Ad Hoc Networks 28 (2015) 38–51

Contents lists available at ScienceDirect

Ad Hoc Networks

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

Cooperative image analysis in visual sensor networks

Alessandro Redondi [a,*], Matteo Cesana [a], Marco Tagliasacchi [a], Ilario Filippini [a], György Dán [b], Viktoria Fodor [b]

[a] Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, P.zza Leonardo da Vinci, 32, Milano, Italy
[b] Lab for Communication Networks, School of Electrical Engineering, KTH Royal Institute of Technology, Stockholm, Sweden

ARTICLE INFO

Article history:
Received 16 July 2014
Received in revised form 28 October 2014
Accepted 19 January 2015
Available online 30 January 2015

Keywords:
Visual sensor network
Distributed processing
Divisible load theory

ABSTRACT

This work addresses the problem of enabling resource-constrained sensor nodes to perform visual analysis tasks. The focus is on visual analysis tasks that require the extraction of local visual features, which form a succinct and distinctive representation of the visual content of still images or videos. The extracted features are then matched against a feature data set to support applications such as object recognition, face recognition and image retrieval. Motivated by the fact that the processing burden imposed by common algorithms for feature extraction may be prohibitive for a single, resource-constrained sensor node, this paper proposes cooperative schemes to minimize the processing time of the feature extraction algorithms by offloading the visual processing task to neighboring sensor nodes. The optimal offloading strategy is formally characterized under different networking and communication paradigms. The performance of the proposed offloading schemes is evaluated using simulations and is validated through experiments carried out on a real wireless sensor network testbed. The results show that the proposed offloading schemes allow to reduce the feature extraction time up to a factor of 3 in the reference scenario.

1. Introduction

Visual sensor networks (VSNs) extend the application fields of traditional wireless sensor networks by adding the capability to acquire and process multimedia signals such as still images and video. VSNs can have a significant impact in scenarios in which visual analysis is currently infeasible, due to the mismatch between the transmission and computational resources and the complexity of the analysis tasks. As an example, in the context of smart cities, the availability of inexpensive visual nodes can enable a much more complete coverage of the urban landscape, reaching a wider area and limiting the costs of the required infrastructure to support applications for traffic monitoring, smart parking metering, environmental monitoring, hazardous situations monitoring, etc. [1,2].

Classical networked systems for visual analysis follow the *compress-then-analyze* paradigm, where image/video analysis is performed last and is decoupled from the acquisition, compression and transmission phases. In the case of VSNs, powerful smart cameras are substituted by vision-enabled, battery-operated sensing nodes with low-power microprocessors and radio chips. The traditional compress-send-then-analyze paradigm may not fit well the computation and communication-related constraints imposed by these VSNs, as it requires large communication resources. For this reason, an alternative paradigm, named *analyze-then-compress* has been recently proposed leveraging the idea that most visual analysis tasks can be carried out based only on a succinct representation of the image [3]; that is, image features can be collected by sensing nodes, processed, and then delivered to the final

---

* Corresponding author. Tel.: +39 02 2399 9614; fax: +39 02 2399 3413.
E-mail address: redondi@elet.polimi.it (A. Redondi).

http://dx.doi.org/10.1016/j.adhoc.2015.01.008
1570-8705/© 2015 Elsevier B.V. All rights reserved.

destination(s), thus avoiding the need for encoding and transmitting redundant pixel-level representations.

Extracting features from visual data is however a computationally intensive task. For the case of local features, the process entails detecting image keypoints and computing the corresponding descriptors, a process whose computational complexity grows linearly with the image size and with the number of scales, that is, the downsampled versions of the image that are used in the feature extraction process. As an example, even when using feature extraction algorithms tailored to low-power architectures (e.g. BRISK [4]), the processing time for detecting keypoints and generating the corresponding descriptors can be as high as 5*s* on an Intel Imote2 sensor platform [5], and in the order of 0.5–1 s when using a BeagleBone [6] based sensor node, as illustrated in Fig. 2.

Such processing time may not be low enough when the outcome of the visual processing task is used to trigger control actions or to carry out analysis tasks such as event detection and object tracking [7,8]. This observation calls for alternative computing strategies, that are able to reduce the processing time of a single frame in the VSN.

Different from classical networks of smart cameras where the camera itself is a stand-alone entity which does all the required processing, in the VSN scenario the camera node is most likely to be close to other wireless nodes, equipped with cameras or other kind of sensors. Such nodes may be responsible for the acquisition and processing of both visual and non-visual data (e.g., temperature, pressure, infrared radiation, etc.), and are characterized by specific processing and transmission capabilities. In such a scenario, proximity can potentially allow a camera node to leverage the resources of the neighboring sensor nodes to reduce the overall processing time of the visual processing task, following a distributed computing approach [1]. The main rationale is to let the camera node "borrow" processing power from the neighboring nodes to process part of the initial load (image), further exploiting the parallelization of the visual processing task. Ideally, parallelizing a processing task always leads to a lower processing time. However, in VSNs parallelization requires additional communication overhead, and therefore its optimization is not straightforward.

In this work, we target the minimization of the processing time, taking into account the limitations of the computational and communication capabilities of the sensors nodes and the specific requirements of the visual feature extraction. To optimize the distributed processing, we need to answer the following questions: (i) what communication paradigm, unicast or broadcast, is best suited for distributing the loads among the cooperators in VSNs, (ii) how many cooperators should be utilized and what is the load share each cooperator should get, (iii) what is the trade-off between processing time and energy consumption involved in the offloading process, and finally, and (iv) what is the impact of different visual contents on the overall processing time.

We formulate the problem of minimizing the processing time in the framework of Divisible Load Theory (DLT), which has been widely used to study how processing load can be optimally divided within processor grids [9]. We derive closed form expressions for the optimal task processing time and for the corresponding load assignment to the cooperating sensor nodes under different networking scenarios and communication paradigms, taking into consideration the specific properties of feature extraction algorithms. The performance of the proposed load distribution schemes are evaluated both through simulation and through the development of a real-life Visual Sensor Network composed of one camera node and several cooperators.

The rest of the paper is organized as follows. Section 2 discusses the related work further providing background on the divisible load theory. In Section 3 the main ideas and algorithms behind visual feature extraction are described; Section 4 defines the reference scenario and the problem statement. In Section 5 the DLT-based offloading framework for visual sensor networks is presented, and Section 6 reports the performance evaluation study of the proposed framework in realistic visual sensor networks. Section 7 concludes the paper.

2. Related work

The challenge of networked visual analysis under the *compress-then-analyze* paradigm is to minimize the amount of pixel data to be transmitted. As lossy coding in the pixel domain affects the visual analysis at the central node, recent works propose image coding schemes that are optimized for feature extraction [10,11].

If the *analyze-then-compress* approach is followed, the main challenge is to minimize the computational load at the camera node, and therefore many of the emerging feature extraction schemes aim at decreasing the computational complexity [12,4]. If the transmission bandwidth of the VSN is very limited, the objective can even be to limit the amount of data to be transmitted by compressing the descriptors or by limiting their number. Descriptor compression techniques are suggested and evaluated in [13,14]. In [15] a progressive transmission scheme is proposed, which terminates the transmission of new descriptors as soon as the image is retrieved. In [16] the number of considered interest points and the quantization level of the descriptors are jointly optimized to maximize the accuracy of the recognition, subject to energy and bandwidth constraints.

The present work is motivated by recent results on the expected transmission and processing load of visual analysis in sensor networks [5], demonstrating that processing at the camera or at the sink node of the VSN leads to significant delays, and thus distributed processing is necessary for real-time applications. To analyze the performance of distributed processing we leverage tools from the divisible load theory (DLT), which addresses the problem of load distribution in a grid of processors. DLT has been applied for different distributed systems [9], and recently also to wireless sensor networks [17,18], for processing large data whose computation may be split among different entities (i.e., in case the load is divisible).

Distributed processing in networks typically considers a central processor, which is the source of the processing