International Journal of Engineering Science Invention (IJESI)
ISSN (Online): 2319 – 6734, ISSN (Print): 2319 – 6726
www.ijesi.org ||Volume 8 Issue 03 Series. I || March 2019 || PP 65-72

Using RGB-Depth Cameras and AI Object Recognition for Enhancing Images with Haptic Features

George Kokkonis[1], Vasileios Moysiadis[2], Sotirios Kontogiannis[3], Dimitrios Tsiamitros[4]

[1]Dept of Business Administration, Western Macedonia University of Applied Sciences, Greece
[2]Dept of Informatics and Telecommunications Engineering, University of Western Macedonia, Greece
[3]Department of Mathematics, University of Ioannina, Greece
[4]Dept of Electrical Engineering, Western Macedonia University of Applied Sciences, Greece
Corresponding Author: George Kokkonis

Abstract: *This paper presents the methodology for enchasing images with haptic features. It presents two methods of enhancement, one with the use of RGB Depth Cameras and one with the use of artificial intelligence and object recognition. The algorithm that is used for the object recognition is the Mask R-CNN. Once the objects are recognized, are used as import to an open source software for image manipulation, the OpenCV. Specific geometrical patterns are used to enhance the object recognized with haptic features. Performance evaluation between the two methods is undertaken, in order to reveal which method provides better experience to the user of the haptic interface.*

Keywords– *Haptics, haptic interfaces, haptic patterns, Mask R-CNN, OpenCV, RGB-Depth Cameras*

Date of Submission: 17-02-2019 Date of acceptance: 03-03-2019

I. INTRODUCTION

Touch is a basic sense for understanding our environment. The sensation of touch involves the ability of the skin to understand the surface properties of an object and determine its position and shape. Visually impaired people depend on the sense of touch to explore their surroundings and communicate.

The reaction that a user gets every time he explores the shape, the texture, the geometry, the mass, the elasticity and the dimensions of the objects with his hands is called tactile feedback.

Tactile interfaces are used for exploration of 3D digital objects in virtual and augmented reality. These devices are often connected to the network and carry haptic data over the Internet. Specific protocols are used to interconnect haptic interfaces via the network [1]. To improve the operation of haptic interfaces, the sampling frequency of haptic data, quantization, compression, encoding, and the significance of each haptic data has been studied [2]. Tactile interface try to maximize its user experience, taking into account the network status and the significance of haptic data [3].

The authors present two methods for enhancing images with haptic features. The Artificial Intelligence (AI) R-CNN algorithm and the RGB-Depth cameras are used to automatically detect objects. The open source OpenCV image editor is used to assign these objects geometrical patterns. The enhanced images are imported to the open source haptic software development Kit (SDK) H3D and ascribe them tactile properties. With the use of the Depth Mapping algorithm, the images are transformed to tactile interfaces. Haptic devices are used to explore the haptic images and perform an evaluation test. The metric for the performance evaluation is the Mean Opinion Score which take into consideration the Quality of Experience (QoE) of the User.

The rest of the paper is organized as follows. Section II presents and describes the operation of RGB-Depth Cameras. Section III analyses the artificial intelligence algorithm for object recognition. Section IV explains the enhancement of the images with geometrical patterns with the use of the OpenCV. Section V describes the assignment of haptic properties to images with H3D Depth Mapping. Section VI performs an evaluation of the two methods presented for the enhancement of images with tactile information. Finally, section VII concludes the paper.

II. 3D RECONSTRUCTION WITH RGB-DEPTH CAMERAS

The recording of our surroundings was until recently done only in two dimensions with 2D RGB cameras. With the evolution of 3D cameras, the recording of the surrounding was transformed from two dimensions to three. RGB-Depth Cameras can be used to record our surroundings in three dimensions and transform simple figures to 3D Digital objects [4]. 3D objects obtains entity and can be manipulated independently in virtual reality. The first RGB –Depth sensor that was used was the Microsoft Kinect Xbox 360 sensor [5]. It is a low cost sensor that is has been widely used for scientific and entertainment purposes. It uses a RGB camera with 8-bit VGA resolution (640 × 480 pixels). The depth information is recorded from a 11-bit resolution stream of 640 × 480 pixels. This means that it provides a 2048 levels of depth sensitivity for each pixel. In order to capture the depth information, a projector emits infrared light to the target. The infrared light is been scattered by target and returns back to the depth sensor of Kinect. The sensor records the time spent for light to travel from the projector to the target and back. This time is called "Time of Flight". Given that the speed of light is known, the "Time of Flight" call reveal the distance between the Kinect sensor and the scanned object. Microsoft provided the community with the open source software development kit for developing applications for the above sensor, the "Kinect for Windows". With the use of specific algorithms, 3D geometry reconstruction of static environments can be developed [6].A sample of this reconstruction is depicted in Figure 1. A simple cup, placed on the floor, has been scattered and imported to Kinect Fusion for the reconstruction of its geometric.

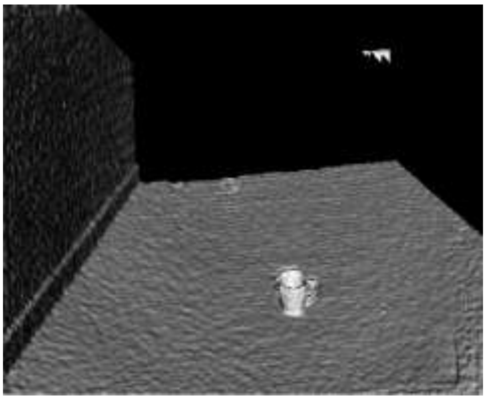


Figure 1. 3D Object reconstruction with Kinect Fusion algorithm.

III. AI Object Recognition with TensorFlow And Mask R-CNN

The core deep learning implementations are on the fields of vision text and speech recognition. Focusing on vision, object recognition and image features extraction are the breakthrough areas where novel neural network algorithms and support vector classifiers are implemented. In this direction, the need for machine learning frameworks for algorithms development started to appear with Theano [7] as the first framework in this direction, followed by TensorFlow framework [8].

TensorFlow is a software application, popular for implementing machine learning algorithms-processes and neural network models. It was developed by google and was released as an open source platform in 2015. It takes as input multidimensional array (tensors) and constructs, trains and maintains a flowchart of operations to be performed on that input. The main advantage of TensorFlow is that it can run on different platforms as well as GPUs. TensorFlow was written in C++ but can be accessed by many language wrappers, especially Python. Furthermore, TensorFlow includes a Tensorboard for visually monitoring each instantiated flow process [8].

On top of TensorFlow framework for imagery features extraction and models high level implementation, lay out interfaces such as Keras [9]. Keras is a wrapper of multiple frameworks such as Theano or TensorFlow. When building and training a model, Keras offers interaction with implemented TensorFlow processes (flows). The layered methodology that Keras uses, includes the definition of a deep learning network-model (that may include multiple TensorFlow processes), network compilation, fit on training data, evaluate network on test data and then perform derived by model predictions.

Real time object detection is also mentioned as instance segmentation. That is, the process of detecting each distinct object of an image. Instance segmentation problem is a combination of two sub problems: Bounding areas object detection (called as Regions Of Interest –ROIs) that deals with the selective identification and classification of image objects. The second part of instance segmentation is semantic segmentation. This is the understanding of an image at the pixel level and the assignment of each pixel to an appropriate instructed by dataset classification class, by delineating the boundaries of each ROI. Using bounding areas detection and semantic segmentation shaded masks together, the corresponding outcome is instance segmentation.

Focusing on bounding object detection, challenge in computer vision and specifically in terms of accuracy and real-time recognition, existing exhaustive search methods failed to comply with the requirement of a fast and close to real-time feature extraction. Towards this direction the proposed selective search method [10] success as a region proposal process lead to the implementation of the R-CNN algorithm [11]. The R-CNN algorithm classifies bounded image areas extracted from proposed regions (called as proposed region networks – PRNs), by applying on per region Convolution Neural Network process (CNNs), inferred by training image datasets to feature-full detected region proposals (RPNs- areas of high probability in containing an object, selected by custom region proposal methods and selective search). Even if R-CNN original implementation was relatively slow in respect to real-time object detections, further algorithm improvements such as Fast and Faster R-CNN lead to time speedups for image detection from 25x up to 250x times faster. More particularly, R-CNN algorithm uses multiple CNN networks for each original image selective search RPN network ROIs. For improving speed, Fast R-CNN method performs once image CNN first and then calculates the ROIs, while Faster-RCNN, maintains a feature map after the CNN process and before the ROIs' calculation, as part of the region proposal network process [12].

As an improvement of Faster R-CNN, Masked Region based convolution neural networks approach (Mask R-CNN) focuses on the problem of instance segmentation [13]. The Mask R-CNN method adds to the Faster R-CNN an object mask prediction process in parallel with the existing bounding box R-CNN process. Mask R-CNN is simple to train and adds only a small overhead to Faster R-CNN, running at 5 fps [13]. In detail, Mask R-CNN initiates at first a Faster-RCNN process [11] for bounding box object detection and classification (ROI pooling or ROI align). Then it outputs the object mask by performing a pixel by pixel alignment on each ROI. These masks are binary masks outputted for each ROI. Computation of this masks is performed in parallel with the ROI classification process. For the purpose of mask computation, Fully Convolution Networks are used on classified images taken from the Facebook Coco dataset [14], [15] in order for the object mask to be extracted. A Mask R-CNN implementation for keras, using TensorFlow framework has been implemented at [16] and used by the authors.

IV. ENHANCE DIGITAL IMAGES WITH GEOMETRICAL PATTERNS WITH OPENCV

The main concept of the proposed approach is to convert images to geometric patterns, in order to be enhanced with tactile information. A linear conversion is used to create a number of grayscales levels, each one will be assigned to a distinct patterns. Python is used as a programming language to enhance the captured images with tactile attributes, based on specific patterns. The open source library OpenCV is used for image processing which is cross-platform and includes various computer vision techniques.

Python code has also been used to create twelve independent geometrical patterns in black and white, as shown in Figure 2. Most of these patterns have been proposed for haptic enhancement in [17]. Numpy library is used for this propose by filling two-dimensional arrays with ones and zeros representing black and white colors.

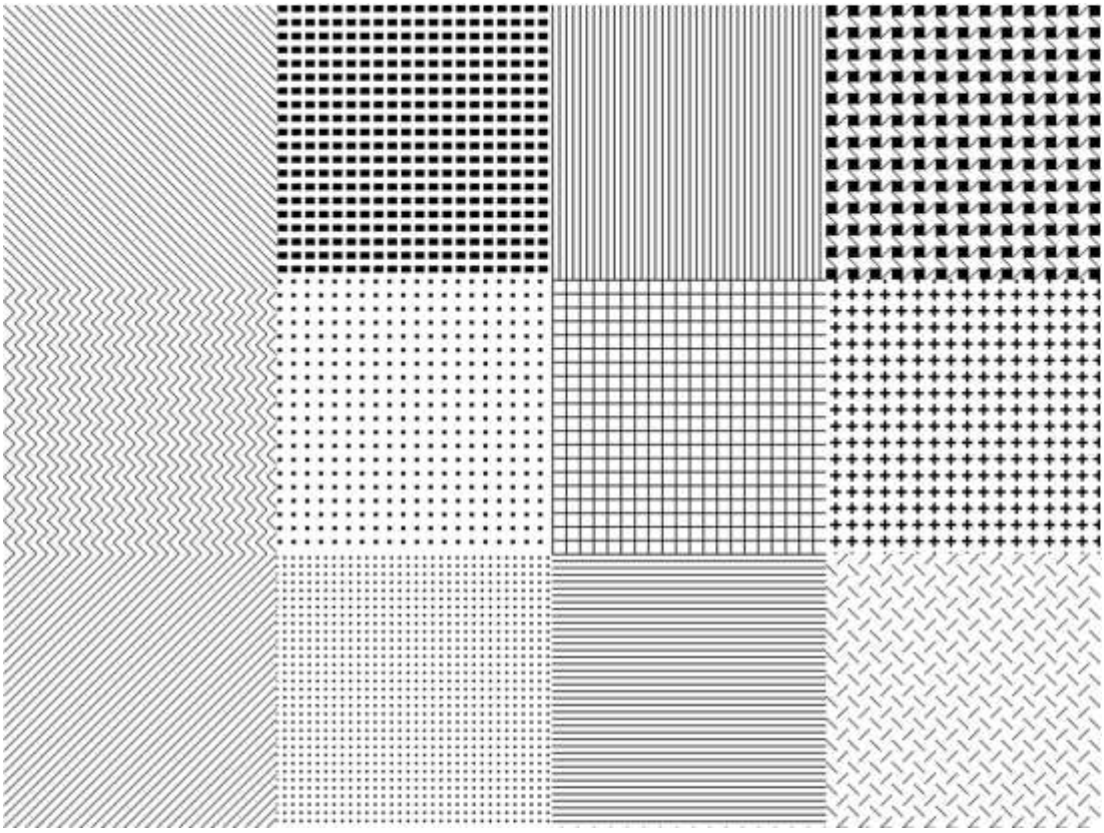


Figure 2 - Programmatically implemented patterns for use in tactile images.

A. Linear Implementation

The linear implementation divides the grayscale levels [0,255] to n equal segments, where n is the desired independent levels in the final haptic image.

Figure 3 - Grayscale segmentations for n = 12 patterns.

n = 12	0 - 22	23-44	45-66	67-88	89-110	111-132	133-154	155-176	177-198	199-220	221-242	243-255

Initially, the depth image generated from the RGB-Depth camera is converted in grayscale. Then, based on the desired number of patterns, different levels of thresholds are being applied on the image, and a mask is creating based on differences between two continuously threshold images. For each mask, the corresponding pattern is applied in the haptic image and the final result consists of the combination of them. The pseudocode of this approach is shown in Algorithm I.

Algorithm I - Pseudocode of the linear approach

```
Read depth image
Convert depth image to grayscale
Create n patterns with numpy
Calculate n segments in [0,255]
Create a blank haptic image H
foreach segment i do
  Create mask i based on pixels belongs to segment
  Apply corresponding pattern in the haptic image H with mask i
```

An example of the implemented algorithm is shown in Figure 4, where Figure 4(a) is the original depth image, Figure 4(b) is the transformed image with 12 segmentation levels.

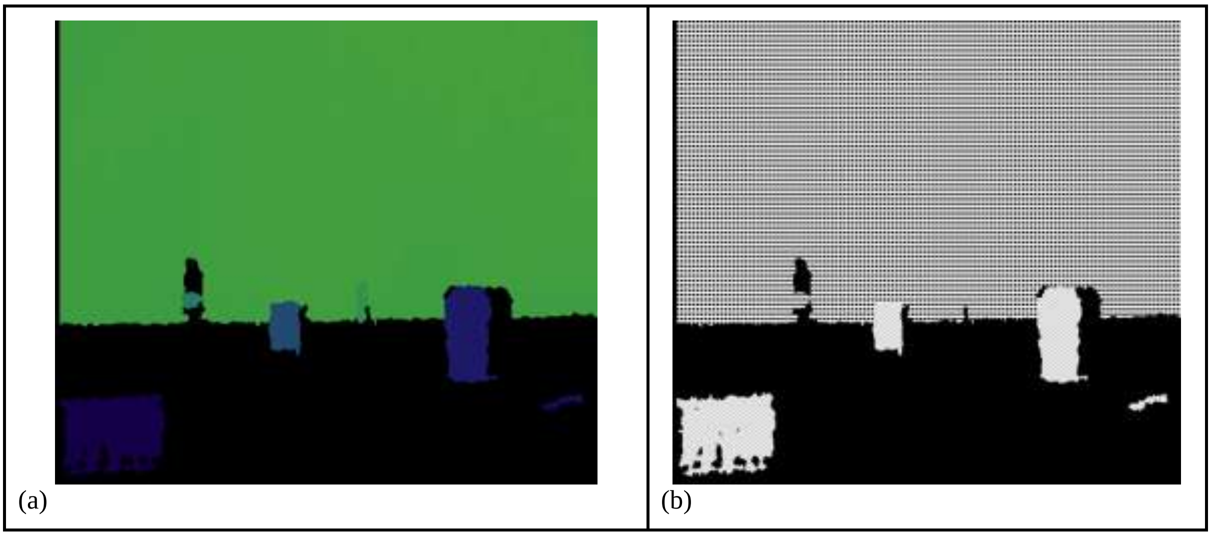


Figure 4 - Samples of the linear approach where (a) is the original depth image, (b) is the transformed image with 12 segmentation levels

V. ASSIGN HAPTIC PROPERTIES TO IMAGES WITH H3D DEPTH MAPPING

In order to attribute haptic properties to the images of Figure 4(b) the open source software H3D has been used. H3D is developed in C++ and uses the OpenGL library for graphics rendering and HAPI for haptics rendering. It uses the open standard ISO-ratified X3D syntax in order to create and represent 3D virtual objects, stereo graphics with haptics properties. In order to represent more complicated 3D Objects and complex haptic properties the Python and C++ programming can be used. HAPI uses the XML syntax, or python, or C++ programming to enhance 3D objects with haptic features. The 3D objects are described as nodes. Node is the basic element for the representation of 3D objects. The geometry of the nodes is described through 3D mesh polygons. The haptic forces between the nodes are represented with predefined force effects inside the HAPI library. Its node is assigned with haptic attributes as mass, static and kinetic friction, and stiffness [18]. The visual and geometric attributes of the nodes, such as the shape, color, appearance, material, are inherited by the

X3D ISO. A sample of the Xml code that attributes the haptic properties to the nodes of our experiments is depicted in the following XML code.

```
<DepthMapSurface
staticFriction="0.7"
dynamicFriction="0.7"
stiffness="0.7"
maxDepth="0.01"
whiteIsOut="False" >
<ImageTexture containerField="depthMap" url="pattern.png" repeatS="false" repeatT="false"/>
</DepthMapSurface>
```

Figure 6 - DepthMapSurface properties of the Haptic nodes.

The depthmapSurface is a technique that corresponds the color of the node to the depth of its depthmap. The whiter the node is the bigger the protuberance of the node will be if the property of whiteisout is true and vice versa. The ImageTexture property assigns the texture/pattern image to the node. This texture/pattern is used for the mapping of the color to the protuberance. The maximum height that a protuberance can get is assigned with the property maxDepth.

In the performance evaluation test described in section VI, the values for the haptic attributes as given in Figure 6. The pattern.png file is created dynamical with and the patterns that are given to each node is given in Figure 2.

The haptic devise that was used for the performance evaluation test was the Novint Falcon, which is depicted in Figure 7. It is a common haptic device with 3 Degrees of Freedom (DOF) that was released in 2007. It is connected to the computer through USB2 interface, it offers a sampling rate of 1000 packets per second and sub-millimeter position resolution. It has 10x10x10 cm of 3D touch space and can exert up to 8.9 Newtons of force [19].

Figure 7–The Novint Falcon Haptic Device.

VI. Performance Evaluation Between RGB-Depth Cameras And AI Object Recognition

For the performance evaluation of the two methods of sections II and III, the Mean Opinion Score (MOS) metric was used. MOS measures the User Experience (UX) [20] of the user that uses a haptic device to touch and haptically explore the 3D objects that were created with the methods describe in sections III, IV and V. The sample users were 10 students at the age between 18 and 24. Five of them were male and the other five female. All of them were postgraduate students. Before the test, they were given 5 minutes to explore and get familiar with the use of the haptic device.

The initial scene that was rendered with the Kinect depth sensor and the Mask-RCNN algorithm is depicted in figure 8.

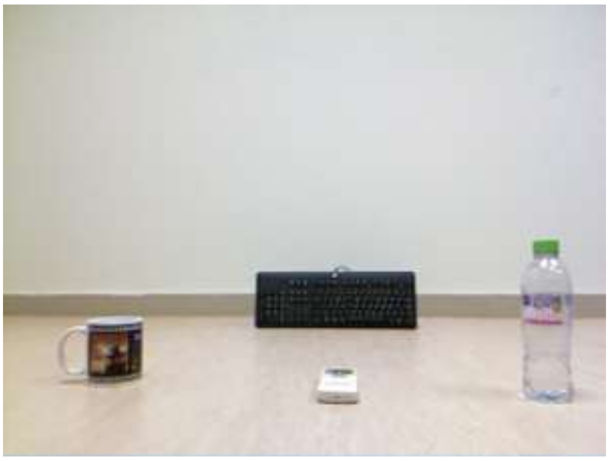


Figure 8–The scene that is sued for performance evaluation.

The image that was created with the Mask R-CNN AI algorithm when the figure 8 was used as import and the pre-trained COCO weights, mask_rcnn_coco.h5 found in [21], were used as a training set, are depicted in figure 9.

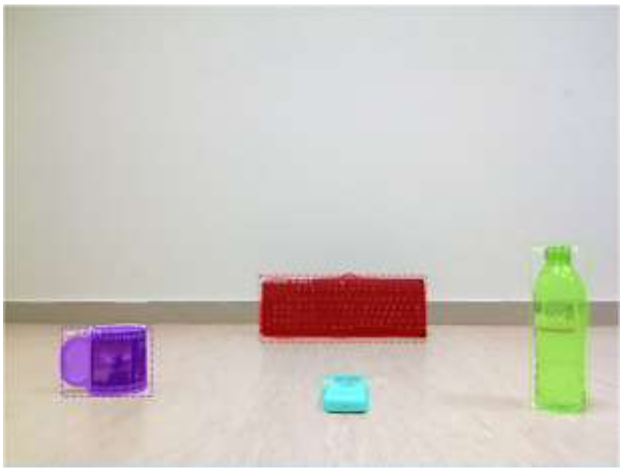


Figure 9–The Mask R-CNN object recognition of the scene in figure 8.

The depth image that was created with RGB-Depth sensor Microsoft Kinect is depicted in figure 10.

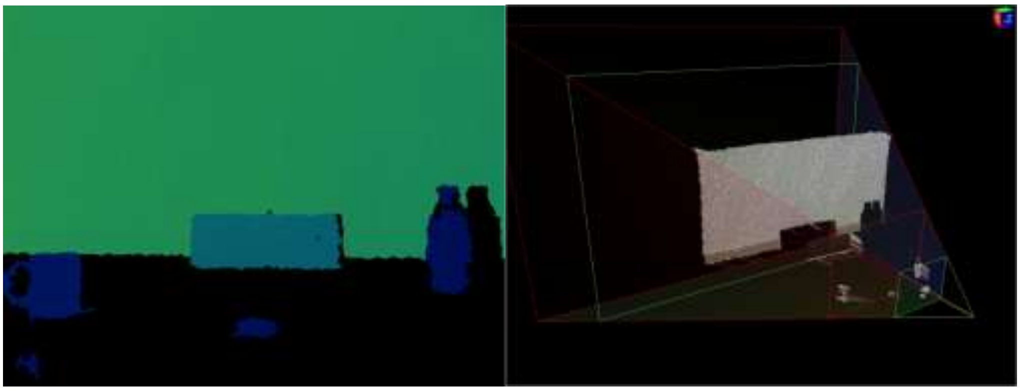
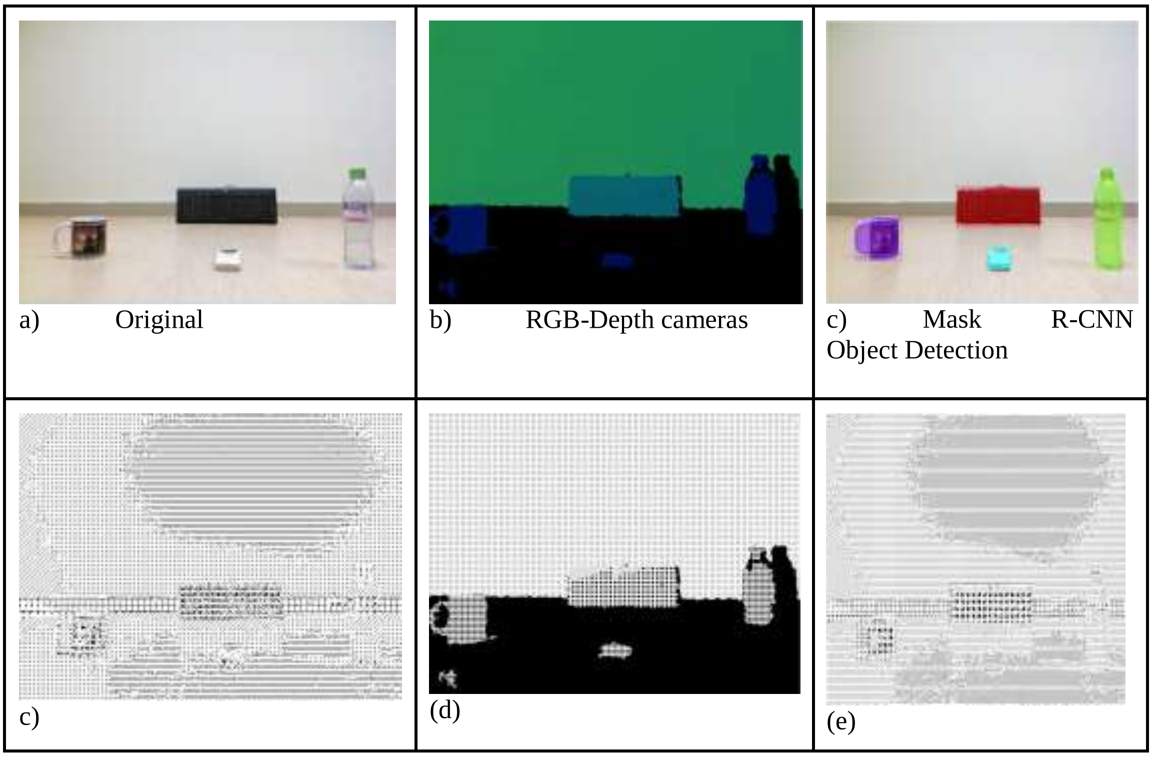


Figure 10–The depth image by scanning of the scene from figure 8 with the Microsoft Kinect depth Sensor.

The three images from Figures 8, 9 and 10 were used as import to the algorithm described in section IV. The output of the algorithm was used as an import to the haptic algorithm described in section V and evaluated with the haptic device Novint Falcon. The produced haptically enhanced pictures are depicted in figure 11.



a) Original

b) RGB-Depth cameras

c) Mask R-CNN Object Detection

c)

(d)

(e)

Figure 11–The produced haptically enhanced pictures a) Original b) from RGB-Depth cameras and c) from Mask R-CNN.

The test users were asked to identify the position and the occupied area of the 4 objects of the scene with their eyes closed. After the test the users were asked three questions regarding the usability metrics of effectiveness, efficiency and Satisfaction. The questions were: A) If they were able to complete the task (effectiveness**)**, b) if it was easy for them to complete task (efficiency) and C) How happy were they with this experience (Satisfaction). The scale for the answers was from 1 to 5, where 1 is lowest perceived quality, and 5 is the highest perceived quality, as depicted in Table 1.

Table 1–Scaling for User Experience for effectiveness, efficiency and Satisfaction.

Rating	Label
5	Excellent
4	Good
3	Fair
2	Poor
1	Bad

The Mean Opinion Score from the User Experience evaluation are depicted in Table 2.

Table 2–User Experience Score from the two proposed methods

User Experience	Original image	RGB-Depth Rendering	Mask R-CNN Rendering
Effectiveness	2.8	4.2	2.9
Efficiency	2.6	4.1	2.6
Satisfaction	2.5	4.0	2.5

From table 2 it is understood that the RGB-Depth method is by far more effective, efficient and satisfying that the other two methods. This based on the fact that the RGB –Depth method renders the scene, detects the physical objects and removes all the unnecessary details and colors from the image. This helps the user not to be confused with unnecessary details as he explores the tangible image with a haptic device.

VII. CONCLUSION

This paper proposed two methods for enhancing images with haptic features. The first method used a RGB-Depth camera to reconstruct a scene to 3D model. The second method used the Mask R-CNN Artificial Intelligent algorithm to detect the objects in the original image. After the initial processing, both images were inserted to an image processing algorithm for transforming the scattered objects to geometrical patterns. The

geometrical patters were used by the H3D haptic program to enhance the images with haptic features.
For the performance evaluation the Mean Opinion Score method for the User Experience was used. The tests revealed that the RGB –Depth method is a more promising method, as it lowers the colors of the image, and as a consequence the geometrical patterns of the image, to a minimum number. As a result, the objects of the image are easier distinguished with a haptic device.

VIII. Acknowledgements

We would like to thank Prof. Papadopoulos Konstantinos, from the Department of Educational & Social Policy, University of Macedonia Greece for guiding us and lending us the haptic device, Novint Falcon, in order to perform the User Experience evaluation tests.

REFERENCES

[1]. G. Kokkonis, K. Psannis, M. Roumeliotis, and S. Kontogiannis, «A survey of transport protocols for haptic applications.,» IEEE 16th Panhellenic Conference In Informatics (PCI), pp. 192-197, 2012.

[2]. G. Kokkonis, K. E Psannis, M. Roumeliotis, «Network adaptive flow control algorithm for haptic data over the internet–NAFCAH,» International Conference on Genetic and Evolutionary Computing, pp. 93-102, 2015.

[3]. G. Kokkonis, K. Psannis, M. Roumeliotis, S. Kontogiannis, Y. Ishibashi, «Evaluating Transport and Application Layer Protocols for Haptic Applications» HAVE 2012 –11th IEEE International Symposium on Haptic Audio Visual Environments and Games, Oct 2012.

[4]. M. Zollhöfer, P. Stotko, A. Görlitz, C. Theobalt, M. Nießner, R. Klein, & A. Kolb. «State of the Art on 3D Reconstruction with RGB- D Cameras,» Computer Graphics Forum, 37(2), pp. 625-652, May 2018.

[5]. Z. Zhang, «Microsoft kinect sensor and its effect,» IEEE multimedia,, 19(2), pp. 4-10, 2012.

[6]. Newcombe, R. A., Izadi, S., Hilliges, O., Molyneaux, D., Kim, D., Davison, A. J., ... & Fitzgibbon, A. «KinectFusion: real-time dense surface mapping and tracking» In Proc. IEEE Int. Symp. Mixed and Augmented Reality (ISMAR), pp. 127-136, 2011.

[7]. MILA group, «Theano Deep Learning Tutorial,» 2008.

[8]. Google Research Team, «TensorFlow: A system for large-scale machine learning,» In 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16), 2016.

[9]. Keras library, Available: https://keras.io/.

[10]. R. R. Uijlings, K. E. A. van de Sande, T. Gevers and A. W. M. Smeulders, «Selective Search for Object Recognition,» International Journal of Computer Vision, Springer, 104(2), pp. 154-171, 2013.

[11]. R. Girchick, J. Donahue, T. Darrell and J. Malik, «Rich Feature hierarchies for acurate object detection and semantic segmentation,» in IEEE converence on Computer Vision and Pattern Recognition (CVPR), pp. 580-587, 2014.

[12]. S. Ren, K. He, R. Girshick and J. Sun , «Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks,» IEEE Transactions on Pattern Analysis and Machine Intelligence, 39(1), pp. 1137-1149, 2017.

[13]. K. He, G. Gkioxari, P. Dollár and R. Girshick, «Mask R-CNN,» in IEEE International conference on Computer Vision (ICCV), 2017.

[14]. Facebook Research, 2017 Available: https://github.com/facebookresearch/DensePose.

[15]. R. Alp Guler, N. Neverova and I. Kokkinos, «DensePose: Dense Human Pose Estimation In The Wild,» The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018.

[16]. W. Abdulla, 2017. Available: https://github.com/matterport/Mask_RCNN.

[17]. G. Kokkonis, K. Psannis, C. Asiminidis, S. Kontogiannis, «Design Tactile Interfaces with Enhanced Depth Images With Patterns and Textures for Visually Impaired People,» International Journal of Trend in Scientific Research and Development, 3(1), Dec 2018.

[18]. G. Kokkonis, «Designing Haptic Interfaces with Depth Cameras and H3D Depth Mapping,» Journal of Scientific and Engineering Research, 5(12), pp. 140-146, Dec 2018.

[19]. N. Karbasizadeh, A. Aflakiyan, M. Zarei, M. T. Masouleh and A. Kalhor., « Dynamic identification of the Novint Falcon Haptic device.,» 4th International Conference on Robotics and Mechatronics (ICROM), Tehran., pp. 518-523, 2016.

[20]. Tractinsky, M. Hassenzahl & N. « User experience-a research agenda,» Behaviour & information technology, 25(20, pp. 91-97, 2006.

[21]. Matterport Inc, Accessed Feb 2019. Available: https://github.com/matterport/Mask_RCNN/releases.

George Kokkonis" Using RGB-Depth Cameras and AI Object Recognition for Enhancing Images with Haptic Features" International Journal Of Engineering Science Invention (Ijesi), Vol. 08, No. 03, 2019, Pp 65-72