Entertainment Computing 4 (2013) 179–186

Contents lists available at SciVerse ScienceDirect

Entertainment Computing

journal homepage: ees.elsevier.com/entcom

A Kinect-based natural interface for quadrotor control

Andrea Sanna *, Fabrizio Lamberti, Gianluca Paravati, Federico Manuri

Dipartimento di Automatica e Informatica, Politecnico di Torino, corso Duca degli Abruzzi 24, I-10129 Torino, Italy

ARTICLE INFO

Article history:
Received 27 March 2012
Revised 7 September 2012
Accepted 20 January 2013
Available online 30 January 2013

Keywords:
Natural user interface
Kinect
Quadrotor control
Interactive systems
Visual odometry

ABSTRACT

This paper presents a new and challenging approach to the control of mobile platforms. Natural user interfaces (NUIs) and visual computing techniques are used to control the navigation of a quadrotor in GPS-denied indoor environments. A visual odometry algorithm allows the platform to autonomously navigate the environment, whereas the user can control complex manoeuvres by gestures and body postures. This approach makes the human–computer interaction (HCI) more intuitive, usable, and receptive to the user's needs: in other words, more user-friendly and, why not, fun. The NUI presented in this paper is based on the Microsoft Kinect and users can customize the association among gestures/postures and platform commands, thus choosing the more intuitive and effective interface.

1. Introduction

The control of mobile platforms plays a key role in several application fields ranging from surveillance to entertainment. A framework to control a quadrotor is presented in this paper; the proposed solution supports both autonomous flight and manual control by user's body postures. The autonomous flight system has been designed for GPS-denied indoor environments, whereas the human–computer interaction (HCI) has been based on gestures/postures, thus implementing a so called natural user interface (NUI). NUIs have been investigated since early eighty's (voice and gestures are used to control a GUI in [5]). Among NUIs, gesture-based interfaces always played a crucial role in human–machine communication, as they constitute a direct expression of mental concepts [25]. For example, nowadays mobile platforms can be remotely piloted by using multi-touch devices [22,23] that also act as display devices through the use of interactive streaming functionalities [18]. The analysis of images coming from on-board cameras allows mobile platforms to perform target tracking and following tasks [19–21]. The variety of hand and body gestures, compared with traditional interaction paradigms, can offer unique opportunities also for new and attractive forms of HCI [24]. Thus, new gesture-based solutions have been progressively introduced in various interaction scenarios (encompassing, for instance, navigation of virtual worlds, browsing of multimedia contents, management of immersive applications, etc. [28,39]) and the design of gesture-based systems will play an important role in the future trends of the HCI.

Human–robot interaction (HRI) is a subset of HCI and can be considered as one of the most important domains of the computer vision. Although a lot of works based on gesture recognition in the domain of HRI are known in the literature (Section 2 briefly reviews the most appropriate ones) recent technological advances have opened new and challenging research horizons. In particular, controllers and sensors used for home entertainment can be exploited also as affordable devices supporting the design and implementation of new kinds of HRI.

The aim of this work is to create a human–robot interaction framework to allow a quadrotor both to perform autonomous navigation tasks (by completing path-following missions constituted by a sequence of pre-specified way-points) and to be controlled by user's body postures (for instance, when complex actions/movements have to be performed). The main requisites needed to implement a system capable of controlling the aerial vehicle by means of user's posture are: (1) extracting spatial information from specific parts of the body (2) recognizing postures from this information (3) associating recognized postures to specific commands to be sent the quadrotor. In this work, the Microsoft Kinect [15] is used as gesture tracking device; recognized postures are then used to control an Ar.Drone quadrotor platform [2] (in the following of the paper the terms: Ar.Drone, quadrotor, and platform will be used interchangeably). The user is the "controller", and a new form of HRI can therefore be experienced. Interaction with quadrotors via Microsoft Kinect is not new. In particular, the ETH Zurich group proposes a way to dynamically interact with quadrotors based on the position of arms [10]. The local 3D coordinates of the user's arms are mapped to the local 3D coordinates of the quadrotor; in this way, a direct mapping between arms coordinates and quadrotor's position can be established. However, the

---

* Corresponding author. Tel.: +39 011 090 7035; fax: +39 011 090 7099.
E-mail address: andrea.sanna@polito.it (A. Sanna).

1875-9521/$ - see front matter © 2013 International Federation for Information Processing Published by Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.entcom.2013.01.001

approach adopted in this paper is different from the one adopted in [10]. Indeed, in this paper gesture recognition is used to trigger discrete control commands without using an external tracker system to obtain the quadrotor position. The two methods can be considered as complementary, since the approach presented in this paper is useful for the navigation over a path which length is not known a priori, whereas the ETH Zurich group approach [10] is useful for performing local navigation tasks where a higher degree of precision is needed. Tests proved that the platform can be easily controlled by a customizable set of body movements, allowing for an exciting, fun, and safe experience even for non-skilled users. In order to allow the platform to autonomously fly indoor environments, a pose estimation system exploiting two different techniques is able to process images received from an on-board camera to support the navigation. In this work, the on-board camera is looking downward. Position and flight altitude are continuously measured by a feature-based pose estimation algorithm that analyzes the image features of the camera view, thus allowing the system to estimate the location and the orientation of the platform in the environment. Moreover, a second technique (namely tag-based pose estimation) exploits some visual markers (tags) placed on the floor, at well known positions, in order to reset localization drifts cumulated by the feature-based pose estimation system [7]. The combination of these two techniques results into an efficient and robust visual odometry algorithm. The overall framework introduces a number of challenges to be addressed, from the control of the quadrotor through body postures to the analysis of the images coming from the on-board camera to infer the flight attributes of the quadrotor in a GPS-denied environment. The main contribution of this paper is the proposal of a new integrated framework able to efficiently and intuitively support both autonomous and piloted flight in indoor environments.

The paper is organized as follows: Section 2 reviews the main HRI solutions and briefly introduces the Ar.Drone. Section 3 describes the NUI and the mapping between gestures and commands. Section 4 presents the quadrotor pose estimation algorithm and its performance. Finally, conclusions are drawn in Section 5.

2. Background

This Section is split in two parts: the first part presents the state-of-the-art of NUIs with a particular focus on HRI, whereas the second part describes the quadrotor used for tests.

2.1. Natural user interfaces

In HRI-based systems, especially in safe critical applications such as the search-and-rescue and military ones, it is increasingly necessary for humans to be able to communicate and control robots in a natural and efficient way. In the past, robots were controlled by human operators using hand-controllers such as sensor gloves and electromechanical devices [32]. With these devices, the speed and simplicity of the interaction were significantly constrained. To overcome the limitations of such electro-mechanical devices, gesture and body posture recognition techniques have been introduced. In particular, body postures can be recognized by using sensors which need to be worn as well as vision based techniques.

For example, the approach of controlling mobile platforms by body postures (e.g., trunk tilt) is presented in [31,34]. A belt interface, encompassing a set of sensors to recognize user bendings, allows the user to control the robot motion and to receive tactile, visual and auditory feedback from the remote mobile robot.

On the other hand, vision based techniques [25] do not require to wear any contact devices, but use a set of sensors and algorithms for recognizing gestures. Therefore, the type of communication based on gestures can provide an expressive, natural and intuitive way for humans to control robotic systems. One benefit of such systems is that they propose natural ways to send geometrical information to the robot, such as: up, down, etc. As seen in [4], through the recognition of gestures, a natural language for HRI can be created, relying on non invasive systems such as a camera to identify user gestures for comparison with a predefined gesture database. Gestures may represent a single command, a sequence of commands, a single word, or a phrase and may be static or dynamic. Such a system should be accurate enough to provide the correct classification of gestures in a reasonable time.

The ability to recognize gestures is important for an interface developed to understand user's intentions. Interfaces for robot control that use gesture recognition techniques have been studied in depth, as using gestures represents a formidable challenge. In fact, several issues arise from environments with complex backgrounds, from dynamic lighting conditions, from shapes to be recognized (in general, hands and the other parts of the human body can be considered as deformable objects), from real-time execution constraints, and so on.

A lot of work has been focused on hand gesture recognition for human robot interaction. For instance, a gesture-based architecture for hand control of mobile robots was proposed in [30]. Gestures were captured by a data glove and gesture recognition was performed by Hidden Markov Model statistical classifiers. Then, the interpreted gestures were translated into commands to control the robot. The use of a data glove was then replaced by markers in [12]. Two cameras provided the information to triangulate the position of the hand markers, allowing gesture recognition to take place and control a 6-DOF (Six Degrees of Freedom) robot with a high precision. An alternative identification of the hand posture was proposed in [8]. The hand posture was identified from the segmented temporal sequence obtained by the Hausdorff distance method. A real-time vision-based gesture recognition system for robot control was implemented in [4]. Gestures were recognized using a rule-based approach by comparing the skin like regions in a particular image frame with the predefined templates in the system memory. Another hand gesture recognition system for robot control, which uses Fuzzy-C-Means algorithm as gesture classifier to recognize static gestures, was proposed in [35,36]. Static and dynamic gestures were recognized by a Fuzzy-C-Means clustering algorithm in [26].

2.2. Quadrotors and the Ar.Drone platform

Quadrotors are used in a large spectrum of applications ranging from surveillance to environmental mapping. Quadrotors are used singularly as well as in swarm; in the latter case, the task of coordination is always a critical issue. Quadrotors can be used both outdoor and indoor; outdoor platforms use, in general, autopilots for autonomous navigation, whereas several localization techniques (mainly based on computer vision) are exploited to determine position and orientation of indoor platforms, where GPS data are unavailable (for instance, [1,6,27,29,37]).

The human interface plays a key role when a quadrotor and, in general, any flying platform has to be directly controlled by the user. RC-transmitters and joysticks are the two most common input devices used to control quadrotors. Innovative solutions use multitouch devices (e.g., Apple iPhone [2] and Microsoft Surface [33]) and game controllers (e.g., Nintendo Wiimote [38]). Initial attempts of Microsoft Kinect usage to control the Ar.Drone have been proposed in [42,43]. In both cases, hand gestures are translated into commands for the platform.

The Parrot Ar.Drone [2] is a quadrotor helicopter with Wi-Fi link and two on-board cameras: a wide angle front camera and a high speed vertical camera. Software clients to control the platform