Microsoft Kinect Sensor Used to Capture Data for Robotics Applications

Ionel Staretu and Catalin Moldovan

Abstract—**The process of capturing digital images has greatly evolved since the initial appearance of photography in general. In recent years, this evolution has been greatly accelerated by the development of high resolution and specialized digital capture sensors which, in turn, has opened the door for research to develop new products and algorithms allowing imaging to be used as input for controlling different other devices or robots. Still, for there to be a real mapping between a digital image and physical word a lot of research has been done in the field of algorithms and sensors, which have lately resulted in the emergence of affordable and specialized devices on the market like Microsoft Kinect or Motion Leap. Initially, the Microsoft Kinect device was exclusively used for the gaming industry, but later captured the attention of the research community, who quickly noticed that the sensor could be used as a very affordable alternative in the three-dimensional mapping process of space. Soon, an SDK was developed by PrimeSense (OpenNI), which allowed the sensor to be used for any other purpose, not just in the field of games. One of these opportunities is the use of the sensor in the field of image analysis for which a product to capture the movement of a human was developed and is presented in this paper along with a proposal to use the capture mechanism to command and control an industrial robotic arm.**

Index Terms—**Digital images, capture sensors, depth data, Microsoft Kinect, robot arm.**

I. Introduction

Initially, digital imaging has been limited to capturing data from the physical environment using RGB sensors, but with the evolution of microprocessors and computing power, depth data capture has become a necessity and, with the evolution of sensors, has become a reality. Obviously, along with the development of advanced sensors for depth data capture, software applications have been developed to take advantage of sensor research evolution. Among the early researches, which have had promising results, are depth data capture methods using triangulation techniques detailed in [1] or [2]. Other approaches, such as measuring the reaction time from sensor to object and back, defined for the first time in [3] have been somehow successful, but because of the very high acquisition costs of the sensors, this approach was not available to the general public.

At present, the three-dimensional data capture method used in the research is based on a mixed approach, that is, it takes advantage, on the one hand, of the image processing evolution and, on the other hand, of the evolution in sensor technology. In this regard, affordable and highly accurate devices have come onto the market, including: Microsoft Kinect and Motion Leap.

A presentation of the relative recent approaches to research on robotic manipulation systems using robotic arms equipped with anthropomorphic grippers can be found in [4], and in [5] a presentation of the existing control methods of mobile robots based on digital image processing algorithms can be found.

Manuscript received October 5, 2018; revised December 23, 2018. This work was supported in part by the CLOOS in Germany and its representative in Romania, Timisoara ROBCON Company.

I. Staretu and C. Moldovan are with the Transilvania University of Brasov, Brasov, Romania and the Academy of Technical Sciences of Romania, Bucharest, Romania (e-mail: istaretu@yahoo.com, mcatalin1983@yahoo.com).

II. Systematizing Methods Used for Three-Dimensional Image Capture

Currently, there are two techniques used to capture three-dimensional data from physical environments. They are classified by capture mode or type of sensors used in *active techniques* and *passive techniques*.

The active mode refers to the use of light projections (flight time) or light patterns (structured light) on a particular type of environment, then measuring the speed at which the light returns to the sensor or the distortion of the template in the environment for the depth calculation [6].

The passive mode refers to the use of methods for examining an image from two different angles, the depth calculation being based on the analysis of points from the two different angles using geometric algorithms.

Into the following paragraphs the triangulation (both active and passive) is introduced and explained how this is used and implemented into a .NET application to digitize the movement of a human hand and how this could be used to control an industrial robot.

A. Triangulation or Stereo Vision

Triangulation refers to the process of determining the depth of a point in three-dimensional space considering as input parameters different projections of the environment. Triangulation can be active or passive [1].

In order to solve the problem of passive triangulation it is necessary to know in advance both the parameters of the cameras which capture the image and the functions of translating the three-dimensional space into two-dimensional space. Knowing these parameters, the distance is calculated using triangulation between the positions of the two cameras and the pixel matching in the captured images (see Fig. 1).

In the Fig. 1, the values are the following: Lc and Rc are the two cameras with parallel optical axes and f is the focal length; d - is the distance between the two cameras (the distance between the two centers) and is perpendicular to the optical axes; XZ is the plane where the two optical axes are located, and XY is the plane parallel to the plane of the image; the X axis is the same as the distance d; Lc - the origin of the reference system that is at the center of the left camera.

DOI: 10.7763/IJMO.2019.V9.681

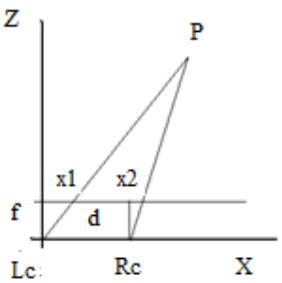

Fig. 1. Representation of triangulation.

Using the above parameters, triangulation equations become:

$$Z = \frac{d \times f}{x_1 - x_2} \quad Y = y_1 \times \frac{Z}{f} \quad X = x_1 \times \frac{Z}{f} \tag{1}$$

B. Structured Light

Structured light is a form of active triangulation [2]. The depth measurement method using this technique consists of designing a predefined template in an environment and then applying the triangulation equations between the captured template and the reference template. A structured light based detection system consists of a light emitter that can project a template and an RGB (CCD or CMOS) sensor used for image detection.

C. Flight Time (TOF Camera)

Flight time measurement is another method of determining depth in images and consists of using a projected light source on a surface. Determining the distance from the transmitter to the object is thus a function of time [3]. The distance is calculated by the time difference between the light pulse emission and the sensor detection of reflected light using the following formula:

$$D = \frac{c \times \delta(t)}{2} \tag{2}$$

in which : c - represents the speed of light as constant; $\delta(t)$ - represents the time measured between the light emitted and the light being detected; $c * \delta(t)$ - is divided by 2 because the distance is covered (from the transmitter to the receiver) twice in the environment; D - represents the calculated distance as a measure of the reflected signal delay.

III. Systematization of the Hardwere Systems used for Capturing Images and Depth Data

An image capture sensor can be described as a system that performs the networking of the coordinates in the physical space in planar coordinates ($\Re_3 -> \Re_2$). If an image capture sensor is used for human hand detection and possibly gesture recognition, a transformation model is essential to solve the image analysis problem. Among research issues in line with the purpose of this paper, there is the issue of hand position detection. Fig. 2 illustrates the projection of a point P from the real space in a point p in the plane of the image.

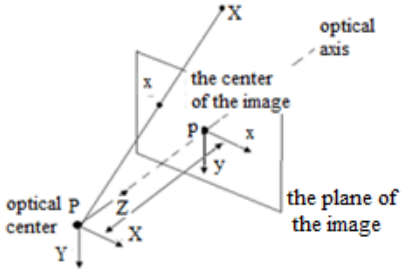

Fig. 2. Model of image capture sensor. Translating the three-dimensional frame into two-dimensional. Figure adjusted after [7].

IV. Microsoft Kinect Device

The Microsoft Kinect device was released on the market by Microsoft by the end of 2010. It was originally used as an accessory for the XBox console. Since its inception, Microsoft Kinect has been a resounding success, with sales of about 10 million units [8] estimated in the first year. Initially, automated robots using Microsoft Kinect type sensors lacked retroaction, but this shortcoming was easily overcome by creating a solution that uses the three-dimensional environment mapping process and defining areas inaccessible to the robot and sending retroaction to the control system for information [9]. This method is used successfully in Stowers' work [10], which shows how robots can be programmed to fly autonomously without hitting other objects.

From a constructive point of view, the Kinect sensor consists of the following components (see Fig. 3): infrared sensor: transmitter and receiver. The transmitter projects a light pattern on a surface, which is then captured by the receiver; RGB camera: which stores data on three channels (RGB) at the resolution of 1280X960 and the frequency of 30 Hz. The visualization field of the Kinect sensor, as specified in Microsoft documentation [8], is 43 degrees vertical and 57 horizontal. The sensor can track people with a 1cm accuracy at a distance of 2m [11]; a system of four microphones to capture sound from different positions; a motor used to tilt the sensor without physical interaction between the user and the sensor; an accelerometer for detecting the current inclination of the sensor relative to the horizon line.

The field of visibility and resolution changes as the distance between the object and the Microsoft Kinect sensor [12] changes, so the field of visibility increases linearly with the distance and the resolution decreases along *x* and *y* direction with increasing distance.

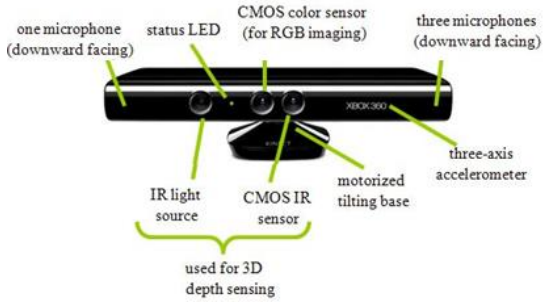

Fig. 3. The elements of the microsoft kinect device.

From a functional point of view, the monochrome CMOS sensor along with the "depth sensor" analyzes the captured image and creates a three-dimensional map visualization field (see Fig. 4).

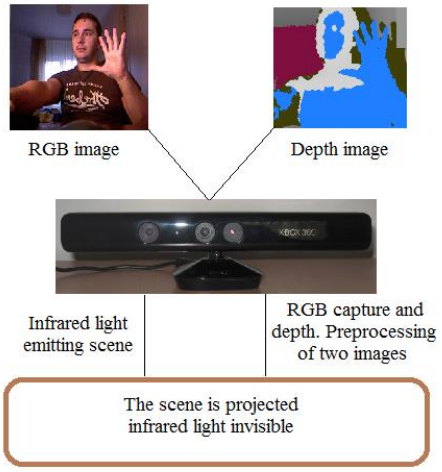

Fig. 4. The invisible IR light is emitted and monitored using the CMOS image sensor. The image processor generates the depth image.

The mix between the monochrome CMOS sensor with the depth one can acquire the image and movement under any ambient light conditions [12]. The depth sensor is adjustable, the SDK that comes with the Microsoft Kinect sensor being able of self-calibration based on the physical environment or on the presence of other physical obstacles, etc. The microphone system is used to detect the location of a voice and to counteract the ambient noise. All of these sensors offer multiple body recognition capabilities in three-dimensional mode, and body motion, facial recognition and voice recognition. The process of obtaining a depth image consists of simultaneously capturing two images, namely RGB image using the RGB CMOS sensor, and the depth image captured by the monochrome CMOS sensor (see Fig. 4).

A. *Analysis of the Depth Measurement Process*

The Microsoft Kinect device uses a three-dimensional space mapping process by joining the depth data to each captured pixel. The value attached to each pixel is the distance from the sensor to the object facing the sensor in the direction of the sensor orientation [13]. To estimate the depth of each pixel individually, the Kinect sensor uses the concept presented above, of structured light. Thus, the infrared sensor emits a known template in the environment, then, based on the data captured by the monochrome sensor, the internal Kinect sensor algorithm attaches a value to each pixel (see Fig. 5).

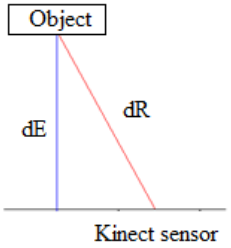

Fig. 5. Estimated distance (dE) and real (dR) between the Microsoft Kinect device and a random object. Image adapted after [13].

where: dE is the estimated distance by Kinect sensor between the object and the sensor; dR represents the actual distance between the Kinect sensor and the object. Based on the depth estimation mode, it can be concluded that the device becomes an efficient way to capture the coordinates (x, y, z) of any three-dimensional object, but there is, however, a difference between the estimated distance and the actual distance (see Fig. 5).

V. Using the Microsoft Kinect Sensor to Capture Human Hand Movements

In digital image processing research, recognition of three-dimensional objects involves the recognition and the determination of three-dimensional objects in an image or frame, part of a video stream. This recognition can be performed in real time, or it can be executed on a video stream that is previously captured and saved in memory. The Microsoft Kinect device combines a set of hardware and software mechanisms that builds a digital, three-dimensional representation of a physical environment. Algorithms created for the recognition of three-dimensional objects, based on the data captured by the Microsoft Kinect sensor, analyze two types of data in parallel: RGB image and image depth data. In this paper, an application has been developed that attempts to overcome limitations of capture with the Web camera, namely, depth capture for Z and Y axes. For the development of the application, Windows SDK and the Natural User Interface (NUI) library were used. In Fig. 6, the conceptually proposed system is exemplified.

To develop the three - dimensional data capture system, four steps were considered and implemented. They are: **Initialization**: where the Kinect sensor driver is loaded into memory; **Detection**: where the system is detecting the human hand in order to be able to later on recognize gestures; **Interpretation and recognition**: Where the system will interpret each image which contains a human hand and digitize the captured gesture and **Visualization**: The recognized gesture must be sent to a virtual simulator to control the a virtual anthropomorphic gripper.

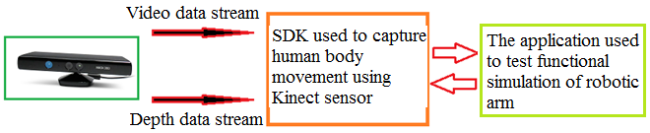

Fig. 6. Interaction between Kinect sensor and the functional simulation test application.

The **system initialization** means that the Microsoft Kinect driver is loaded in computer memory and the sensor is initialized. For this operation, Microsoft Kinect SDK library was used, which allows a user to get a logical instance of the Microsoft Kinect sensor that can be programmatically worked on. Microsoft Kinect SDK library also allows, in addition to initializing the sensor, a way for a user's actions on the body to be recognized implicitly at the arm level and even at the human hand level (Fig. 7). Using the Microsoft Kinect SDK library, an application can localize up to 20 user joints in parallel. Once the system is initialized, the Microsoft Kinect SDK library maps each joint in the three-dimensional space, making the x, y, z coordinates of each joint accessible to the programmer.

For system development, a Visual Studio solution was created using Microsoft C#. The created solution references the Microsoft.Kinect.DLL object that is later used by the C#

compiler to make system function calls to use the hardware capabilities of the Microsoft Kinect sensor. To initialize the sensor, the following code section is used by declaring, initializing, and creating an instance of a KinectSensor parameter : KinectSensor _sensor . Because several Microsoft Kinect sensors can be connected to a computer, it is checked if the set of sensors has at least 1 element, in which case the KinectSensor class is instantiated using the instruction:

if(KinectSensor.KinectSensors.Count> 0)
_sensor = KinectSensor.KinectSensors [0];

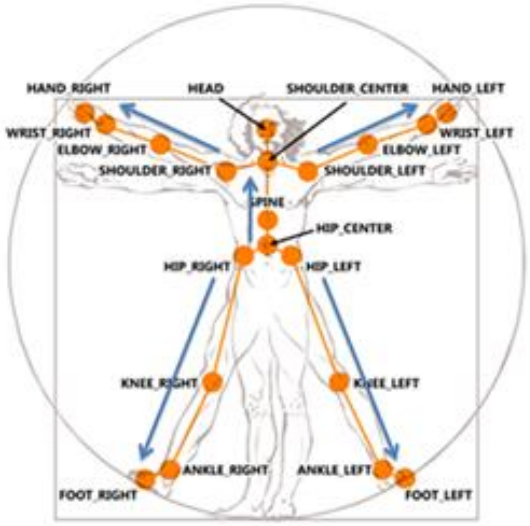

Fig. 7. Joints that can be detected using the Kinect device [14].

The KinectSensor class allows the commands processing by the user. Once the *_sensor* instance is created, the application can capture depth data, RGB images, sounds, or the human skeleton structure in different data streams. These streams are: The **ColorStream**: which is a property of the flow of images in different resolutions or formats. The format determines how the image is encoded, RGB, YUV or Bayer; The **DepthStream**: is a video stream property that contains image depth data for each frame. Depth flow consists of pixels that contain the distance (in mm) from the plane of the sensor to the nearest object. An application that uses depth stream can track human hand movements and identify background image; The **SkeletonStream**: represents a collection of properties, including: (1) TrackingState: which is a Boolean property that verifies if an object is tracked or not. (2) Joints property - represents a collection of parts of the human body detected (see Fig. 7). (3) TrackingID property: represents a unique identifier of a detected user. (4) Position property: which represents the global position of a user through Microsoft Kinect sensor reference frame. (5) ClippedEdges property: means that part of a user's body is not completely in the Kinect sensor field of view.

Using the KinectSensor class start() method, initialization of depth and RGB cameras is performed:

_sensor.Start ();

Following the call of the start() method, the Kinect sensor starts capturing RGB and depth frames (Fig. 8) that are transmitted to the recognition step.

To track the user's movements, the Skeleton class is used, which can be adjusted for each frame, minimizing the jitter rate and stabilizing the objects tracked along a video stream. Microsoft Kinect SDK provides a mechanism for applying an object position smoothing filter to a frame of a video stream.

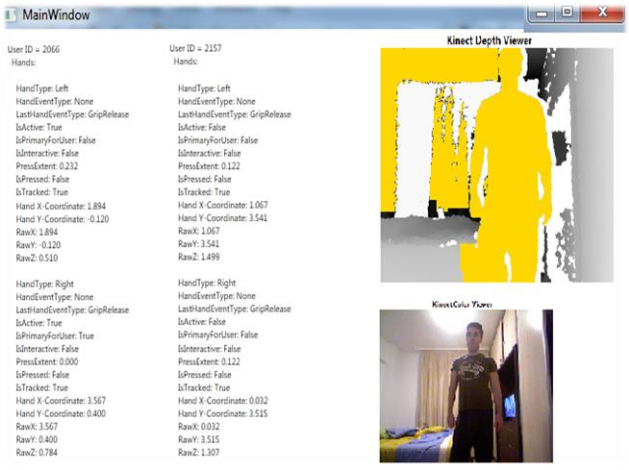

Fig. 8. RGB and depth data captured using the Kinect sensor.

To initiate a user tracking process through Kinect sensor, some parameters must be set with baseline values. If the parameters are not set properly, the application will not properly filter the captured data, so each frame will have noise joining the images.

As a result of the tests performed, the following empirical values of smoothing parameters for human hand detection were found, namely: Smoothing = 0.3f, Correction = 0.0f, Prediction = 0.0f, JitterRadius = 1.0f, MaxDeviationRadius = 0.5f.

VI. Future Work

In order to automate the process of capturing depth data and use it for the control of a robotic arm ABB endowed with a gripper, this paper set up the initial concept to be implemented (see Fig. 9). The Kinect device could be used to detect the movement of human arm and then, to transmit its movements to the industrial robotic arm. In order to control such an ABB robot, the ABBCOMMANDER module was defined conceptually, which, once implemented, can capture the movement of the human arm and conveys the gesture to the ABB robotic arm (see Fig. 9).

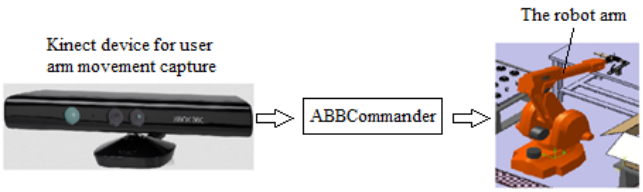

Fig. 9. The concept of robotic arm control.

VII. Conclusions

This paper briefly presents the main methods of capturing depth data and shows how the Microsoft Kinect device can be used to capture depth data and how to use robotic applications to control a robotic arm by capturing human arm movements.

The presented issues can be used with minimal adaptations and for more advanced Kinect sensor variations that are

already underway or will be carried out in the future for similar applications.

ACKNOWLEDGMENT

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] Y. B. Mahdy, K. F. Hussain, and M. A. Abdel-Majid, "Projector calibration using passive stereo and triangulation," *International Journal of Future Computer and Communication*, vol. 2, no. 5, 2013.
[2] Q. Chen, D. Li, and C. Tang, "Compressive structured light for recovering inhomogeneous participating media," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, Vol. 35, No. 3, 2013.
[3] M. Guilhaus, "Special feature: Tutorial principles and instrumentation in time-of-fight mass spectrometry physical and instrumental concepts," *Journal of Mass Spectrometry*, vol. 30, pp. 1519-1532, 1995.
[4] H. Yousefa, M. Boukallela, and K. Althoeferb, "Tactile sensing for dexterous in-hand manipulation in robotics — A review," *Sensors and Actuators A: Physical*, vol. 167, no. 2, pp. 171-187, 2011.
[5] R. S. Pieters, "Direct methods for vision-based robot control: Application and implementation," PhD thesis, Eindhoven University of Technology, The Netherlands, 2013.
[6] A. Butler, S. Izadi, O. Hilliges, D. Molyneaux, S. Hodges, and D. Kim, "Shake'n'Sense: Reducing interference for overlapping structured light depth cameras," in *Proc. CHI '12 SIGCHI Conference on Human Factors in Computing Systems*, pp. 1933-1936, New York, USA, 2012.
[7] R. Nq, "Digital lightfield photography," PhD Thesis, Stanford University, 2006.
[8] Information. [Online]. Available: http://msdn.microsoft.com/ en-us/library/hh438998.
[9] F. Ryden, H. Chizeck, S. N. Kosari, H. King, and B. Hannaford, "Using kinect and a haptic interface for implementation of real-time virtual fixtures," in *Proc. 2nd Workshop on RGB-D: Advanced Reasoning with Depth Cameras (in conjunction with RSS 2011)*, Los Angeles, USA, 2011.
[10] J. Stowers, M. Hayes, and A. Bainbridge-Smith, "Altitude control of a quadrotor helicopter using depth map from microsoft kinect sensor," in *Proc. IEEE International Conference on In Mechatronics (ICM)*, pp. 358-362, 2011.
[11] T. Osunkoya and J. C. Chern, "Gesture-based human — Computer -interaction using Kinect for windows mouse control and powerpoint presentation," in *Proc. the 46th Midwest Instruction and Computing Symposium (MICS 2013)*, La Crosse, Wisconsin, USA, 2013.
[12] M. R. Andersen, T. Jensen, P. Lisouski, A. K. Mortensen, M. K. Hansen, T. Gregersen, and P. Ahrendt, "Kinect depth sensor evaluation for computer vision applications," *Electrical and Computer Engineering Technical Report ECE-TR-6* , 2012.
[13] M. T. Draelos, "The kinect up close: Modifications for short-range depth imaging," in *Proc. IEEE Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, pp. 251 - 256, Hamburg, Germany, 2012.
[14] Uncategorized. [Online]. Available: http:// gmv.cast.uark.edu/ uncategorized/ working-with-data-from-thekinect/ attachment/kinect-sensors-on – human -body/

Ionel Staretu was born in Rusavat, Buzau, Romania, on April 16, 1957. He is a graduate of the TCM Faculty of the Transylvania University of Brasov (1983), Romania. He obtained a PhD degree in industrial robots specialization in 1995. Specializations in: Tribology (Transylvania University of Braşov-1990), Robotique et Productique (INSTN of Saclay, France-1992/1993), Organization Management (IAI and Transylvania University of Braşov -1999/2000), Quality Management (2003) and Quality Audit (2004) at the Transilvania University of Brasov. Since 2003, he is an Outsourced Technical Expert and Certified Consultant by CERTEXPERT Bucharest and A.E.X.E.A. Paris.

He has been working at the Department of Product Design and Robotics since 1985, currently the Department of Product Design, Mechatronics and Environment at the Transylvania University in Brasov. He published: 6 books (Gripping systems in USA, 2011); Sisteme de prehensiune, Romania, 2006, 2010; Elements of medical robotics and prosthesis, Romania, 2005), 5 didactic works and over 220 scientific articles in the country and abroad. He is the author or co-author of 11 patents. He has helped solve over 28 national and international scientific research grants (at 4 as grant director).

Prof. Staretu is the President of the Brasov Branch of the Romanian Society of Robotics, vice president of AGIR Braşov Branch, member of ARoTMM and expert in Robotics of the Academic Society of Romania, member CRIFŞT-Romanian Academy; member of the Committee of Publishers in Romania and abroad (USA, Serbia, India) and Scientific Committees at national and international scientific events. He is a PhD supervisor in the field of Industrial Engineering. Since 2017 he is a correspondent member of the Academy of Technical Sciences of Romania-ASTR.

Catalin Moldovan was born in Brasov, Romania on 17 May 1983. He is a graduate of the Transylvania University of Brasov, the Faculty of Mathematics and Computer Science, the bachelor's degree in informatics (2007) and the master of algorithms and software products (2008). He was admitted to PhD in the Industrial Engineering field in 2009 and was awarded the PhD degree in 2014.

He was employed by several companies in Brasov, first in Dynamic Ventures, then at IBM Romania, IT Software Department, where he is currently active. He has dealt with several themes in the field of body modeling in virtual reality, has developed several software modules for commanding an anthropomorphic gripper in virtual reality and an anthropomorphic five-finger gripper, made by rapid prototyping. He published 18 ISI or BDI indexed papers in the prestigious conference proceedings: Robotics 2012, OPTIROB, RAAD, or in magazines: AGIR Newsletter, Univ.Dunarea de Jos in Galaţi, Mechanical Engineering series; IJARS, etc.