VOL. 15, NO. 9, MAY 2020 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

DETECTION AND GRIPPING OF AN OCCLUDED TOOL USING DAG-CNN

Paula Catalina Useche M. and Robinson Jiménez-Moreno
Faculty of Engineering, Nueva Granada Military University, Bogotá D. C., Colombia
E-Mail: u3900235@unimilitar.edu.co

ABSTRACT

The techniques of gripping elements by means of manipulative robots have had a wide advance during the last years, allowing them to perform complex tasks such as the follow-up of paths with evasion of both static and dynamic obstacles, in order to reach an objective, hold it and deliver it to the user. However, these algorithms do not allow to directly grasp the elements of interest when they present occlusions, which leads to the need to develop a new sequence of displacement that allows recognizing and eliminating possible occlusions on the desired object, before performing the grip. and delivery thereof. The development of the research work was carried out through the use of Convolutional Neural Networks (CNN) type DAG-CNN (Directed Acyclic Graph CNN), an anthropomorphic robot and VRML simulation, where an occlusion elimination sequence was programmed that allows remove unwanted elements located on the object of interest, before making its grip and delivery to the user, both in a physical and virtual environment. The program achieved 100% success in holding and delivering desired objects with less than 5 occlusion elements, with 99% accuracy in the DAG-CNN for the classification of the desired element with and without occlusions.

Keywords: VRML simulation, DAG-CNN, occluded objects, robot manipulator, object classification.

1. INTRODUCTION

Within the field of robotics, a significant advance has been generated in techniques of grip and manipulation of elements, with the realization of various works focused on a wide range of methods and processes of subjection, such as the use of the haptic sensor for identification and extraction of characteristics of tactile objects, acquired through a single grip equipped with manometric pressure sensors, as developed in [1], or as the work done in [2], where the grip process was focused on the detection of the most stable clamping point, based on the search for the point closest to the geometric center of the object, and with a high percentage of contact between the clamp and the element.

On the other hand, cases such as the one presented in [3], have focused on the manipulation of everyday elements, such as kitchen instruments, using real-time stereo vision and a 5-finger gripper, while in [4] He focused the process of gripping moving elements, using 3D information and a manipulator with clamp, to expand the field of robotics towards dynamic environments. In other cases such as the one shown in [5], a computational algorithm was generated for the grip of 3D objects with autonomous robotic hands of multiple fingers, and in [6] a comparison is made between human and robotic grip, in order of understanding the differences in both processes, correlating them, and obtaining a manageable math for human grip processes.

The grip problems have also been solved by artificial intelligence techniques, such as those performed in [7] and [8], where convolutional neural networks (CNN) were used for the detection and selection of grip points on objects of various geometries, without the use of detection tables or region proposal techniques, through a CNN regression.

On the other hand, in works such as the one developed in [9], a CNN was used to predict the probability of success of a gripping process, using images from a monocular camera, where the network evaluates the spatial relationship between the clamp and objects of the scene to achieve eye-hand coordination, or as the one performed in [10], where a CNN was implemented to incorporate exteroception and proprioception in the evaluation of compression stability in the grip process.

CNNs are neural networks that train convolutional filters for the extraction of characteristics in images, in order to classify them into categories, or to recognize patterns, as explained in [11] [12]. However, networks with DAG structure (Directed Acyclic Graph) were developed that consist of more than one line of CNN layers, which converge in a Fully Connected to subsequently perform the classification, as explained in [13]. These types of structures (known as DAG-CNN) allow greater extraction of features because each line of layers can focus on general or specific aspects of the image, depending on the size of the filters defined by the user.

An example of the use of these networks is shown in [14] where a diagnosis of an inflammatory bowel disease (IBD) was made through a histological examination of tissue imaging, in which the multilayer structure of the DAG-CNN is used for the detection and extraction of characteristics of various sizes and shapes in images of muscular and disordered regions, which is of vital importance for the diagnosis of said disease.

As it can be observed, in none of the works developed for the grip and manipulation of elements, were considered any situation with the presence of some type of occlusion on the object of interest, but the approach was directed to the process of grip and monitoring of the desired element, in real time, considering that it is free of obstructions.

These situations limit the working conditions of the algorithms, causing failures in their execution when

the desired element is not free, so, in the following article, it was decided to perform a process of gripping occluded elements using a 3DOF manipulator and a monocular vision system with classification of objects by DAG-CNN, where the obstructions are on the element of interest, preventing a direct grip of it, and the user provides general information of the work environment through an interface, to facilitate the removal of occlusions, prior to the object's grip wanted.

The present article is divided into 6 main sections: the first is the introduction, where the current state of object grip and manipulation algorithms is presented, and their relationship with artificial intelligence techniques such as CNN and DAG-CNN. In the second section, the basic operation of the program and the use of the user interface are presented. In the third section, the process of detecting and classifying occlusions on the element of interest is explained, while in the fourth section the process of eliminating occlusions is presented, from the capture of the work environment to the grip of the desired element. In the fifth section, the results and analyzes obtained during the physical tests of the algorithm are presented and, finally, in the sixth section, a series of conclusions related to the program, its operation and its field of action are proposed.

2. OCCLUSION ELIMINATION ALGORITHM

The program developed is responsible for detecting and classifying the elements present in a work environment through a DAG-CNN that allows to determine the current state of the desired tool (with or without occlusions), and thereby define a sequence of grip that eliminates the occlusions, one by one, before performing the final grip.

For the following program, objects that can be found in a toolbox, such as a scalpel, screwdriver, spanner, scissor and pliers, were established as work tools, where the scalpel was selected as the element of interest, because its geometry is more symmetrical than the other objects, facilitating their classification, and that regardless of which section of the tool is occluded, it is still possible to identify it.

A user interface was developed that allows to observe the image captured by the camera (Camera), the results of the detection and classification (Scalpel Detected), the simulation of the environment in VRML (Simulation VRML), and allows to enter information about the environment, such as the number of occlusions expected (Number of Occlusions) and the thickness of each one (Thickness of Occlusions), as shown in Figure-1.

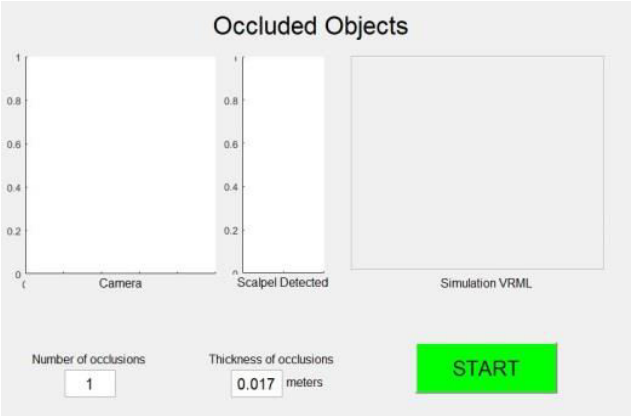


Figure-1. Graphical User Interface (GUI).

Once the program is executed, a window opens that allows you to select the camera to be used in the application, as shown in Figure-2, where you can select between the computer camera or one, or several, external cameras.

Figure-2. Camera selection for the application.

The user must select the camera, wait for the GUI to open, and in it specify the number of occlusions and their thickness, then press START and wait for the variables to initialize. The program takes a picture of the work environment, without any tools present, and instructs the user to wait while the image is being processed, as shown in Figure-3.

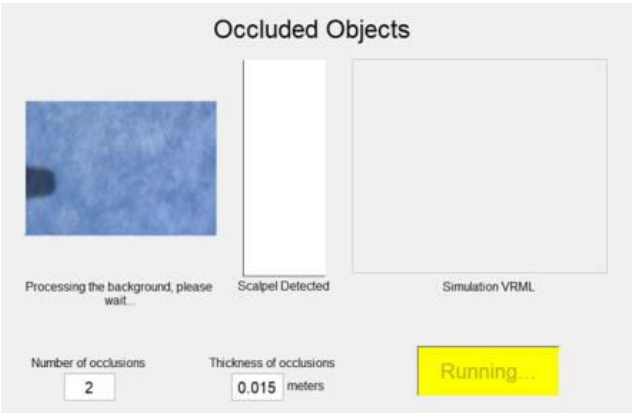


Figure-3. Background processing.

Subsequently, the program loads the VRML and instructs the user to prepare the tools to be identified, as

shown in Figure-4, then the progressive account is started to make the second capture of the work environment, with the tools located in a random position, with or without the presence of occlusions, and the image is captured by completing 15 seconds of waiting.

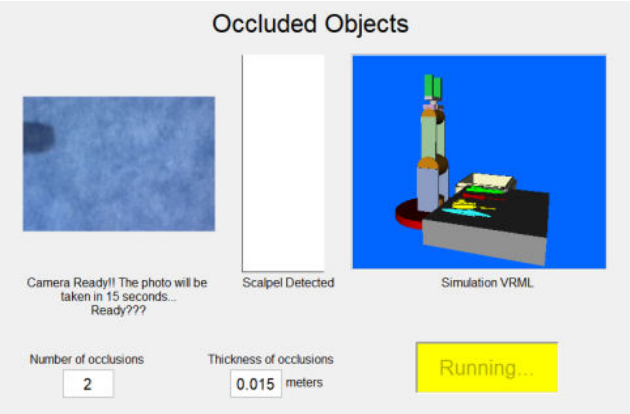


Figure-4. Indication to prepare tools.

Internally, the detection and classification process is applied, whose final results are shown in Figure-5 where, for the example shown, the scalpel is classified as occluded with 100.0% accuracy, and is enclosed in a detection box to indicate its location within the physical environment. Then, the positions of the tools in the virtual environment are simulated, as shown in Figure-6 and the program for the removal of occlusions is prepared.

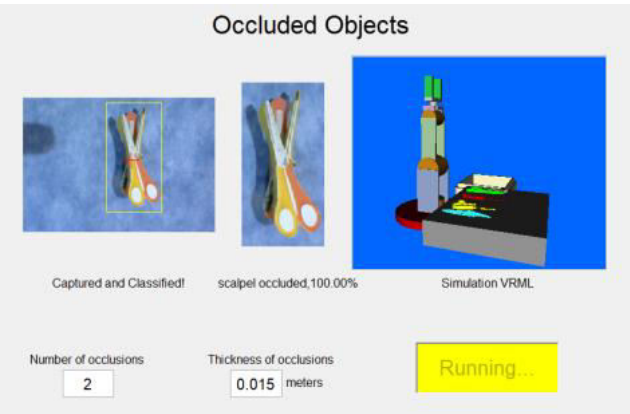


Figure-5. Scalpel detection and classification.

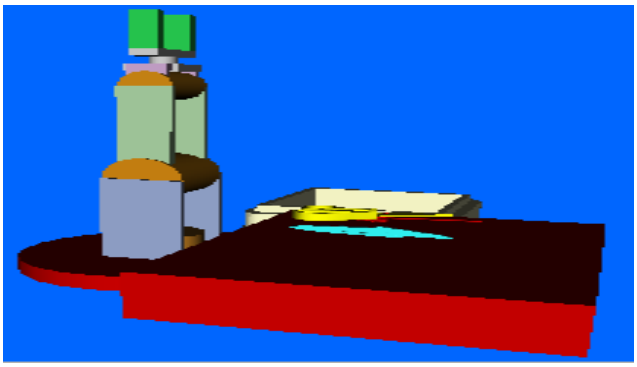


Figure-6. Work environment simulation.

The program proceeds to eliminate each of the occlusions by moving the manipulator, both physical and virtual, to the position of the upper occlusion. There, he takes the tool and deposits it to a box, as shown in Figure-7, then the scalpel is re-classified to verify if there are still occlusions. The process is repeated until all occlusions are removed and it is possible to grab the scalpel, as shown in Figure-8, or until the scalpel is detected earlier than expected and a direct grip is made.

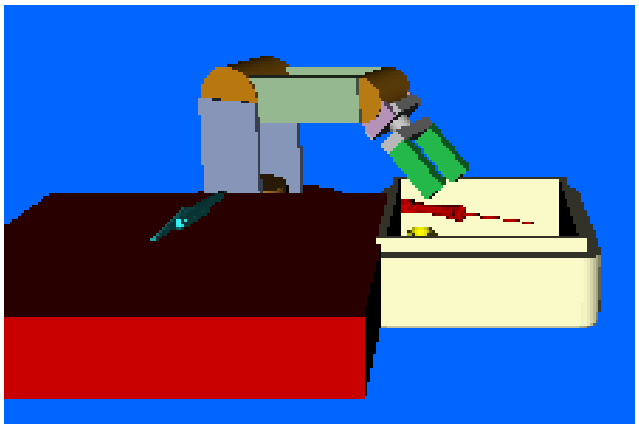


Figure-7. Occlusion elimination.

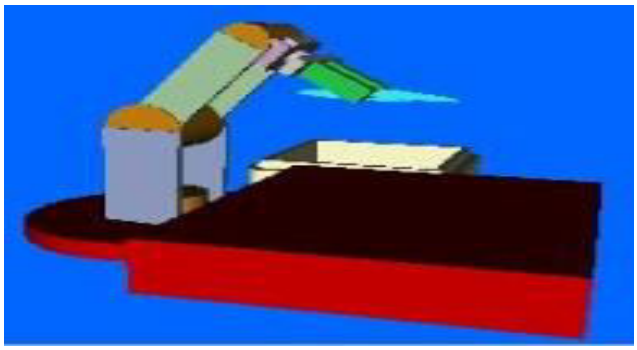


Figure-8. Final scalpel grip.

Once the program is finished, the interface indicates its completion, as shown in Figure-9, where the scalpel located in the center of the table is shown (see VRML simulation), and its detection as “occlusion free”.

The operation of the algorithm is shown, in summary form, in the flow chart of Figure-10, where the process begins with the selection of the camera, and ends with the final delivery of the scalpel in the center of the table.

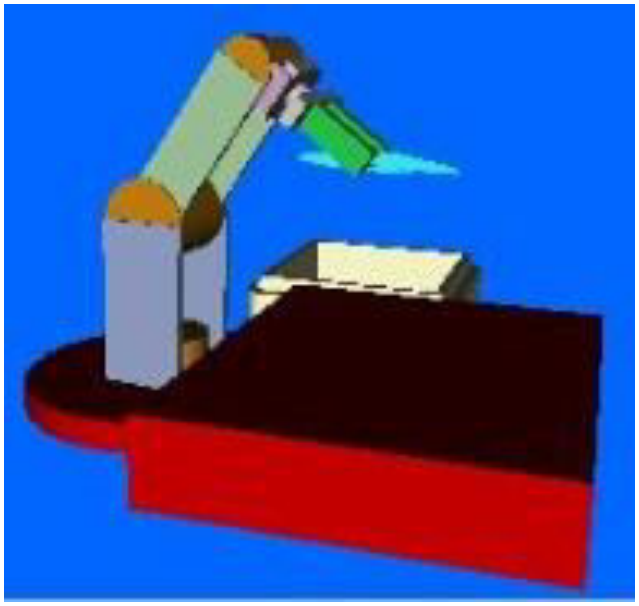


Figure-9. Program completion.

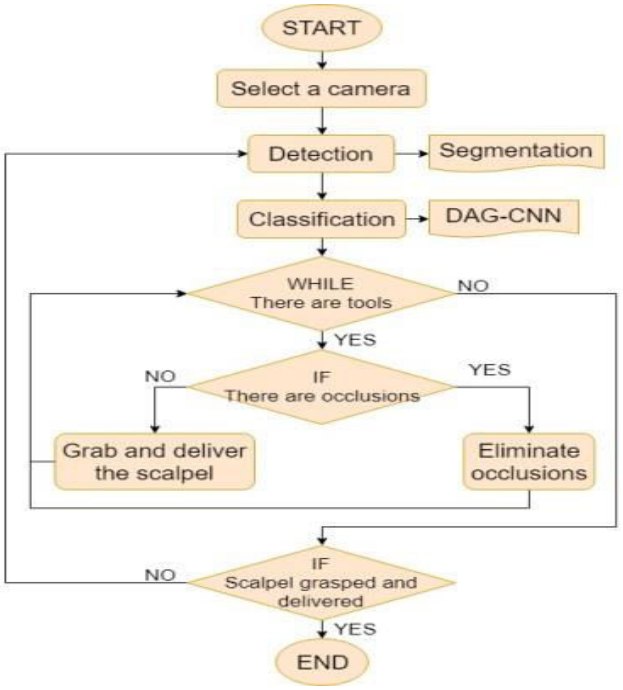


Figure-10. Algorithm flow chart.

In the first step, the process of detection and classification of the desired tool is carried out, where all objects are extracted from the work environment, covered with a detection box and classified with the DAG- CNN, to determine if the scalpel is occluded or not. In the second step, the manipulator is moved to each of the occlusions and removed them, one by one, until free the scalpel, which is grabbed and left in the center of the table. The program ends only when all occlusions have been removed and the scalpel has been ordered.

3. DETECTION AND CLASSIFICATION OF OCCLUSIONS

For the process of detecting objects in the environment, a comparison was made between two images: the background captured by the camera at the beginning of the program (see Figure-4), and the photo of the work area with the elements located, made at finish the 15 second count (see Figure-5).

From both images, a subtraction was made between the average color of the background, and the average of the image with the tools, so that the presence of any element on the table represents a change of hue in specific areas of the image, and makes the difference between them greater than a set threshold.

On each area where the difference exceeds the threshold (defined according to the sensitivity of the camera to lighting changes), a detection box is generated that completely encloses it, and allows only that portion of the entire image to be extracted, to process it later with the DAG-CNN.

Since the dimensions of the detection box depend on the area occupied by the element detected in the image, it is necessary to adjust the size of said box to the height/width ratios of the input image of the DAG-CNN, in order to ensure that, when resizing the image to 80x40 pixels (network input), no deformations are generated in the tools. To achieve this, the height of the detection box was kept constant, and the width was changed to one that would ensure the desired proportion, using (1) and (2), where the first calculates the expected width for the existing height (Adjust), and with the second one, the number of pixels required on each side of the detection box is calculated to achieve the desired width (pxW).

$$Adjust = High/ratio \quad (1)$$

$$pxW = (Adjust - Width)/2 \quad (2)$$

On each side of the detection frame, the result of equation (2) was added, and that image was captured as the input of the DAG-CNN, resizing it to 80x40 pixels. Subsequently, for the classification process, a DAG-CNN was trained for the detection of occlusions on the scalpel, which generates the “occluded” or “not occluded” states as output, depending on the elements that are next to or on the scalpel.

The database was built with 5 types of scalpels, of different shades and dimensions, which were accompanied by other tools such as scissors, screwdrivers, spanners and pliers, which were used as occlusions. For each case, aspects such as light intensity, the presence of shadows, and changes in position and orientation were varied, and then the database was increased, using the Data Augmentation algorithm developed in [15], to obtain a total of 3000 training images and 1000 test, by category.
An example of the images that make up each category is presented in Figure-11, showing the 5 different types of scalpels used, and some of the tools that generate occlusions.

Figure-11. Database a) scalpel, b) scalpel occluded.

The network was trained for 30 epochs, reaching 99% accuracy, with an input image of 80x40 pixels and an architecture like the one shown in Table-1, where the layers used in each of the branches are specified. The entry of each branch is the image to be classified, and its junction point is "addition", which is followed by a Fully Connected layer, a Dropout and a Softmax.

Where CONV corresponds to a Convolution layer, RELU to a Rectified Linear Units, BATCH to a Batch Normalization, MAXPOOL to a Maxpooling, FC to a Fully Connected, DROP to a Dropout, and SOFT to a Softmax (explained in [11]).

The dimensions of the output image of each convolution layer were calculated using equations (3) through (5), where W and H are the width and height of the image, respectively, Fw, Fh and D are the width, height and depth of the filter of convolution, Pw and Ph are the padding for width and height of the image (equal to zero for the trained network), sw and sh are the stride for the width and height of the filter, K is the number of filters, i is the current convolution layer, and i+1 is the output of said layer.

Table-1. Architecture of the branches of the DAG-CNN.

LAYERS BRANCH 1	**Filter Size HxW**	**Stride**	**Number of Filters**
CONV+BATCH+RELU	4x3	1	16
CONV+BATCH+RELU	4x3	1	64
MAXPOOL	2x2	2	--
CONV+BATCH+RELU	4x3	1	128
CONV+BATCH+RELU	4x3	1	256
MAXPOOL	3x2	2	--
CONV+BATCH+RELU	4x3	1	512
FC1+RELU+DROP	--	--	--
FC2+RELU+DROP	--	--	--
Addition	--	--	--
LAYERS BRANCH 2	Filter Size HxW	Stride	Number of Filters
CONV+BATCH+RELU	3x3	1	16
CONV+BATCH+RELU	3x3	1	128
CONV+BATCH+RELU	3x3	1	192
MAXPOOL	2x2	2	--
CONV+BATCH+RELU	3x3	1	256
CONV+BATCH+RELU	3x3	1	512
MAXPOOL	3x3	2	--
FC1+RELU+DROP	--	--	--
FC2+RELU+DROP	--	--	--
Addition	--	--	--

$$W_{i+1}=((W_i-Fw_i+2Pw_i)/sw_i) + 1 \quad (3)$$

$$H_{i+1}=((H_i-Fh_i+2Ph_i)/sh_i) + 1 \quad (4)$$

$$D_{i+1}=K_i \quad (5)$$

In Figure-12 the confusion matrix is shown, where it is possible to observe an error of about 0.5% in the classification of each category, with failures of less than 15 images per 1000 test. Due to the high percentage of accuracy, the network was used to detect scalpel occlusion.

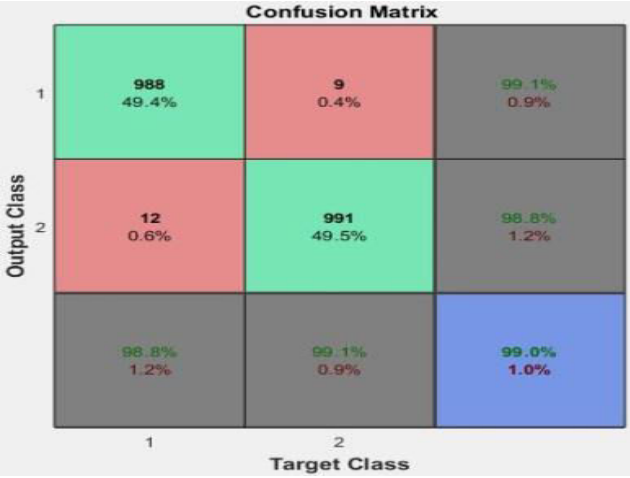


Figure-12. Confusion matrix.

4. PROCESS OF ELIMINATION OF OCCLUSIONS

Next, the occlusion elimination process developed for the application is explained, which is divided into 3 parts: definition of grip points, current state of the tool, and displacement of the manipulator, which are presented in the following subsections.

A. Definition of grip points

The first step in the process of removing occlusions is to obtain, store and organize the positions of all objects detected in the workspace.

For this, the coordinates of the center of the detection box, previously generated, were used as the grip points (X, Y) for all the elements, and the height (Z) of each one was calculated from the thickness of the occlusions and the number of user-defined objects, as shown in (6), where "vari" is an arrangement from 1 to the number of objects (n), and "esp" is the thickness

$$Pz=(1:n)= vari(1:n)*esp \quad (6)$$

All coordinates were organized in a single arrangement (Pg) of dimension nx3. The arrangement contains, in each row, the X, Y coordinates of the detection box, and one of the points of Pz, where the last row saves the height of the scalpel (minimum Pz), and the first saves the height of the highest occlusion (maximum Pz).

After obtaining each of the grip points, the end points (PF) of each sequence were established, where the box is the end point of the occlusions, and the center of the table is the end point of the desired object.

B. Status of the tool

In the second step, the DAG-CNN is used to verify the existence of occlusions on the scalpel, and to confirm that the object detected at the end of the tool stack is a scalpel.

Before the process of gripping and manipulating elements, the detection box was evaluated using the DAG-CNN, and only if the object present in the image is considered an occlusion, it is removed, otherwise, where an occlusion is detected as a scalpel, the robot proceeds to take it, deliver it and terminate the program, regardless of whether the number of occlusions set by the user is greater than the number of elements handled.

When the scalpel appears earlier than expected, the program shows the message in Figure-13, indicating the lack of occlusion and the decision to make a direct grip on the tool.

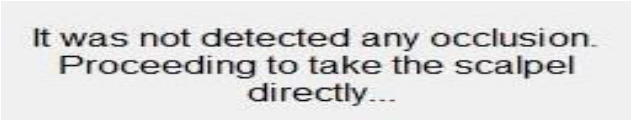


Figure-13. Scalpel found earlier than expected.

C. Displacement of the manipulator

The next step consists of use the coordinates of the matrix Pg and PF to generate the displacement of the manipulator.

For the application, an anthropomorphic robot with 3DOF (Degrees of Freedom) and two-finger gripper was used, whose dimensions and direct and inverse kinematics are developed in [16].

For the displacement of the final effector, a straight line path was established between the start and end point of each section of the route, where the linear displacement speed, with respect to each of the axes, presents a parabolic behavior as shown in Figure-14, where the maximum speed is reached in the middle of the route, and the minimum at the beginning and at the end.

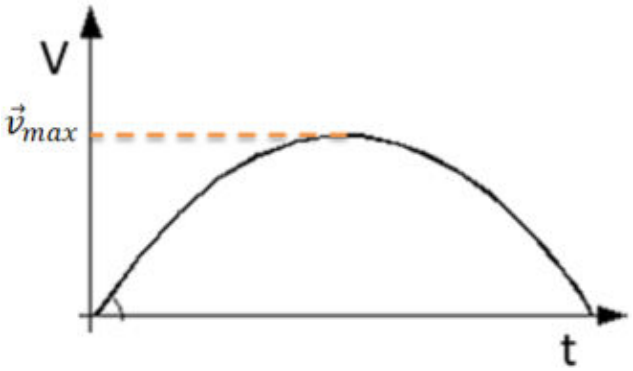


Figure-14. Final effector speed.

The trajectory was divided into a number of intermediate points where, to ensure the previously established speed, the distance between them was defined by a cubic behavior, such as the one shown in Figure-15,

where the position changes are small at the beginning and at the end of the trajectory, and big in the middle.

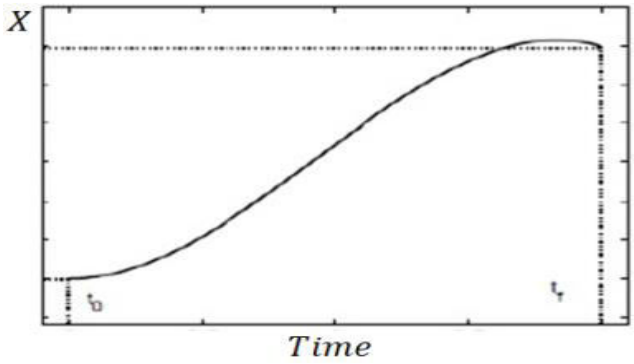


Figure-15. Final effector displacement.

To calculate each of the points of the trajectory in the three coordinate axes according to the desired displacement time and the established cubic behavior, (7) was used, where x(t) is the position function for any of the coordinate axes (X, Y, Z), a_1,a_2,a_3,a_4 are constants and t is time. For the calculation of each of the constants of (7), the velocity function of (8) was obtained, deriving (7) with respect to time, and it was established that the initial and final velocity should be equal to zero, as shown in Figure-14.

$$X(t) = a_1t^3 + a_2t^2 + a_3t + a_4 \quad (7)$$

$$v(t) = a_1t^2 + a_2t + a_3 \quad (8)$$

Then, the position and velocity functions were evaluated for a time t=0 and t=t_f, where t_f is the total displacement time, to find the constants a_1, a_2, a_3, a_4.

To execute the sequence of occlusion elimination, multiple initial and final points of displacement were established, and between each of them straight path trajectories were defined by recalculating the constants of (7) and (8) for each case. The points were defined in such a way that the manipulator went down and up in a straight line, at the time of taking or leaving an element, and then moved to the next point of the sequence, and so on, until finishing with the final grip of the scalpel and its delivery at the table.

5. RESULTS AND DISCUSSIONS

In Figure-16 the simulated occlusion removal process in VRML is shown for the case of 2 existing obstructions on the scalpel, where the sequence is numbered from 1 to 9, and in each image, is shown a different process step.

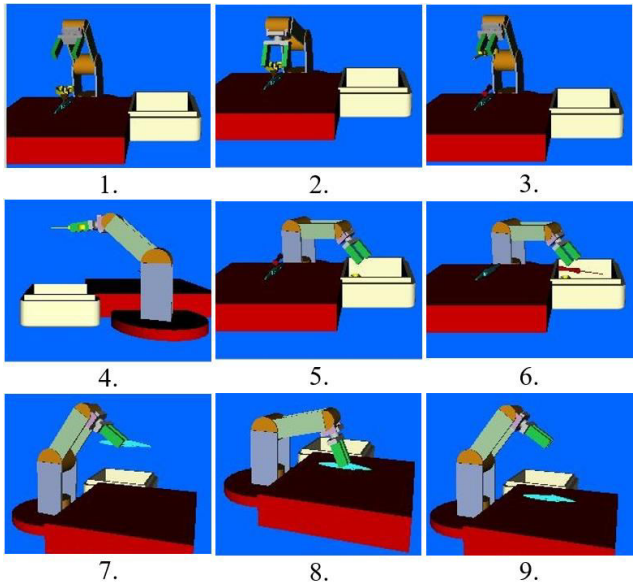


Figure-16. Simulation of occlusion removal.

In the first 5 steps, each of the intermediate points of the robot's displacement is shown during the elimination of one of the occlusions, in step 6 the simulation of the fall of a tool in the box is presented, and in steps 7 to 9 shows the process of taking and delivering the scalpel.

On the other hand, in Figure-17 the process of removing occlusions in a physical environment is presented, in the case of 3 obstructions, where it is possible to appreciate the grip points made by the manipulator, and the final delivery of the scalpel.

In the first two steps, the workspace and the initial location of the tools are shown in Figure-17, in steps 3 to 6 the process of eliminating the first occlusion is detailed, from steps 7 to 10 the elimination of the following occlusions is presented, and in steps 11 and 12 the grip and delivery of the scalpel is shown.

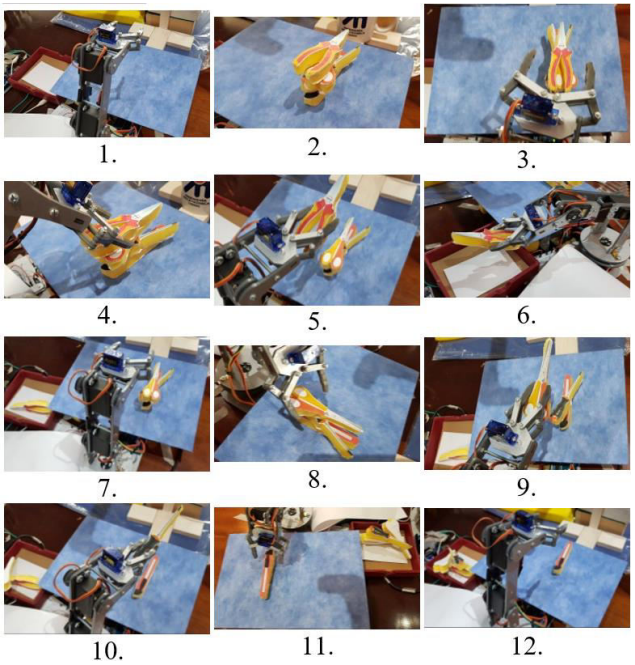
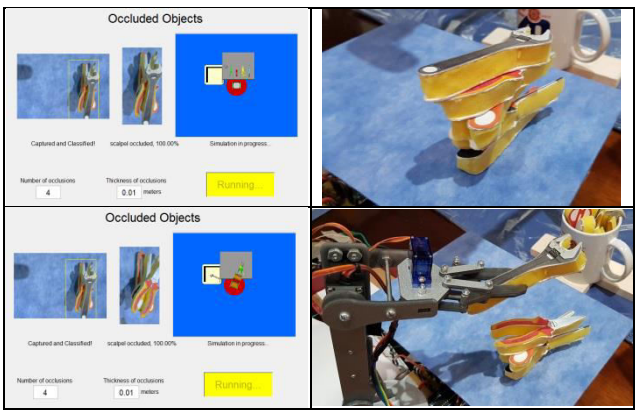


Figure-17. Elimination of occlusions in a physical environment.

For a test case with 4 occlusions, each of them was removed without altering the position and orientation of the elements below the upper occlusions, nor making false grips neither more than one tool grip when closing the gripper, such as it is shown in Figure-18, where the updated GUI for each case and the grip of each element are presented.

As can be seen in the examples presented, one of the requirements to ensure the proper functioning of the algorithm is that the geometric center of the tools be as close as possible to the X, Y coordinates of the detection box, to ensure a grip without shocks.

Additionally, the proposed process requires that the occlusions be on top of each other, completely horizontal, in order to ensure that the calculated height coincides with the real one, so that the algorithm can fail in situations with inclined elements, as shown in the example of Figure-19, where, additionally, the position of the tools does not match with the detection box.



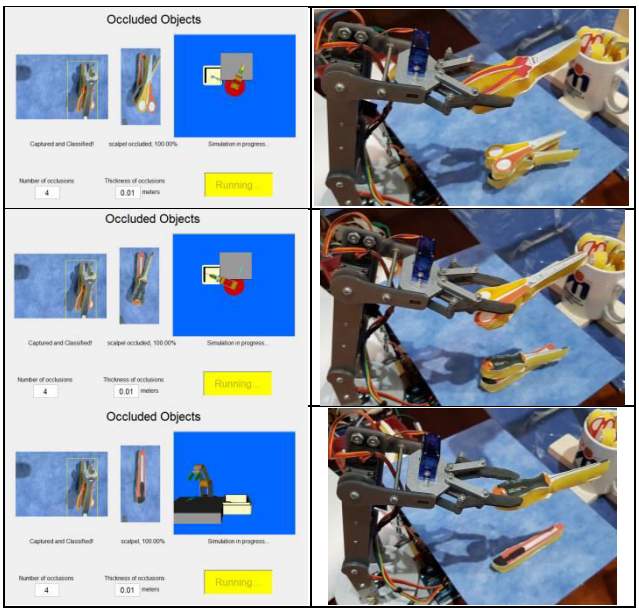


Figure-18. Elimination of 4 occlusions.

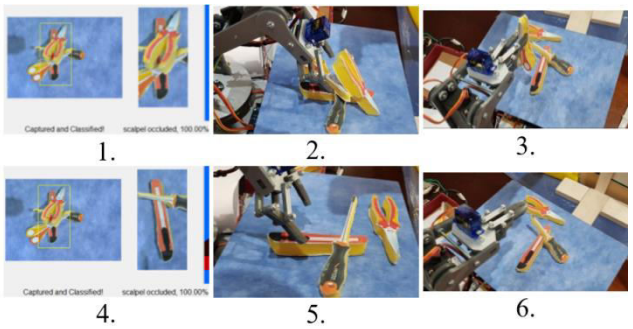


Figure-19. Elimination of 4 occlusions.

As you can see, in steps 2 and 3, the robot moves towards the coordinates of the detection box, where only one of the 3 occlusions is found, and when grabbed, it pushes the others, leaving them as shown in step 3 Subsequently, when the robot searches for the next occlusion, it makes a false grip, since no objects are found in that position, as shown in steps 5 and 6. In steps 1 and 4 the image captured by the camera and its classification are presented.

Next, an evaluation of the behavior of the algorithm against a different number of occlusions (No. Oc.) between 1 and 6, is presented, where the number of tools successfully grasped is determined for 3 tests per case (TRY), as shown in Table-2, which establishes the number of successes with respect to the total number of occlusions (successes / occlusions), the percentage of successes (%), and the general average of successes (Mean).

As shown, the algorithm works without failures in a range of between 1 and 4 occlusions, considering that the thickness of each of the tools is approximately 1cm, the maximum width of 3.5cm, and the maximum length of 10cm. In the tests carried out, the geometry of the tools caused the stack of objects to become unstable by

increasing the number of occlusions above 4 objects, increasing the chances of collapse when removing an element, as happened in the third try of 6 occlusions, where the grasping of the first object caused the fall of others.

Table-2. Detection of program failures against a different number of occlusions.

Nº Oc.	TRY			Mean
	1	2	3	
1	1/1 100%	1/1 100%	1/1 100%	100%
2	2/2 100%	2/2 100%	2/2 100%	100%
3	3/3 100%	3/3 100%	3/3 100%	100%
4	4/4 100%	4/4 100%	4/4 100%	100%
5	5/5 100%	3/5 60%	5/5 100%	87%
6	4/6 67%	5/6 83%	1/6 17%	56%

Additionally, it was observed that when closing the gripper in the upper occlusions, a movement and rotation effect is generated in the rest of the elements, due to the contact between them, which causes the position of the lower objects to differ with respect to the initial one, until it no longer matches with the established grip coordinates, causing failures of grasping, as happened in the first two tests for 6 occlusions, and in the second test with 5 occlusions.

6. CONCLUSIONS

The occlusion elimination program proposed in the article demonstrates that it is possible to use a DAG-CNN to detect occlusions in the desired object, and from there execute a sequence of grip and elimination of elements that generate the obstruction.

The algorithm is able to eliminate up to 4 occlusions on the desired object, without affecting the location of the other elements, or pushing them. A greater number of elements, causes instability in the stack of objects and makes it more sensitive to any movement generated by the closing of the gripper on any tool. To solve it, it is proposed to use occlusions with smaller thickness and upper width, or use a gripper that allows to adjust the degree of inclination of the grip with respect to the orientation of the object, in order to reduce the movement of the tool during the grasping.

The program allows the manipulator to take decisions about the grip of elements according to the existence, or not, of occlusions, using the information extracted by the image, and granted by the user. However, to reduce the influence of the user on the program and increase the working conditions of the algorithm, it is necessary to use an additional method to recognize the number of occlusions and their independent positions in the environment, in order to avoid grip failures such as those presented previously, for what it is proposed to use, in future works, three-dimensional information of the work environment.

Additionally, it is proposed to detect the location of each element in the workspace prior to each grip of the manipulator, in order to update its position, in case the grasping of a previous element caused changes in the rest of the objects, as in the example in Figure-19.

ACKNOWLEDGMENTS

The authors are grateful to the Nueva Granada Military University, which, through its Vice chancellor for research, finances the present project with code IMP-ING-2290 and titled "Prototype of robot assistance for surgery", from which the present work is derived.

REFERENCES

[1] Spiers A. J., Liarokapis M. V., Calli B. & Dollar A. M. 2016. Single-grasp object classification and feature extraction with simple robot hands and tactile sensors. IEEE transactions on haptics. 9(2): 207-220.

[2] Useche Paula, Jimenez Robinson, Hernandez Rubén. Algorithm for Tool Grasp Detection. 12(1).

[3] Morales A., Asfour T., Azad P., Knoop S. & Dillmann R. 2006, October. Integrated grasp planning and visual object localization for a humanoid robot with five-fingered hands. In Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on (pp. 5663-5668). IEEE.

[4] Allen P. K., Timcenko A., Yoshimi B. & Michelman P. 1993. Automated tracking and grasping of a moving object with a robotic hand-eye system. IEEE

Transactions on Robotics and Automation. 9(2): 152-165.

[5] Sahbani Anis; El-Khoury Sahar; Bidaud Philippe. 2012. An overview of 3D object grasp synthesis algorithms. Robotics and Autonomous Systems. 60(3): 326-336.

[6] Cutkosky Mark R.; Howe Robert D. 1990. Human grasp choice and robotic grasp analysis. En Dextrous robot hands. Springer, New York, NY. pp. 5-31.

[7] Redmon Joseph; Angelova Anelia. 2015. Real-time grasp detection using convolutional neural networks. En Robotics and Automation (ICRA), 2015 IEEE International Conference on. IEEE. pp. 1316-1322.

[8] Guo D., Kong T., Sun F. & Liu H. 2016, May. Object discovery and grasp detection with a shared convolutional neural network. In Robotics and Automation (ICRA), 2016 IEEE International Conference on (pp. 2038-2043). IEEE.

[9] Levine S., Pastor P., Krizhevsky A. & Quillen D. 2016, October. Learning hand-eye coordination for robotic grasping with large-scale data collection. In International Symposium on Experimental Robotics (pp. 173-184). Springer, Cham.

[10] Kwiatkowski Jennifer; Cockburn Deen; Duchaine Vincent. 2017. Grasp stability assessment through the fusion of proprioception and tactile signals using convolutional neural networks. En Intelligent Robots and Systems (IROS), 2017 IEEE/RSJ International Conference on. IEEE. pp. 286-292.

[11] Krizhevsky Alex; Sutskever Ilya; Hinton Geoffrey E. 2012. Imagenet classification with deep convolutional neural networks. En Advances in neural information processing systems. pp. 1097-1105.

[12] Bichler Olivier. 2017. Convolutional neural network. U.S. Patent Application No 15/505,231, 13 Jul.

[13] Yang Songfan; Ramanan Deva. 2015. Multi-scale recognition with DAG-CNNs. En Computer Vision (ICCV), 2015 IEEE International Conference on. IEEE. pp. 1215-1223.

[14] Wang J., MacKenzie J. D., Ramachandran R. & Chen D. Z. 2016, October. A deep learning approach for semantic segmentation in histology tissue images. In International Conference on Medical Image Computing and Computer-Assisted Intervention (pp. 176-184). Springer, Cham.

[15] P. C. U. Murillo, J. O. P. Arenas and R. J. Moreno. 2017. Implementation of a Data Augmentation Algorithm Validated by Means of the Accuracy of a Convolutional Neural Network. Journal of Engineering and Applied Sciences. 12(20): 5323-5331.

[16] Murillo Paula Useche; Moreno Robinson Jimenez; Mauledeox Mauricio. 2016. Multi User Myographic Characterization for Robotic Arm Manipulation. International Journal of Applied Engineering Research. 11(23): 11299-11304.