A Freehand System for the Management of Orders Picking and Loading of Vehicles

Pedro J.S. Cardoso[1,2(✉)], João M. F. Rodrigues[1,2], Luís Carlos Sousa[1,2], Andriy Mazayev[3], Emanuel Ey[1], Tiago Corrêa[3], and Mário Saleiro[2]

[1] Instituto Superior de Engenharia, University of the Algarve, Faro, Portugal
{pcardoso,jrodrig,lcsousa,amazayev, eevieira,tiagodcorrea,masaleiro}@ualg.pt
[2] LARSys (ISR-Lisbon), University of the Algarve, Faro, Portugal
[3] Faculdade de Ciências e Tecnologia, University of the Algarve, Faro, Portugal

Abstract. The process of picking goods from a warehouse and loading distribution vehicles is done in a systematic manner which in general corresponds to a certain order. For instance in the delivery of goods, it may be important to load the transportation vehicles in the reverse order of the customers visit, in furtherance of better accessing the products when unloading/delivering. This management can be troublesome if the human-computer interface requires the use of devices, like mouses or keyboards, which are difficult to be used under certain condition (e.g., human with dirty hands or wearing thick gloves/clothes). In this paper it is presented a proof-of-concept in the area of picking goods from a warehouse and the corresponding vehicle loading when using equipments which do not allow easy use of common human-computer interfaces. In this sense, an application using a 3D sensor was programmed to implement the human-computer interaction based on simple swipe gestures to navigate through the options, menu and their (de)selection.

Keywords: 3D sensor · Leap Motion · Orders picking and loading · Vehicle Routing Problem

1 Introduction

It is quite easy to find definitions of Human-Computer Interaction (HCI). The most probable to be find on the web is "Human–computer interaction involves the study, planning, design and uses of the interaction between people (users) and computers. It is often regarded as the intersection of computer science, behavioral sciences, design, media studies, and several other fields of study" [12]. It is quite therefore easy to recognize that the boundaries of what is HCI are quite fuzzy. Historically, in a non-exhaustive overview, HCI evolved from a set of switches, to punched cards, monitors, keyboards, mouse pointers, etc. Things are changing fast. Dialogs like the ones in the "2001: A Space Odyssey" movie between the computer HAL 9000 and humans [14] or the library host hologram in "Time Machine" movie where the hologram communicates and interacts naturally with

M. Antona and C. Stephanidis (Eds.): UAHCI 2015, Part IV, LNCS 9178, pp. 422–431, 2015.
DOI: 10.1007/978-3-319-20687-5_41

a time traveler [25] were once science fiction. The truth is that, as J. Grudin says, HCI is a moving target [10]. We expect the future of HCI to be supported on ubiquitous communication where computers communicate to give universal access to data and computational services, high functional systems where accessing those functionalities is natural, mass availability of computer graphics, high-bandwidth interaction, wide variety of displays (e.g., on common surfaces, with flexibility, large and thin), and embedded computation.

Currently most computers and mobile devices have the computational capacity and are equipped to mimic human's capacities like sight and hearing, and even, with the appropriate sensors, temperature, taste, smell, touch, balance and measure acceleration [11,24]. Some of these capabilities (e.g., touch and gesture) can be used to control machines in a natural and intuitive way. A huge amount of sensors can be used to that purpose, such as embedded cameras, touch screens or mobile 3D sensors (e.g., Structure Sensor [22], the Leap Motion [16], or the Kinetic sensor [13]). Some examples are the interaction with art installations [2], in robotics [6], for head pose classification [26], in assistive technologies [5], in the operation of wheelchairs [1] or in the interaction with holograms for teaching technical drawing [7].

Bearing the previous context in mind, this article aims to present a proof-of-concept of a system designed to be used in the picking and loading area of a warehouse, where employees use clothes (e.g., gloves or thick clothes) or have dirty hands which make troublesome the use of common interfaces, like keyboards or mouses. In particular, we propose an application which makes possible navigate through a set of menus presenting routes, products and loading order. The HCI is supported on 3D sensor and uses simple swipe gestures to navigate through the options, menu and their (de)selection.

The remaining document is structured as follows. Section 2 presents a more deep contextualization and problem formulation. Section 3 explains the 3D sensor API and its configuration, while the interface implementation is shown in Sect. 4. The final section presents some conclusions and future work.

2 Contextualization and Problem Formulation

The present work was a study thought as part of the Intelligent Fresh Food Fleet Router (*i3FR*) project, developed by the University of the Algarve and X4Dev, Business Solutions. The project has the objective of building a system to manage and optimize the distribution of fresh and frozen goods by a distribution fleet [4]. In short, the system integrates an Enterprise Resource Planning (ERP) software with an optimization system (called *i3FR*) to minimize the costs of the distribution routes [3]. The routing optimization computes routes from the depots to the delivery points using cartographic information and takes into consideration multiple objectives to be minimized, namely: the number of vehicles and the total traveled distance.

In a sequence line the distribution process can be described as follows (see Fig. 1 for a sequence diagram of the procedure described next). First a client or a seller sends an order with a list of products and quantities to a seller that

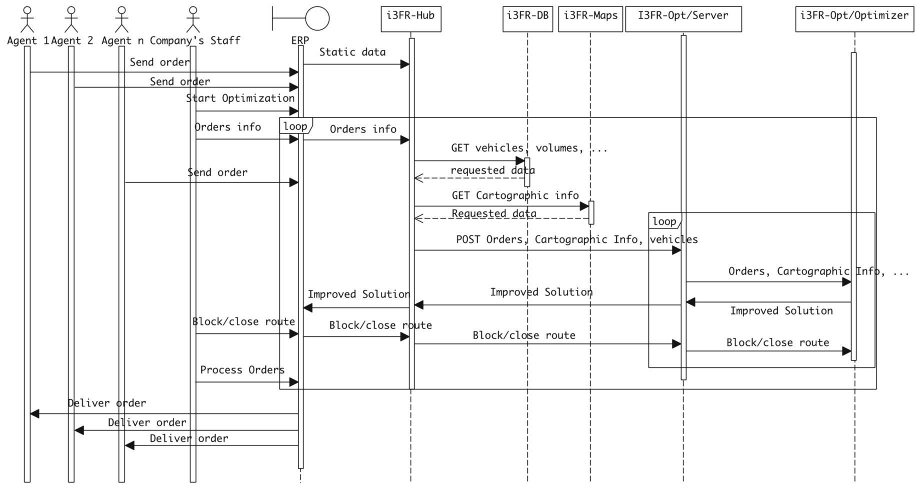


Fig. 1. Sequence diagram of the flow from the customers to the optimizer and reverse.

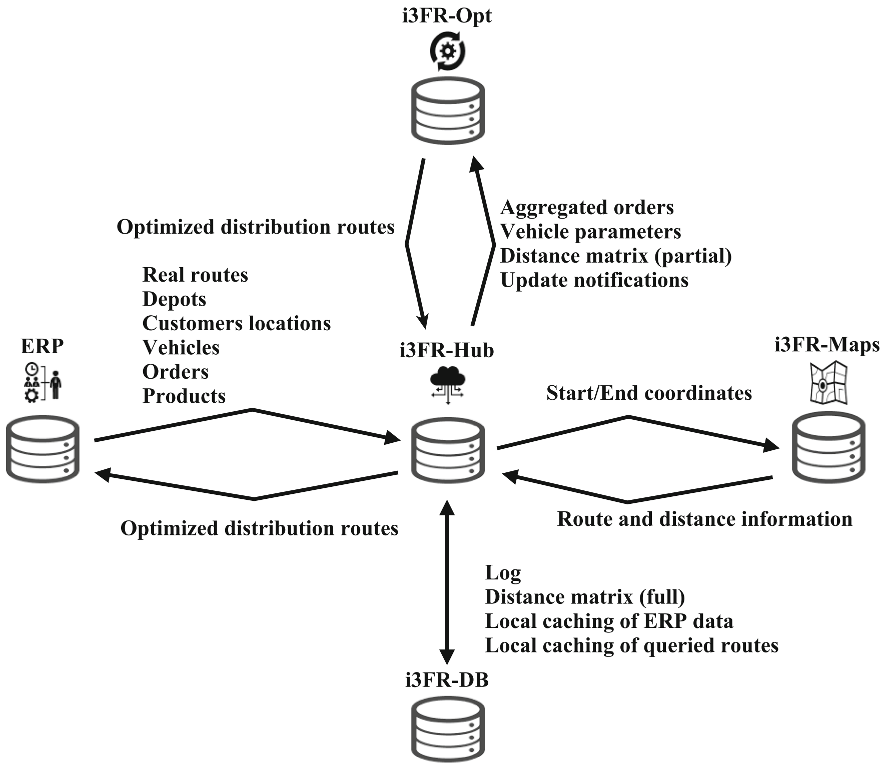


Fig. 2. *i3FR* components and the data flow them.

introduces the data into an ERP. This order is for one of the next days. Then, at a convenient moment, the system manager starts the optimization procedure by sending a signal to the *i3FR* system which, for modularity convenience, is divided in four main modules: *i3FR-Hub*, *i3FR-Opt*, *i3FR-DB* and *i3FR-Maps*(see Fig. 2). The signal triggered by the administrator goes to the *i3FR-Hub* which begins by requesting the necessary data from the ERP (e.g., orders, new customers data, available vehicles and new products data) and store it on the *i3FR-DB*. In this sense, updated data, i.e., data already fetched that was not changed since previous optimization sessions (e.g., older customers delivery locations, vehicle data), was stored in the *i3FR-DB*. The *i3FR-DB* acts as local database avoiding the overload of the ERP with repetitive requests of data.

The insertion of new customers trigger the *i3FR-Maps* module to update its distance matrix which stores the routes between all possible delivery locations. On other words, *i3FR-Maps* maintains a $n \times n$ matrix (n being the number of delivery locations) of routes including distances, travel time and corresponding routing directions details. The routes data is obtained from a Open Street Maps Routing Machine (OSRM) server [18] and Google Maps [9]. The use of the OSRM overcomes the limit of accesses to Google's API. The routes data is also stored in the *i3FR-DB*, which is supported on a MongoDB database [17].

Finally, in presence of the necessary data, the *i3FR-Hub* send a start signal (and the necessary data) to the actual optimization module, *i3FR-Opt*. The *i3FR-Opt* implements an hybrid algorithm based on the Push Forward Insertion Heuristic (PFIH) [20] which computes solutions for a Capacitated Vehicle Routing Problem with Time Windows (CVRPTW) [3,8], optimizing the number of routes and the total distance necessary to make all deliveries. Whenever good solution for the active CVRPTW are achieved, they are send to the *i3FR-Hub* which in turn sends them to the ERP. In the presence of the solutions, the ERP administrator can send signals to stop totally (or partially) the optimization process and start the loading of the vehicles.

A solutions is a set of routes. Each route starts at the depot and passes through a set of customers, returning to the depot. In order to better access the products when making the deliveries to the customers, it is important to load the vehicles in the reverse order of the customers visit, keeping the first products to be delivered near the doors and so forth. Given the routes, the loading order information is sent to the warehouse where the picking and loading of the vehicles is to be made by workers. Problems my arise if for instance the workers can't easily interact with the computers having the orders information. This problems can arise for instance if the worker are using thick clothes, gloves or, simply have dirty hand for directly operating products (e.g., fresh meat/fish).

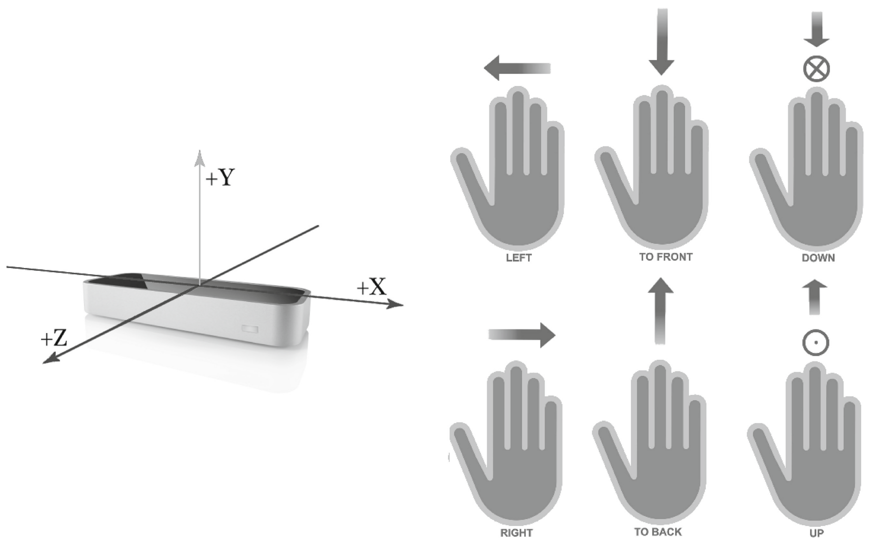
The next sections present an HCI alternative to manage the picking and loading of the products. The solution is based on a 3D sensor which allows, with simple gestures, to interact with a loading and picking interface.

3 3D Gesture Recognition with Leap Motion

Leap Motion [16] has an Application Programming Interface (API) capable of detect multiple hand gestures, such as a circular movement by a finger, a straight line movement by the hand with fingers extended, a forward tapping movement by a finger/hand or a downward tapping movement by a finger/hand. The configurations and proper combinations of gestures detection, allow the recognition of a large set of actions, such as, open or close hand done with one or both hands, generally used to implement zoom in or out. Other examples are the movement from side to side with the hand to indicate a swipe gesture or a finger poking forward can indicate a screen tap gesture [15,19,21,23].

The Leap Motion senses the space as 3D space with standard Cartesian coordinate system, also known as right-handed orientation (see Fig. 3(a)). The origin of the coordinate system is centered at the top of the device. The x-axis is placed horizontally along the device, with positive values increasing from left to right, the z-axis is placed also on horizontal plane, perpendicular with x-axis with values increasing towards the user (the front side of the device) and the y-axis are placed is the vertical, with positive values increasing upwards.

A minimum set of intuitive movements where considered for the implementation of the problem in question, namely: the horizontal and vertical swipe gestures. The horizontal swipe gesture were used to navigate between options and menus, while the vertical swipe gestures were used to select and deselect options and menus (see Sect. 4). In this sense, the Leap Motion API has a direction vector for the swipe gesture that associates a 3D direction vector ranging from -1.0 to $+1.0$ in each direction, after a gesture is completely recognized, given the minimum length and velocity configurations properties. As this gesture was the only one used, it was necessary to detect various types of swipe. The interface was designed to react to two of the six different independent types of swipe gestures



(a) (b)

Fig. 3. (a) Leap Motion coordinate system; (b) Six types of swipes possibles with LeapMotion gesture recognition.

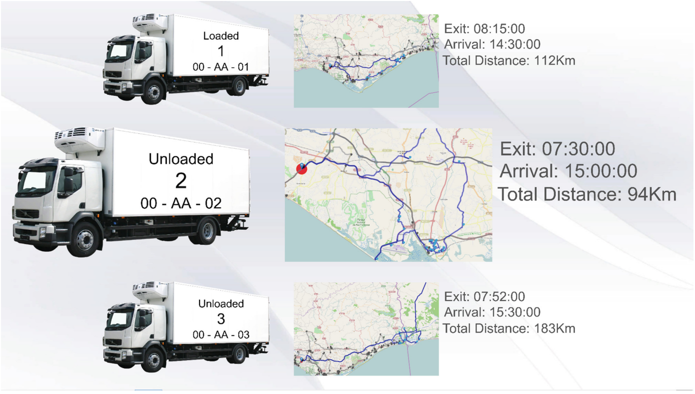


Fig. 4. Screen-shots from the user's interface: route/vehicle listing.

Fig. 5. Screen-shots from the user's interface: list of products to be picked.

Fig. 6. Screen-shots from the user's interface: list of customers and products to be loaded in the corresponding order.

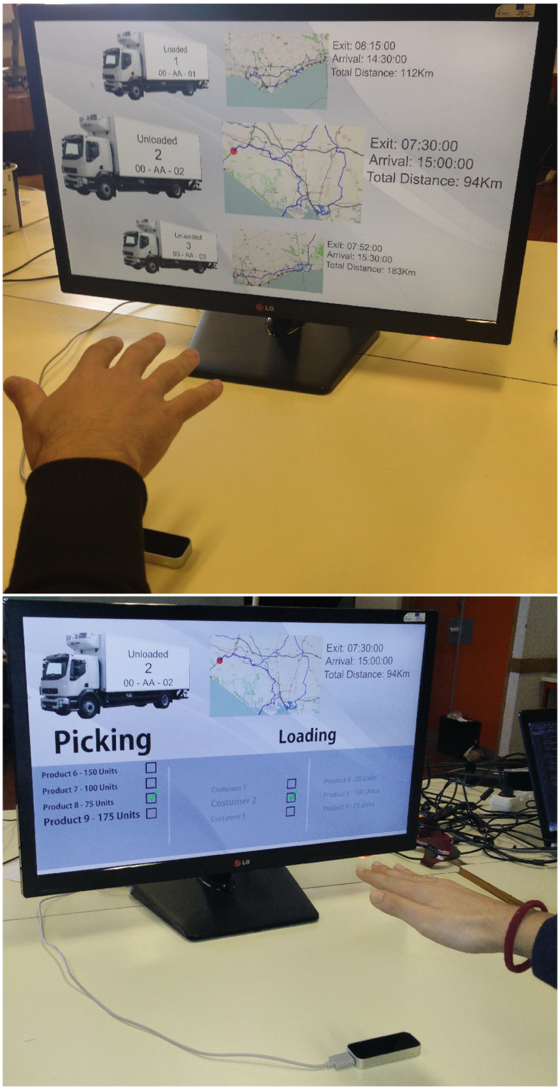


Fig. 7. User interacting with the interface, navigating: in the list of routes (top) and list of products to be picked from the warehouse (bottom).

(three of each swipes are the opposite of the other three) as shown by Fig. 3(b), namely: (i) Up/Down swipe, accomplished by a top to bottom or a bottom to top gesture (y-axis); (ii) Front/Back, accomplished by a front to back or back to front gesture (z-axis); and (iii) Left/Right swipe, accomplished by a left to right or right to left gesture (x-axis).

For the first case, of a up/down swipe, the movement depends mainly on the y-axis. The direction vector has an upward direction if $y \approx +1$ (used as a "deselect" action) and a downward direction if $y \approx -1$ (used as a "select" action). Since it is almost impossible to do a swipe gesture with a vector direction component of exactly $x = 0$, $y = \pm 1$ and $z = 0$, it was considered a range of values to detect and differentiate between swipes types. Therefore, any swipe direction that agrees with the condition $y \in [-1, -0.5[$ and $x, z \in [-0.5, 0.5]$, is considered as a downward swipe. Oppositely, a swipe direction that agrees with the condition $y \in]0.5, 1]$ and $x, z \in [-0.5, 0.5]$, is considered as an upward swipe.

The remaining cases, (ii) and (iii), are defined by similar equation taking into consideration that the front/back and the left/right movements are defined along the z-axis and x-axis, respectively. These swipes are mutually independent that is, for every type of swipe there is only one possible choice.

4 Interface Implementation and Test

To provide a proof-of-concept we devised a simple interface composed by three main interface pages. These pages are associated with the routes, picking and loading actions.

The application starts by opening a page listing the routes (see Fig. 4). For each route is presented a set of informations namely: the vehicle license plate and number ID, the state of the vehicle (loaded/unloaded), the corresponding route represented in a map, the hour that the vehicle should leave the depot (such that the customer's time windows are satisfied), the estimated arrival time and the total distance. Some of these values, like the exit from the depot hour, is used to sort the loading of the vehicles (the ones exiting earlier are usually loaded first). For this interface, horizontal front/back or back/front (z-axis) swipes are used to navigate through the routes/vehicles. A vertical swipe selects the vehicle and moves to another interface showing the a list of products to be picked from the warehouse (see Fig. 5). Again, horizontal front/back or back/front (z-axis) swipes are used to navigate through the picking list. The active product has a larger font and a vertical down/up swipe activates/deactivates a check box, used to register the products already picked. When the products are to be loaded, a horizontal swipe from right to left (x-axis) is used to activate the loading page. The loading page presents a list of costumers in the vehicle's loading order. Horizontal back/front (z-axis) swipes allow to navigate between the customers and the corresponding list of products to be loaded (see Fig. 6). Again, a vertical down/up swipe activates/deactivates a check box, used to register the products/customers already loaded.

Figure 7 shows two examples of users navigating in the list of routes (top) and list of products to be picked from the warehouse (bottom). The tests where made with the sensor regulated a minimum length swipe of 130 mm and 200 mm/s of velocity.

5 Conclusions and Future Work

This paper presents a proof-of-concept study conducted to implement a HCI capable of overcome the difficulties of working with common devices (e.g., mouses or keyboards) when wearing thick clothes/gloves or having dirty hands. The study considers also the problem of picking products from a warehouse and their load into vehicles to be delivered to customers.

Test showed that the interface can be useful but some problems may arise at the beginning since it is necessary some adaptation to the use of the 3D sensor. Nevertheless, the used sensor has a finger tracking fast and accurate, is small (with dimensions of 13 mm × 13 mm × 76 mm and weighting 45 g) and inexpensive, which allows it to be placed almost with every display. Possible drawbacks are the supported conditions of operating like the temperature (0° to 45° C), the relative humidity (5 % to 85 %) or light conditions (bright sunlight, bright light sources or reflective surfaces are not recommended).

As future work, we intend to further explore the proposed prototype in other conditions and compare it with other 3D sensors.

Acknowledgements. This work was partly supported by project *i3FR*: Intelligent Fresh Food Fleet Router – QREN I&DT, n. 34130, PRHOLO: The realistic holographic public relations – QREN I&DT, n. 33845, POPH, FEDER, the Portuguese Foundation for Science and Technology (FCT), project LARSyS FCT [UID/EEA/5009/2013]. We also thanks to project leader X4DEV, Business Solutions.

References

1. Abedan Kondori, F., Yousefi, S., Liu, L., Li, H.: Head operated electric wheelchair. In: Southwest Symposium on Image Analysis and Interpretation, pp. 53–56. IEEE (2014)
2. Alves, R., Madeira, M., Ferrer, J., Costa, S., Lopes, D., da Silva, B.M., Sousa, L., Martins, J., Rodrigues, J.: Fátima revisites: an interactive installation. In: Proceedings of the International Multidisciplinary Scientific Conference on Social Sciences and Arts, pp. 141–148. SGEM (2014)
3. Cardoso, P.J.S., Schütz, G., Mazayev, A., Ey, E.: Solutions in under 10 seconds for vehicle routing problems with time windows using commodity computers. In: Proceedings of the 8th International Conference on Evolutionary Multi-Criterion Optimization, EMO 15. p. (accepted for publication)
4. Cardoso, P.J.S., Schütz, G., Semião, J., Monteiro, J., Rodrigues, J., Mazayev, A., Ey, E., Viegas, M., Neves, C., Anastácio, S.: Integration of a food distribution routing optimization software with an enterprise resource planner. In: Proceedings of International Conference on Geographical Information Systems Theory, Applications and Management, GISTAM-15. p. (accepted for publication)
5. Chiang, I.T., Tsai, J.C., Chen, S.T.: Using Xbox 360 Kinect games on enhancing visual performance skills on institutionalized older adults with wheelchairs. In: IEEE International Conference on Digital Game and Intelligent Toy Enhanced Learning, pp. 263–267. IEEE (2012)
6. El-laithy, R.A., Huang, J., Yeh, M.: Study on the use of Microsoft Kinect for robotics applications. In: IEEE/ION Position Location and Navigation Symposium, pp. 1280–1288. IEEE (2012)

7. Figueiredo, M., Sousa, L., Cardoso, P., Rodrigues, J.A., Goncalves, C., Alves, R.: Learning technical drawing with augmented reality and holograms. In: Proceedings of the 13th International Conference on Education and Educational Technology, pp. 11–20. WSEAS (2014)
8. Gambardella, L.M., Taillard, É., Agazzi, G.: MACS-VRPTW: a multiple colony system for vehicle routing problems with time windows. In: New Ideas in Optimization. Citeseer (1999)
9. Google: Google Maps (2015). http://maps.google.com. Accessed 20 January 2015
10. Grudin, J.: A Moving Target: The Evolution of HCI, 3rd edn. Taylor & Francis, New York (2012)
11. Halder, A., Mahato, M., Sinha, T., Adhikari, B., Mukherjee, S., Bhattacharyya, N.: Polymer membrane electrode based potentiometric taste sensor: a new sensor to distinguish five basic tastes. In: 2012 Sixth International Conference on Sensing Technology (ICST), pp. 785–789, December 2012
12. HCI: Human–computer interaction – Wikipedia, the free encyclopedia (2015). http://en.wikipedia.org/wiki/Human-computer_interaction. Accessed 20January 2015
13. Kinect: Kinect for windows (2015). http://www.microsoft.com/en-us/kinectforwindows/. Accessed 20 January 2015
14. Kubrick, S.: 2001: A space odyssey (1968)
15. Leap Developer Portal: Leap developer portal (2015). https://developer.leapmotion.com/documentation/skeletal/csharp/devguide/Leap_Overview.html. Accessed 20 January 2015
16. Leap Motion: Leap motion (2015). https://www.leapmotion.com. Accessed 20 January 2015
17. MongoDB Inc: MongoDB (2015). http://www.mongodb.com/. Accessed 20 January 2015
18. OSRM: OSRM - Open Source Routing Machine (2015). http://project-osrm.org/. Accessed 20 January 2015
19. Potter, L.E., Araullo, J., Carter, L.: The leap motion controller: a view on sign language. In: Proceedings of the 25th Australian Computer-Human Interaction Conference: Augmentation, Application, Innovation, Collaboration, pp. 175–178. ACM (2013)
20. Solomon, M.M.: Algorithms for the vehicle routing and scheduling problems with time window constraints. Oper. Res. **35**(2), 254–265 (1987)
21. Spiegelmock, M.: Leap Motion Development Essentials. Packt Publishing Ltd., Birmingham (2013)
22. Struture: Struture sensor (2015). http://structure.io/ Accessed 20 January 2015
23. Sutton, J.: Air painting with corel painter freestyle and the leap motion controller: a revolutionary new way to paint! In: ACM SIGGRAPH 2013 Studio Talks, p. 21. ACM (2013)
24. Villarreal, B.L., Gordillo, J.L.: Perception aptitude improvement of an odor sensor: model for a biologically inspired nose. In: Carrasco-Ochoa, J.A., Martínez-Trinidad, J.F., Rodríguez, J.S., di Baja, G.S. (eds.) MCPR 2012. LNCS, vol. 7914, pp. 126–135. Springer, Heidelberg (2013)
25. Wells, S.: Time machine (2002)
26. Yun, Y., Changrampadi, M.H., Gu, I.Y.: Head pose classification by multi-class AdaBoost with fusion of RGB and depth images. In: International Conference on Signal Processing and Integrated Networks, pp. 174–177. IEEE (2014)