, leveraging recent results in the area of parallel computation for VSN [3].

Each node in the VSN runs the same software, which behaves differently depending on the role of the node in the network.

2.3 Central Controller and GUI

The central controller implements all the logic needed to control the VSN. In particular, it features a Graphical User Interface (GUI) that allows to display and analyze the information received from camera nodes, remotely modify the VSN operational parameters (e.g., the communication technology) and monitor the performance of the VSN in real-time (see Figure 1).

3. APPLICATION SCENARIOS

To demonstrate the flexibility of the framework, two different application scenarios are implemented and showcased: (i) object recognition and (ii) parking monitoring. In the former, a user can use the VSN to recognize different real objects. In the latter, images from a parking lot are used to count and detect the number of vacant spaces. For both the scenarios, a user can change in real-time several operational parameters and obtain an immediate feedback on the VSN performance. As an example, when CTA is selected, a user can decrease (increase) the JPEG compression quality factor. This allows on the one hand to increase (decrease) the system frame rate, on the other hand to decrease (increase) the visual analysis performance. If ATC is selected, several parameters can be changed (e.g., number of features to be transmitted, type of compression to be performed on the features, etc...) [4]. In general, ATC allows to obtain the best performance in terms of energy consumption and system frame rate, without affecting the visual analysis accuracy.

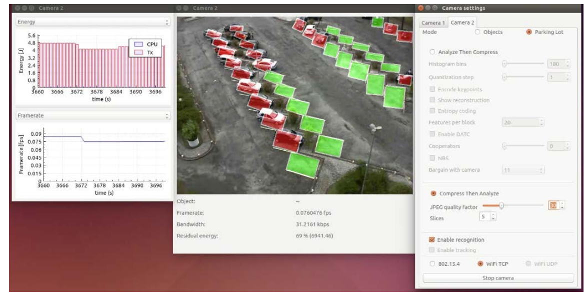

Figure 1: The GUI used to control the VSN for the case of parking monitoring. From left to right: the real-time performance monitoring panel, the camera view and the control panel.

4. CONCLUSIONS

This demo showcases an open-source and flexible framework for building VSN applications. It provides software for different types of nodes in the network, together with a graphical user interface for viewing the data coming from the VSN and monitoring its performance in real time. A detailed video describing the demonstrator is available at www.greeneyesproject.eu.

5. ACKNOWLEDGMENTS

The project GreenEyes acknowledges the financial support of the Future and Emerging Technologies (FET) programme within the Seventh Framework Programme for Research of the European Commission, under FET-Open grant number:296676.

6. REFERENCES

[1] I. Akyildiz, T. Melodia, and K. Chowdhury, "Wireless multimedia sensor networks: Applications and testbeds," *Proceedings of the IEEE*, vol. 96, no. 10, pp. 1588–1605, Oct 2008.

[2] A. Redondi, L. Baroffio, M. Cesana, and M. Tagliasacchi, "Compress-then-analyze vs. analyze-then-compress: Two paradigms for image analysis in visual sensor networks," in *Multimedia Signal Processing (MMSP), 2013 IEEE 15th International Workshop on*, Sept 2013, pp. 278–282.

[3] A. Redondi, M. Cesana, M. Tagliasacchi, I. Filippini, G. Dan, and V. Fodor, "Cooperative image analysis in visual sensor networks," *Ad Hoc Networks*, vol. 28, no. 0, pp. 38 – 51, 2015.

[4] A. Redondi, L. Baroffio, J. Ascenso, M. Cesana, and M. Tagliasacchi, "Rate-accuracy optimization of binary descriptors," in *Image Processing (ICIP), 2013 20th IEEE International Conference on*, Sept 2013, pp. 2910–2914.