Available online at www.sciencedirect.com

ScienceDirect

Procedia Engineering 181 (2017) 575 – 582

Procedia Engineering

www.elsevier.com/locate/procedia

10th International Conference Interdisciplinarity in Engineering, INTER-ENG 2016

An Anthropomorphic Hand with Five Fingers Controlled by a Motion Leap Device

Catalin Constantin Moldovan, Ionel Staretu*

Transilvania University of Brasov, Brasov 500036, Romania

Abstract

This paper presents a new solution for command and control of the one anthropomorphic gripper with five fingers intended to be used in industrial robots equipment assemblies used for achieving low and medium complexity. The command solution is based on Motion Leap device and some software module: HandCommander, HandProcessor and HandSIM. The object to be gripped is recognized, using the SpatialVision application based on the image analysis, the 3D model is loaded in the GraspIT application. The user gesture is recognized and sent to the gripping test module and the RoboHand component to grip the objects preconfigured. The object is gripped in the physical environment by the RoboHand component, the anthropomorphic gripper with five fingers. We shown as example tennis ball gripping.

Peer-review under responsibility of the organizing committee of INTER-ENG 2016

Keywords: Anthropomorphic gripper; Motion Leap; Software; Simulation; Command; GraspIT.

1. Introduction

Anthropomorphic grippers are inspired from the human hand, having various constructive and functional features. Compared to other classes of grippers like jaw grippers [1,2] or tentacular grippers [2], they have obvious advantages because they are more similar to the human hand, both constructively and functionally, considering the human hand as the most perfected gripper [2]. Today there are more versions of anthropomorphic grippers, some of which only designed, others as prototype and some as commercial products [3,4,5,6]. An important issue is the choice of the command and control system for these grippers. After the study in the current stage, we consider that the optimization of the existing methods of grippers control, and research and implementation of new command and

* Corresponding author. Tel.: +40-744-309-186; fax: +40-268-412-921.
E-mail address: istaretu@yahoo.com

Peer-review under responsibility of the organizing committee of INTER-ENG 2016
doi:10.1016/j.proeng.2017.02.436

control techniques significantly facilitate anthropomorphic grippers operation. In addition, due to the relatively high costs of grippers' acquisition, virtual reality simulation can greatly reduce the time and cost associated with the implementation of a project. Starting from the current stage of research in the field, this paper identified components needed to create a command and control module for an anthropomorphic gripper, both virtually and physically. Compared to existing methods, it can reduce significantly costs associated to a project and it may allow natural interaction between human hand and gripper. In the development of the command and control module, we taken into account innovations in the field of digital image processing, based on which, we will develop an alternative to existing techniques for the command and control of the anthropomorphic grippers, using data gloves. As a result, we adopted as objective of the research presented in this paper the design and the implementation of an innovative command and control system for an anthropomorphic gripper, using innovations in the digital image processing and the development of specialized algorithms for the capture of hand movements and its gestures in videos. On this purpose, we suggest using the Motion Leap device, and in the paper we present briefly the necessary software modules for the command and control of an anthropomorphic gripper with five fingers, for which we also present the test results for gripping a tennis ball.

2. Design, implementing and use of the SpatialVISION application

Locating and classifying objects in a scene automatically is a difficult and current issue. A robot would benefit from the ability to classify unknown objects in object classes, which would be very useful in engineering or domestic applications. Currently, there are many ways of modelling objects and recognition techniques.

In many applications currently used to recognize objects, we use the concept of primitives [8,9], but this concept does not allow the recognition of complex objects. Because of this, we developed algorithms based on search of features in images [9] and [10] or classifiers [11].

For recognizing objects, in this paper we use algorithms based on search of features in images. We also use the research result of [12] to obtain invariance to noise by providing a smoothing filter on image and the research result of [13] to obtain invariance to illuminating conditions through the use of image derivatives reported to the level of gray.

To identify the features detection algorithm, we identified in the specialty literature the following major approaches: **SIFT** [14] – it is an algorithm used in image processing, by extracting features of objects; **GLOH** [15] – it is an algorithm that uses image descriptors that can be used in actions for objects recognition; **SURF** [16] – it is a way used to detect objects invariant under rotation; **HOG** [17] – it is an algorithm for detection of objects used in image processing.

To implement SpatialVISION application, we used the SIFT algorithm implementation. The SIFT algorithm implementation presented in this paper is our own approach in C # using EmguCV library for working with images.

To use the application SpatialVISION, we also conceived a utility called SVAdminUtility, which is used to populate a database of three-dimensional objects that will later be recognized. The software component for the database management of objects that can be recognized by SpatialVISION application is called SVAdminUtility. With this component, a knowledge database can be populated initially and subsequently enriched with new three-dimensional models and different physical properties.

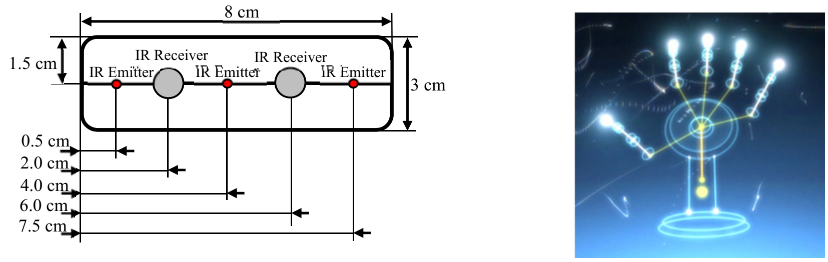


Fig.1. (a) Motion Leap sensor elements; (b) Human hand captured by using the Motion Leap device.

All data, object, three dimensional model, images and points are stored in the knowledge database. To do this, we proposed the scheme of tables set forth in the diagram shown in Fig. 2, using the normal form three in the database theory.

In the diagram UML (Unified Modelling Language) of the database developed within the application, three problems are solved: adding objects and assigning images to an object; adding to the three-dimensional model and associating an object; adding lists of physical properties to each object. In the diagram UML, see Fig.2, of the database developed within the application, three problems are solved: adding objects and assigning images to an object; adding to the three-dimensional model and associating an object; adding lists of physical properties to each object. The database originally attached to the application SVAdminUtility is populated by 12 different objects from different classes of objects. Table 1 shows objects from various classes for which three-dimensional models were attached. All objects were chosen randomly and a three-dimensional model was originally added.

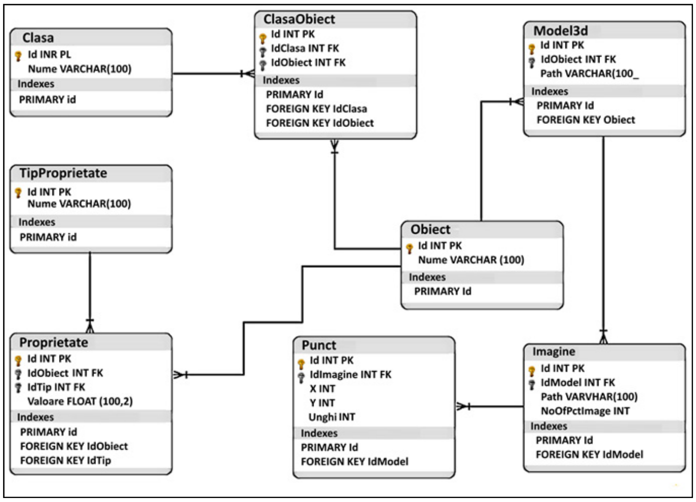
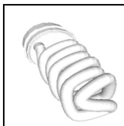
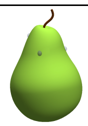
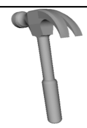
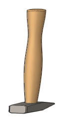


Fig. 2. UML diagram of tables that make up the data model.

Because the application SpatialVISION makes comparisons between images, for each model, we will generate a number of images from different perspectives (see Fig. 3). It was noted that the low number of perspectives is the main cause of instability in the case of detection algorithms for the features in scenes from video streams in real time ([18,19] and [20]). To implement the database we used the standard SQL, namely Microsoft SQL Server version. For working with database, we made the application SVAdminUtility using Microsoft .NET framework and Microsoft Visual Studio 2013.

Table 1. Objects in the database

 a	 b	 c	 d	 e	 f	 g	 h

SVAdminUtility application takes advantage of the concept of Windows Forms to create human-computer interfaces. This concept is available in the Microsoft .NET framework and presents a paradigm used to create graphical user interfaces for the Windows operating system. For each model stored in the database, because SIFT algorithm is partially invariant to rotation, images were generated in different perspectives, with SIFT points for each image individually. For each object that is added, to generate images from different perspectives (Fig. 3) the

OpenGL framework is used. For every image separately, we search, using the classification method SIFT, the points of interest.

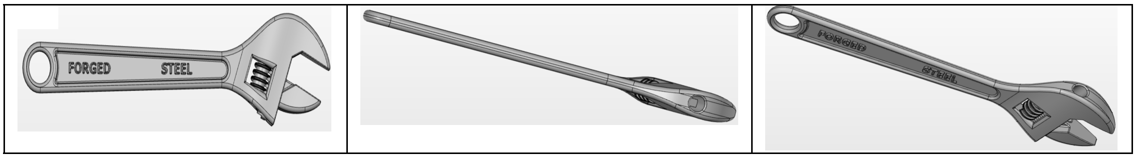


Fig. 3. Images generated using a three-dimensional model from different perspectives.

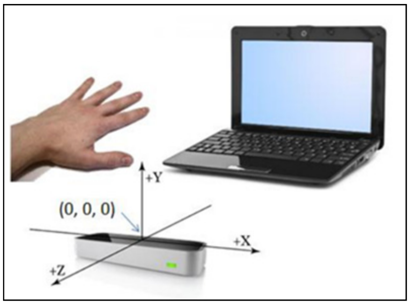
SpatialVISION application presented in this paper represents the implementation of a software module for the following purpose: recognizing objects in images or video stream based on image processing algorithms implemented; estimating physical properties of objects recognized; creating a virtual three-dimensional object based on the recognized object in the image. The application allows the loading of models or parts of an object, having the possibility to specify the physical features that may influence interaction with other objects. For storing and transferring objects with physical properties, we use the three-dimensional model of files XML – COLLADA [21]. To work with SpatialVISION application, we need to follow the following steps: (1) **System configuration** using the SVAdminUtility component and (2) **System operation** through the user component SpatialVISION. The SpatialVISION application, presented in this paper is to implement a software module for the following purpose: recognizing objects in images or video stream based on image processing algorithms implemented; estimating physical properties of the objects recognized; creating a virtual three-dimensional object based on the object recognized in the image.

The application allows the loading of models or parts of an object, having the possibility to specify physical features that may influence interaction with other objects. For storing and transferring objects with physical properties we use the three-dimensional model of files XML – COLLADA [21]. The aim of the application SpatialVISION is to achieve connection between the real environment and the virtual reality simulator. The software module is used to generate the three-dimensional model of the object that will then be gripped. The general architecture of the module classes in the application SpatialVISION used to recognize objects consists of the following components: ImageObject; SIFTRecognizer; SURFRecognizer.

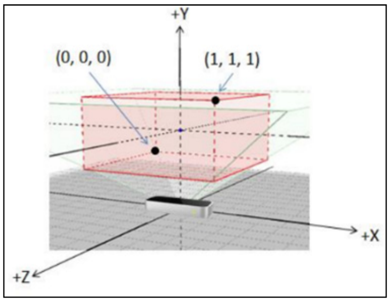
For detection of interest points, SpatialVISION application was created, which is able to work with a video stream as follows: data capture is performed, via capture component. The capture component receives a video stream and it divides it in frames. Each frame is then forwarded to processing; each image is sent to the processing function, where points of interest are generated, which are subsequently forwarded to the search component; all the interest points are searched in the database, and they serve to recognize or not an object. Based on points of interest, the three-dimensional file attached is found. This file is sent to the functional simulation module using the library EmguCV.

3. Using the Motion Leap sensor to detect human hand and capture complex gestures

Information about a user's hand, fingers and gestures can be recognized if the human hand is between 25 mm and 500 mm above the center of the sensor (Fig. 4a). The precision of the Motion Leap sensor is of the sub-millimeter order, as specified in the documentation of MotionLeap, more exactly 0.01 mm [22]. Using the programming interface that comes with the MotionLeap device, the user has access to data on the direction, the speed and the rotation of the human hand. Motion Leap device has a programming interface attached, called Motion Leap SDK. This allows programmatic access to depth data and it was created so as to allow programmers to use high-level or average languages (Objective C, C ++, Java). From a functional perspective, the sensor MotionLeap, analyzes a data stream which is divided by SDK frames, directly dependent on time, allowing calculation of gestures or speed at which the human hand moves above the sensor. One great advantage of the sensor MotionLeap compared to its predecessors is that it can distinguish each finger individually. The identification algorithm attaches to each finger, gesture, hand a unique identifier, making it easy for each entity to be traced along the flow. If, at some point, a detected object, to which it was attached a unique identifier disappears from the frame, then reappears, it will have a unique identifier attached. Positional data that can be captured for each item separately are provided with a millimeter accuracy on each axis of coordinates X, Y, Z, from the center of the device (Fig. 4b).



a



b

Fig. 4. (a) Device center and axes of coordinates; (b) Interaction cube [23].

Spatial data captured from the real environment, are translated by using the SDK in different environments, which are related to the following classes: Screen, InteractionBox, Touchzone. To use and test the system, we considered first, recognition of different gestures of the human hand, then using these gestures we tested gripping different objects in the virtual environment. For hand recognition, more sequence of data captured by the device MotionLeap were analyzed. These sequences were captured in different lighting conditions (daylight or artificial light) and from different perspectives. In Fig. 5a we can see gestures used in the recognition test. Empirically, we observed that the recognition of gestures is not dependent on lighting conditions when using the device MotionLeap. In Fig. 5b. we can see hand gestures recognized and each joint of the fingers.

To assess and forward human hand gestures in the virtual environment, we consider creating a module for interaction with the virtual environment. The system is divided into five components: hardware, frames processor, gestures detector and the component HandCOMMANDER. **The hardware component** represents the sensor MotionLeap and is responsible for providing raw data. Once the data are available, the device MotionLeap sends the data to **the frames processor for** further evaluation.


a

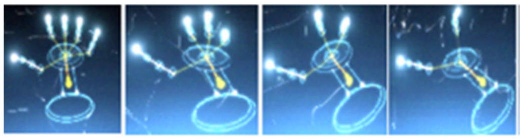

b

Fig. 5. (a) Human hand gestures for which recognition is sought; (b) Gestures recognized using the MotionLeap device.

The frames processor consists of the software module MotionLeap and the **gestures detector.** The MotionLeap software module processes the input from the sensor and calculates the information about the current status of the human hand. The human hand is forwarded to the gestures detection module whose main activity is to detect gestures, such as selection, move and object release gestures. When one of the gestures was detected, the gestures detection module notifies the **HandCOMMANDER** functional simulation component for transmitting data to perform the gesture in the virtual environment. The component architecture is divided into module Helpers - which contains helper classes that are used by other components; HandProcessor – frames processor that detects hand gestures; HandSIM interface - communication interface with virtual reality simulator, which is represented by the component HandCOMMANDER.

As results of the system performance, detected gestures are used to control a virtual hand (Fig. 6).

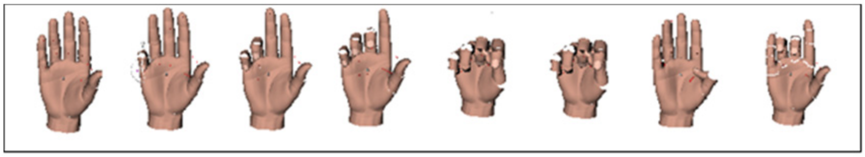


Fig. 6. Virtual hand gestures.

4. Handling virtual objects

The interaction presented between human hand (using the device MotionLeap) and the virtual environment (HandCOMMANDER component), is an alternative to the interaction based on the use of data gloves. The system is presented graphically in Fig. 7.

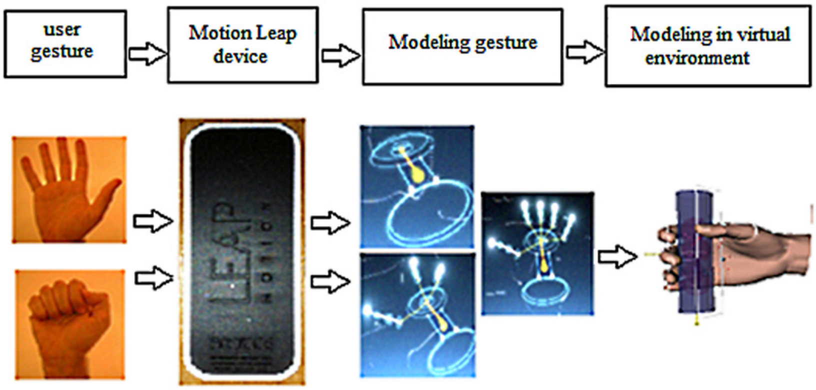


Fig. 7. Detection and recognition system for human hand gestures used to command and control a virtual hand.

As a test of implementation and communication between the MotionLeap sensor and the virtual environment, we performed gripping, moving and releasing of a French key. Gripping and releasing are simulated using gestures captured by MotionLeap and are simulated in the application GraspIT [24] (Fig. 8). For gripping and releasing the object, we should read two ways to configure the distinct human hand, namely, all the fingers open and all fingers closed. Gestures are transferred to the virtual environment for various gripping operations and handling, strength gripping (Fig. 8).

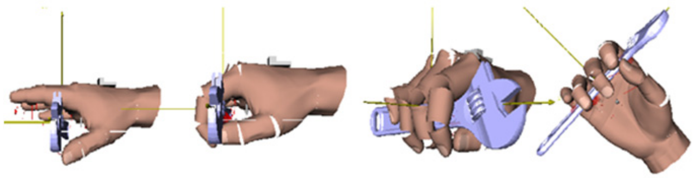


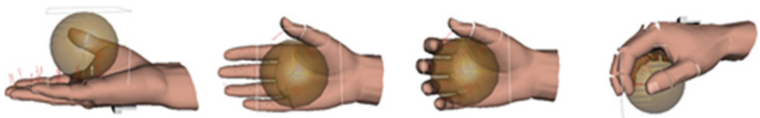
Fig. 8. French key grip in different positions.

The result of implementing the system and of handling tests performance is positive. The system performs actions of objects handling in the virtual environment in real time with little delay when attempting to detect multiple gestures in a very short period of time. This delay comes because the data transfer between the component HandProcessor and the component HandSIM is via the serial port, so the delay is caused by the hardware system. If the user does not perform fast gestures, the delay is not perceived by the user, and the system implementation can be considered a truly natural interaction between humans and the virtual environment.

5. Concept of the experiment to test the command and control strategy

The objective of the experiment was gripping a tennis ball, a bearing, a shaft and a French key. The experiment is performed in the following manner: the object to be gripped is recognized using the application SpatialVISION based on image analysis, the three-dimensional model is loaded in the application GraspIT; the user gesture is recognized, it is transmitted to the gripping testing module and it is pre-configured the RoboHAND component for gripping the object; gripping in the physical environment is performed by the component RoboHAND. Actual gripping of the objects is tested in the gripping test component, angle values between phalanxes are sent back to the anthropomorphic gripper using the RoboCOMMANDER interface that interprets these values and activates the RoboHAND component fingers. The data captured by the capture interface are interpreted by the virtual gripping test module to calculate the kiematic model and to calibrate an anthropomorphic gripper. The fingertip positions are calculated in the three-dimensional space. The positions are then transferred in the physical space XYZ using RoboCOMMANDER interface.

We present as example tennis ball gripping, the gripper is set initially in the position of Fig. 9 in the virtual environment. To achieve the object gripping, a gripping action is performed, it is forwarded to the virtual environment using the HandCOMMANDER interface (Fig. 9c, d) and to the physical environment through the RoboCOMMANDER interface (Fig. 10).



a b c d

Fig. 9. (a, b) Sphere positioned on hand; (c, d) sphere gripped from different perspectives.

Fig. 10. Gripping tennis ball and moving gripper in different positions to demonstrate the quality of the gripping.

6. Conclusions

On the basis of this work the following conclusions can be drawn:

- MotionLeap device enables natural interaction between the user and virtual reality at the level of gestures, this is comparable to the interaction based on data gloves. To follow human hand effectively we noted that MotionLeap device is a device by which the human hand gestures can be recognized with great precision;
- the system presented provides an interface through which a virtual hand can be animated directly using human hand. Once animated, this virtual hand can perform different actions in the virtual environment with ease;
- using image sensors, one can draw the final conclusion of this research, that we can create ways based on image analysis for detection, tracking and recognition of human hand gestures efficiently, which can be used successfully to command and control an anthropomorphic gripper with five fingers through the use of specialized software modules.

Acknowledgements

We express our gratitude to the company CLOOS in Germany and its representative in Romania, Timisoara ROBCON company, for supporting our research whose results are presented in part in this paper.

References

[1] I. Staretu, Gripping systems (in Romanian), Lux Libris Publishing House, Brasov, 2010.
[2] I. Staretu, Gripping systems, Derc Publishing House, Tewksbury, Massachusetts, 2011.
[3] H. Kawasaki, T. Komatsu, K. Uchiyama, Dexterous anthropomorphic robot hand with distributed tactile sensor: Gifu hand II, IEEE/ASME Transactions on Mechatronics, 7 (3) (2002) 296-303.
[4] Available online at: http:// www.shadowrobot.com /products/, 2013.
[5] M. Ceccarelli et al, Design and tests of a three finger hand with 1-DOF articulated, Robotica Journal, 24 (2) (2006) 183-196.
[6] L. Birglen, C. Gosselin, Optimal Design of 2-Phalanx Underactuated Finger, Intelligent Manipulation and Grasping 2004, pp. 110-116.
[7] E. Chng, New Ways of Accessing Information Spaces Using 3D Multitouch Tables, Proceedings of the International Conference on Cyberworlds (CW), Birmingham, UK, 2012, pp. 144-150.
[8] P. Carr, Y. Sheikh, I. Matthews, Monocular Object Detection Using 3D Geometric Primitives, ECCV'12 Proceedings of the 12th European conference on Computer Vision, 7572 (2012) 864-878.
[9] A. Berner, J. Li, D. Holz, J. Stuckler, S. Behnke, R. Klein, Combining Contour and Shape Primitives for Object Detection and Pose Estimation of Prefabricated Parts, Proc. of IEEE International Conference on Image Processing (ICIP), (2013) 3326-3330.
[10] C. Vondrick, A. Khosla, T. Malisiewicz, A. Torralba, Inverting and Visualizing Features for Object Detection, Technical Report 2013.
[11] C. Shen, P. Wang, S. Paisitkriangkrai, A. Van den Hengel, Training Effective Node Classifiers for Cascade Classification, International Journal of Computer Vision, 103 (3) (2012) 326-347.
[12] R. Maania, S. Kalrab, Y. Yanga, Noise robust rotation invariant features for texture classification, Pattern Recognition, 46 (8) (2013) 2103-2116.
[13] K. Maeno, H. Nagahara, A. Shimada, R. Taniguchi, Light Field Distortion Feature for Transparent Object Recognition, IEEE Conference on Computer Vision and Pattern Recognition, (2013) 2786-2793.
[14] D.G. Lowe, Distinctive Image Features from Scale Invariant Keypoints, International Journal of Computer Vision. 60 (2) (2004) 91-110.
[15] K. Mikolajczyk, B. Leibe, B. Schiele, Local Features for Object Class Recognition, Proceedings of the 10th IEEE International Conference on Computer Vision, Beijing, China, 2005, pp. 1792-1799.
[16] H. Bay, T. Tuytelaars, L. Van Gool, SURF: Speeded Up Robust Features, Computer Vision and Image Understanding (CVIU), 110 (3) (2008) 346-359.
[17] N. Dalal, B. Triggs, Histograms of Oriented Gradients for Human Detection, IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1 (2005) 886 - 893.
[18] S.H. Zhang, J.X. Liu, A Self-Occlusion Detection Approach Based on Depth Image Using SVM, International Journal of Advanced Robotic Systems 2012.
[19] F. Mahmood, F. Kunwar, A Self-Organizing Neural Scheme for Door Detection in Different Environments, International Journal of Computer Applications, 60 (9) (2012) 13-18.
[20] M. Zeeshan Zia, M. Stark, K. Schindler, Explicit Occlusion Modeling for 3D Object Class Representations, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2013, pp. 3326- 3333.
[21] Sony Corporation, COLLADA – Digital Asset Schema Release 1.5.0. Technical specifications, 2008.
[22] F. Weichert, D. Bachmann, B. Rudak, D. Fisseler, Analysis of the Accuracy and Robustness of the Leap Motion Controller, Sensors (Basel). 5 (5) (2013) 6380–6393.
[23] J. Guna, G. Jakus, M. Pogačnik, S. Tomažič, J. Sodnik, An Analysis of the Precision and Reliability of the Leap Motion, Sensors, 14(2) (2014) 3702–3720.
[24] A.T. Miller, P.K. Allen, Graspit! A versatile simulator for robotic grasping, IEEE Robotics & Automation Magazine, 11 (4) (2004) 110-122.