Robotics and Autonomous Systems 68 (2015) 86–103

Contents lists available at ScienceDirect

Robotics and Autonomous Systems

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

A real-time data acquisition and processing framework for large-scale robot skin ☆

S. Youssefi *, S. Denei, F. Mastrogiovanni, G. Cannata

Department of Informatics, Bioengineering, Robotics and Systems Engineering, University of Genoa, Via Opera Pia 13, 16145, Genoa, Italy

HIGHLIGHTS

- Hard real-time data acquisition and distribution algorithms.
- Modular and scalable design which ensures data coherency and consistency.
- Provides skin technology abstraction, supporting heterogeneous technologies.
- Enables portability of skin-based algorithms among different technologies.
- Extensive validation and performance tests are performed.

ARTICLE INFO

Article history:
Received 15 May 2014
Received in revised form
9 January 2015
Accepted 19 January 2015
Available online 9 February 2015

Keywords:
Robot skin
Tactile
Middleware
Skinware
Software
Framework

ABSTRACT

Large-scale tactile sensing applications in Robotics have become the focus of extensive research activities in the past few years, specifically for humanoid platforms. Research *products* include a variety of fundamentally different robot *skin* systems. Differences rely in technological (e.g., sensory modes and networking), system-level (e.g., modularity and scalability) and representation (e.g., data structures, coherency and access efficiency) aspects. However, differences within the same robot platform may be present as well. Different robot body parts (e.g., fingertips, forearms and a torso) may be endowed with robot skin that is tailored to meet specific design goals, which leads to local peculiarities as far as technological, system-level and representation solutions are concerned. This variety leads to the issue of designing a software framework able to: (i) provide a unified interface to access information originating from heterogeneous robot skin systems; (ii) assure portability among different robot skin solutions. In this article, a real-time framework designed to address both these issues is discussed. The presented framework, which is referred to as *Skinware*, is able to acquire large-scale tactile data from heterogeneous networks in real-time and to provide tactile information using *abstract* data structures for high-level robot behaviours. As a result, tactile-based robot behaviours can be implemented independently of the actual robot skin hardware and body-part-specific features. An extensive validation campaign has been carried out to investigate Skinware's capabilities with respect to real-time requirements, data coherency and data consistency when large-scale tactile information is needed.

1. Introduction

The sense of touch is essential for human beings [1] and has been shown to be of the utmost importance in Robotics for manipulation tasks [2,3], safety issues [4], and social behaviours [5]. Tactile sensors have been the subject of extensive research for three decades [6,7]. However, only recent activities in physical human–robot interaction [8] have stressed the importance of using large-scale tactile systems, specifically in the form of robot *skin* [9].

Focusing on robot skin systems, possibly covering large surfaces of a robot body, different solutions as far as technological, system-level and data representation aspects are concerned, have been investigated. From a technological perspective, transduction, distributed communication as well as networking issues must be considered [9]. At the system-level design, novel issues arise when adopting large-scale robot skin. Such issues include scalability and modularity [10], conformance to robot surfaces [11,12], as well as how to scale wiring patterns [13] and embedded communication networks [14,15], just to name but few.

According to these design principles, many examples of robot skin systems have been presented during the past few years, either based on modular designs [11,12,16,17], or on non-modular but

☆ A preliminary version of the work discussed in this article has been presented in Youssefi et al. (2011) [70].

* Corresponding author.
E-mail addresses: shahbaz.youssefi@unige.it (S. Youssefi), simone.denei@unige.it (S. Denei), fulvio.mastrogiovanni@unige.it (F. Mastrogiovanni), giorgio.cannata@unige.it (G. Cannata).

http://dx.doi.org/10.1016/j.robot.2015.01.009
0921-8890/

flexible and highly conformable sensor arrays [18–25]. Besides the adopted sensing mode [9], these robot skins have adopted both industrial and *ad hoc* solutions in actual robots, such as VMEbus [18], PCI bus [26], CAN bus [19,12,16], SMBus [11], EtherCAT [15], UART bus [17] or other methods [20–24,27,25].

However, at the representation level, it is necessary to identify strategies for data acquisition and processing, which must be independent from both technological and system-level solutions. Furthermore, huge amounts of tactile data originating from distributed sources are of limited use if not processed in time and according to well-defined contact models [28]. Tactile data representation and processing is the basis to implement tactile-based robot behaviours, e.g., exploiting local information related to the distribution of force exerted on the robot surface. At a first glance, it may be argued that tactile data processing can be tackled by extending established *computer vision* techniques. However, there is no shortage of reasons to conclude that these two scenarios are significantly different.

- Tactile elements (i.e., the discrete sensing locations of robot skin systems, henceforth referred to as *taxels*) are distributed over robot body parts with varying shape and curvature. As a consequence, taxel locations usually do not form a regular square grid. This renders the concept of a *bitmap* not directly applicable, as it assumes data to be organized on a regular and well-defined 2D grid. On the contrary, *tactile images* are hardly structured [29,30] in a regular arrangement.
- Tactile images depend on specific robot–environment configurations [31]. Contacts may arise as a consequence of robot postures originating from control actions. Moreover, such interaction phenomena as robot *self-touch* need to be considered, where the tactile image strictly depends on the robot configuration only.
- Different robot body parts may demand different requirements from a robot skin system in terms of tactile sensor density, resolution and sensitivity. Thus, tactile images need to be processed taking into account the peculiar characteristics of the body area originating tactile data.
- In analogy with biological skin, robot skin may be designed to embed different types of transducers, e.g., temperature or vibration [17], whose values may be correlated over time and space [32]. At a representation level, different tactile images referring to different sensing modes may need to be jointly processed, e.g., to better describe the contact event [17].

In order to properly manage the acquisition and the efficient processing of tactile data, the need for a robot skin *middleware* arises, i.e., a software framework capable of providing (i) real-time data access strategies such as *best effort*, *periodic* and *event-based*, and (ii) data abstraction features for tactile-based robot behaviours, specifically with respect to the robot skin hardware and mechanical implementation, the robot kinematic configuration and the specific robot body part originating tactile data.

The focus of this article is to propose such a software framework, providing a hardware abstraction layer for tactile-based robot behaviours, able to guarantee a consistent, common interface towards different technological and system-level solutions. To the best of the authors' knowledge, the need for such a software framework is solely acknowledged in [33], where a method for organizing tactile data originating from a sensory body suit is presented.

The development of software frameworks for Robotics is topic of active research. Relevant examples of such frameworks include both non real-time and real-time systems. The first class includes, in chronological order, Ethnos/SeART [34–36], OROCOS [37], CLARAty [38], Player/Stage [39,40], Orca [41], RT-Middleware [42], YARA [43], YARP [44] and ROS [45]. The second class includes OpenDDS (an open source implementation of DDS [46]),

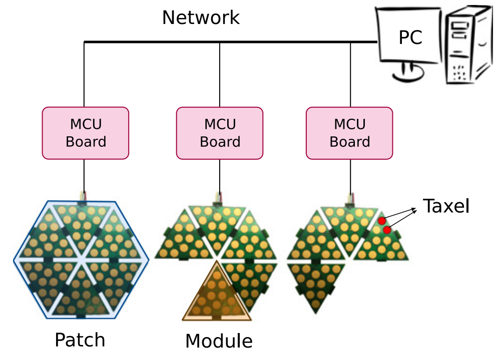

Fig. 1. Taxels, modules, patches and the network in the considered robot skin technology.

MiRPA [47] and RDDS [48]. However, these frameworks provide general-purpose mechanisms to design and implement concurrent and distributed software robot components. On the contrary, the proposed software framework as a whole, which will be henceforth referred to as *Skinware*, can be used as a specialized component in any of these frameworks, none of which provides any comparable alternative. It is noteworthy that although the same term *middleware* is used to both refer to the proposed software framework and the aforementioned Robotics software frameworks, Skinware is placed at a layer closer to hardware than that of those frameworks and does not try to provide an alternative to them.

The main contribution of the article is two-fold. On the one hand, the definition of a software architecture able to manage large-scale robot skin systems in real-time and to provide users with data structures and communication protocols which are independent from the actual robot skin hardware and robot configuration. On the other hand, a detailed account of the choices that have been made to implement and validate the defined software framework in real-world scenarios. The implementation of Skinware is available online[1] as free software. A preliminary version of the work discussed in this article has been presented in Youssefi et al. (2011) [70].

The article is organized as follows. Section 2 presents the terminology associated with the Skinware software framework. Section 3 gives an overview of the conceptual architecture of Skinware as well as the implemented structure. The main features of Skinware are discussed in Section 4. Relevant algorithms are discussed in Section 5. Tests performed on Skinware are presented in Section 6. Section 7 contains further discussions and future directions. Conclusion follows.

2. Terminology

2.1. Robot skin hardware

In this article, we will adopt as a reference, the robot skin technology proposed in [12] (see Fig. 1) which was further developed in [10], and finally tested on several various robots [49]. This kind of robot skin system is an example of modular and scalable design, which is aimed at covering large-scale robot surfaces. Since mostly all the available modular and scalable designs for robot skin systems discussed in the literature share many similarities, it is noteworthy that application of the concepts discussed in this article to other skin designs is straightforward. Nonetheless, few examples are provided following the definitions.

The typical robot skin system based on modular and scalable design choices [11,12,16,17] is hierarchically organized on the basis of the following elements.

Taxel. A single tactile transducer, i.e., a discrete sensing point on the robot surface.

[1] https://github.com/maclab/skinware/.