Available online at www.sciencedirect.com

ScienceDirect

IFAC-PapersOnLine 49-19 (2016) 277–282

Smartwatch-Enhanced Interaction with an Advanced Troubleshooting System for Industrial Machines

Valeria Villani, Lorenzo Sabattini, Nicola Battilani, Cesare Fantuzzi

Department of Sciences and Methods for Engineering (DISMI), University of Modena and Reggio Emilia, Reggio Emilia, Italy, (e-mail name.surname@unimore.it)

Abstract: Smartwatches are unobtrusive everyday devices which can be also exploited for effective gesture-based human-machine interaction. In this paper, we propose the use of a smartwatch to interact with an advanced troubleshooting application to be used in industrial environment. The application is a hypermedia information system aiming at assisting workers in performing preventive and corrective machine maintenance. The smartwatch allows a hands-free interaction, thus facilitating the use of the whole system when wearing personal protective equipment such as gloves or having fingers greased with oil or dust, which prevent operating touch screens. The algorithm for gesture recognition we have devised, which is based on template matching, is described in the paper, together with its experimental validation. Finally, we present a preliminary usability assessment of the overall system, meant as integration of the smartwatch with the hypermedia system.

Keywords: Human-machine interaction; unobtrusive interaction; gesture recognition; smartwatch; troubleshooting; hypermedia information system.

1. INTRODUCTION

Human-computer interaction has been widely studied in the literature (Sears and Jacko, 2008). A very natural and intuitive way, for a human, to interact with a machine is represented by *gestures*. Gesture-based interfaces, which let users control applications on nearby computers with, for example, hand or finger motions, are becoming increasingly popular (Garber, 2013). Such interfaces exploit gesture-recognition algorithms to identify body movements. The systems then determine which device command a particular gesture represents and take the appropriate action. For example, moving a hand sideways might mean that a user wants to turn a page on an e-reader screen.

The main advantage of gesture-based interfaces over traditional interfaces is that they allow communicating with a machine in a natural manner, without a mouse or other intermediate device, which serves as a barrier between users and technology (Garber, 2013). On the contrary, the natural communication of people includes body language, and gesture-based interfaces let users make normal real-life motions, such as flipping a page, to perform similar functions on a device (Garber, 2013). Hence, people can easily use gestures to perform simple actions and interact with devices without having to slow down.

Several approaches can be found in the literature, on the use of gestures for human-machine interaction (Mitra and Acharya, 2007; Sears and Jacko, 2008). Such methods can be generally classified into two classes (Moeslund and Granum, 2001; Sears and Jacko, 2008; Porzi et al., 2013): vision-based approaches, which rely on one or more cameras, either stationary or wearable, to detect and analyze body movement from the video sequences, and non-vision-based approaches, which utilize devices (e.g., sensorized gloves, accelerometers, touch screens, joysticks, etc.) that are physically in contact with the user.

A comprehensive survey of *vision-based* human motion capture literature can be found in (Moeslund and Granum, 2001). Generally speaking, vision-based techniques require controlled and uniform lighting conditions and proper camera angles (Raheja et al., 2012; Porzi et al., 2013). Additionally, these solutions suffer from limited wearability, high computation and energy requirement, thus being not ideal in real mobile environments (Raffa et al., 2010). Also, objects in the foreground or background of a scene may block the target or make recognition of it more difficult.

To overcome these limitations, physical interaction can be exploited for implementing *non-vision-based* methodologies. One of the most popular techniques consists of recognizing gestures utilizing sensorized gloves. In particular, sensorized gloves are used to directly measure hand or arm joint angles (Junker et al., 2008). While these devices provide a high measurement precision, they heavily limit the freedom of motion of the user, who is forced to wear uncomfortable devices (Raheja et al., 2012). Moreover, non-vision-based gesture recognition can be obtained utilizing touch screen technologies. Along these lines, (Evans et al., 2012; Micire et al., 2009) consider the use of multi-touch hand-held technologies such as tablets to control robots. Although very intuitive, these interfaces require the use of both hands and, hence, are not practical in many

2405-8963 © 2016, IFAC (International Federation of Automatic Control) Hosting by Elsevier Ltd. All rights reserved.
Peer review under responsibility of International Federation of Automatic Control.
10.1016/j.ifacol.2016.10.547

circumstances. To overcome this issue, techniques have been developed that rely on hand-worn accelerometer and gyroscope data (Porzi et al., 2013; Akl and Valaee, 2010; Khan et al., 2012; Liu et al., 2009). These sensors have the advantage of being low cost, low power and extremely compact and have been successfully used in mobile and pervasive computing (Raffa et al., 2010). In this regard, modern smartwatches are a good solution, since they are versatile everyday devices allowing a non-obtrusive interaction with surrounding environment. Previous works (Bieber et al., 2012; Porzi et al., 2013; Raffa et al., 2010) have shown that, equipped with accelerometers and a vibration feedback, they can be successfully used for building a simple user interaction system based on gesture recognition.

In this paper, we propose the use of a smartwatch to control an automatic machine in industrial environment. In particular, we have devised a hypermedia information system (Setchi and White, 2003; Crowder et al., 2003; Greenhough and Tjahjono, 2007; Fakun and Greenhough, 2002) assisting shopfloor workers in performing machine maintenance. The application, which we called *MyAID*, aims at guiding the user to: i) identify the fault, i.e., understand which subpart of the machine is not working properly; ii) solve the fault, once it has been detected (Villani et al., 2016). The tool runs on a tablet or a panel PC and requires worker's interaction through an input device, such as a touch screen or a mouse. However, in typical scenarios this is impractical and a hands-free interaction is required for two distinct reasons. First, workers are required to wear protection gloves as personal protective equipment and removing gloves each time they need to input the application would be very annoying for them. Second, in industrial environment, fingers of workers are usually greased with oil or dust, which prevent operating touch screens. We tackle this issue allowing the worker to control the troubleshooting application via a smartwatch, thus implementing a hands-free gesture-based interaction. The system has been devised in the framework of a collaboration between the university and a top-level multinational beverage company.

The paper is organized as follows. First, we present the overall system and describe the interactive troubleshooting tool and the gesture recognition algorithm. Then, we summarize the preliminary experimental validation of the proposed system. Finally, we point out some concluding remarks.

2. PROPOSED SYSTEM

2.1 Interactive troubleshooting

MyAID is an interactive troubleshooting system embodying a guided procedure for the identification of machine failures and their resolution. The possible failures occurring in the machine and the step-by-step instructions to solve them were firstly listed by the machine's manufacturer. These have been ordered from the less specific (e.g., dirty parts requiring some cleaning) to the most specific (e.g., involving internal parts of the machine or requiring parts substitution) and a decision tree was derived, covering all the possible failures. Thus, *MyAID* guides the user through the decision tree to isolate the current failure.

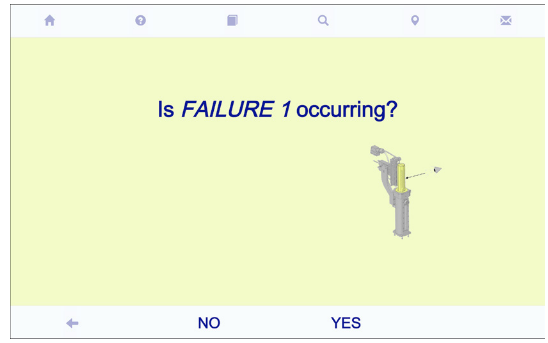

Fig. 1. Example of *check* screen, where the user is asked whether *FAILURE 1* is occurring.

Fig. 2. *Fix* screen showing the instructions to solve *FAILURE 1*.

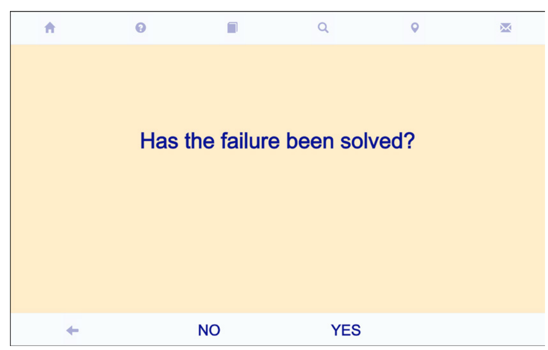

Fig. 3. *Confirmation* screen, where the user is asked if the failure has been solved by the procedure just accomplished.

The interface includes three different kinds of screen: i) *check*, i.e., failure identification , ii) *fix*, i.e., instructions to solve a failure, and iii) *confirmation*. In *check* screens appointed for failure identification, the worker is asked whether she/he is facing the failure that is currently shown on the screen. An example is provided in Fig. 1. In the case of negative answer, the following possible failure is prompted. Otherwise, in the case of positive answer, i.e., the user claims that the prompted failure is occurring, the corresponding *fix* screen is shown, reporting the instruction to solve that failure. An example of *fix* screen is reported in Fig. 2. At the end of the procedure, the user, via the *confirmation* screen, is asked whether the issue has been fixed, i.e., the machine has recovered its normal function, or not, as shown in Fig. 3. In the former