



Article Design of Demonstration-Driven Assembling Manipulator

Qianxiao Wei, Canjun Yang *, Wu Fan and Yibing Zhao

State Key Laboratory of Fluid Power and Mechatronic Systems, Zhejiang University, Hangzhou 310027, China; wqx@zju.edu.cn (Q.W.); 3150104930@zju.edu.cn (W.F.); ybzhao@umich.edu (Y.Z.)

* Correspondence: ycj@zju.edu.cn; Tel.: +86-571-87951271 (ext. 6318)

Received: 26 March 2018; Accepted: 10 May 2018; Published: 16 May 2018



Abstract: Currently, a mechanical arm or manipulator needs to be programmed by humans in advance to define its motion trajectory before practical use. However, the programming is tedious and high-cost, which renders such manipulators unable to perform various different tasks easily and quickly. This article focuses on the design of a vision-guided manipulator without explicit human programming. The proposed demonstration-driven system mainly consists of a manipulator, control box, and camera. Instead of programming of the detailed motion trajectory, one only needs to show the system how to perform a given task manually. Based on internal object recognition and motion detection algorithms, the camera can capture the information of the task to be performed and generate the motion trajectories for the manipulator to make it copy the human demonstration. The movement of the joints of the manipulator is given by a trajectory planner in the control box. Experimental results show that the system can imitate humans easily, quickly, and accurately for common tasks such as sorting and assembling objects. Teaching the manipulator how to complete the desired motion can help eliminate the complexity of programming for motion control.

Keywords: automatic control of manipulator; identification of target object; assembly system; manipulator learning; robotic imitation learning

1. Introduction

Assembly is required when a machine is built of many individual parts. In the past few decades, assembly has been manually completed by workers [1]. Thanks to the rapid development of industrial automation, more and more work is now completed by machines [2]. Peg-hole assembly, flat panel parts assembly, and many other automatic assembly machines have been invented for this purpose [3,4].

Though these assembly machines could accomplish the desired task satisfactorily, they are usually complicated, and can only be used in some specific scenarios. In a large-scale automobile assembly system, flexible and adaptable assembly technology and strategies include robotic fixtureless assembly, self-reconfigurable assembly systems, and increased modularity in the assembly process [5]. In flexible manufacturing, it is required that the assembly machine be able to perform various tasks easily and quickly. However, it is expensive to adjust the existing assembly line, since the equipment needs to be replaced to performance different tasks. Besides, different tasks have different requirements for the robot's initial and final postures as well as the placement location of the workpieces. The traditional assembly system not only has difficulties in sorting and assembling various different workpieces, but also may fail to grab the required positions of workpieces that change dynamically in different tasks, and thus has negative influences on production efficiency [6,7].

To further improve the universality of assembly machines, 6-DOF (Degree Of Freedom) robot manipulators are widely used in automated assembly because of their flexibility. However, how to control the manipulators efficiently, accurately, and easily is always a difficult and urgent problem.

Vision-guided manipulators were proposed to tackle this problem [8–10]. To improve the performance of object recognition in the system, Laptev proposed a method of boosted histograms [11]. Influential research by Aivaliotis and Zampetis presented a method for the visual recognition of parts using machine learning algorithms to enable the manipulation of parts [12]. Chunjie Zhang came up with an image representation method using boosted random contextual semantic spaces [13]. Panagiota Tsarouchi proposed an online two-dimensional (2D) vision system combined with offline data from CAD (Computer Aided Design) files for the computation of three-dimensional (3D) POI (Points of Interest) coordinates [14].

In robotic assembly with a vision system, many new technologies are emerging. Corner detection is quite efficient in part identification and quick enough to be used for online applications [15]. Part recognition using vision and ultrasonic sensor, which is installed at the end-effector of the robot, is used to recognize objects for a robotic assembly system [16]. Two cameras are coordinated to capture images, which are used for golf club head production. The system adopts automated mechanical arms matched with cameras for 3D space alignment [17]. For large-scale components, robotic assembly system guided by two vision sensors and three one-dimensional (1D) laser sensors were realized to automatically localize and search the best 3D position and orientation between the object and the installation location [18].

This article focuses on a demonstration-driven industrial manipulator based on the object recognition and vision-guided method. One can demonstrate how to assemble the workpieces manually to the manipulator in advance, and then the manipulator can understand the task based on visual information processing. Teaching the manipulator how to complete tasks by simple demonstration can eliminate the complexity of explicit programming for manipulator control for different tasks.

2. Methods

The proposed system shown in Figure 1 works in two phases: the demonstration phase and the automatic assembly phase.

First, in the demonstration phase, one manually shows to the manipulator each step (e.g., move one workpiece on top of another) of a specific task. The 3D camera obtains the spatial positions of the workpiece dynamically during the demonstration, and then transform the positions based on the coordinate system of the industrial manipulator. According to the inverse kinematic model of the industrial manipulator, the control information of each manipulator joint is obtained for each position of the workpiece. Then, in the auto-assembly phase, the manipulator is driven to complete the task automatically based on the control information generated above.



Figure 1. Route chart of teaching a manipulator by vision guiding, without programming the device.

The proposed automatic assembly system consists of a manipulator, a control box, and a camera, as Figure 2 shows. NVIDIA Jetson TK1 was chosen as the image processing module for this system.

YOLO (You Only Look Once) [19,20] was adopted and implemented for object detection. It is a fast algorithm based on convolutional neural network that can detect the target of interest in the input image accurately. The detection speed is very fast, up to 45 frames per second. The robot control circuit uses STM32F103 embedded structure to control the movements of seven servomotors, six of which are the same joint servomotors, while the seventh is a claw servomotor. The time of execution of the control algorithm on the STM32 microcontroller is about 3 ms. The parameters of the robot dynamics of the six joint servomotors are shown in Table 1.



Figure 2. The structure of the automatic assembly system.

Table 1.	The parameters	of the robot	dynamics	of the six	joint servo	motors.

Parameter	Numerical Value
Туре	X20-8.4-50
Torque	38 Kg∙cm
Speed	$0.2 \text{ s}/60^{\circ}$
Load	130 g
Frequency	333 Hz
Voltage	DC 6.0 V

The theoretical safety factor can be obtained as follows:

$$N = \frac{M^*}{M} = \frac{3.7278 \text{ N} \cdot \text{m}}{2.082 \text{ N} \cdot \text{m}} = 1.79$$
(1)

Here, M^* is the maximum output torque of the X20-8.4-50 steering gear and M is the theoretical total torque calculated from the actual mass and length. Therefore, the safety factor provides sufficient allowance for the uncalculated friction force and non-inertial force of the mechanism.

The visual signal processing hardware (NVIDIA TK1), including a 3D camera, can recognize the objects in real-time and capture their motions. According to the signals given by TK1, the manipulator controller can grasp the object, move it to a specified place (as demonstrated by a human), and release it. A wireless communication module is used to increase the flexibility of the manipulator. The structure of the control box is shown in Figure 3.

The underlying models for robot control are shown in Figure 4 below. The boxes with solid lines next to the green arrows represent the input image which contains target objects. The blue boxes represent the sensors and controller components (including a camera, upper computer, lower computer, and rudder). The boxes with solid lines next to the orange arrows represent the output of each component. The boxes with dotted lines next to the green arrows represent the transmission link.

The control interface of the upper computer is written in C# language. It takes the position and shape of the target object as input. Given the arm length of the manipulator and the initial value of each joint, the rudder angles of each joint are solved by the D-H algorithm and are then sent to the lower machine. The upper computer can initialize and configure the minimum and maximum rotation angles of each joint rudder to ensure that there is no collision with the parts or environment. The upper computer also has several other functions which enables fast adjustment of the configurations of the manipulator. First, the rotation angle of each joint rudder can be adjusted and the initial values

can be set manually. Second, aided by a human's demonstration, the manipulator can move along a specific trajectory. The angle values of the joint rudder are sent to the lower computer through the wireless module and the output of the joint rudder rotation is controlled through the PWM (Pulse Width Modulation), which works at 10 Hz. The angle of each transformation can be set by the upper computer and the velocity range is 1°/s–180°/s. In order to prevent jitter, progressive acceleration and deceleration are used when starting and stopping.



Figure 3. The structure of the control box.



Figure 4. The underlying models for robot control.

3. Implementation

3.1. Kinematics Modeling of Manipulator

The D-H kinematics model (as shown in Figure 5) of a 6-DOF manipulator is solved, and the positive and inverse kinematics equations are obtained, which provides the designed parameters for intelligent control of the manipulator. According to the actual grasping situation, the degree of the

freedom of the manipulator is simplified and the complex mathematical model is transformed into a simple kinematic equation. According to the parameters of the D-H linkage model of the manipulator joint [21,22], the transformation matrix of the adjacent joint is obtained. The pose expression of the end (Equation (3)) is easily obtained according to Equation (2). The degree of the joints is shown in Table 2 according to the D-H coordinate system.

$$T_{i-1}^{i} = R_Z(\theta_i) D_Z(d_i) D_X(a_i) R_X(\alpha_i)$$
⁽²⁾

$$A = \frac{1}{2}T_3^2 T_4^3 T_5^4 T_5^5 T_6^6 T \tag{3}$$



Figure 5. Kinematics modeling of the manipulator.

Joint	eta_i	No.	α_i (°)	A_i (mm)	<i>d_i</i> (mm)	$ heta_i$ (°)
1	Adduction/Abduction $-90 < \beta 1 < 90$	2(1–2)	90	28.57	101.59	90 + β1
2	Flexion/Extension $-90 < \beta 2 < 90$	3(2–3)	0	120.05	0	β2
3	Flexion/Extension $-90 < \beta 3 < 90$	4(3-4)	90	20	122.35	β3
4	Flexion/Extension $-90 < \beta 4 < 90$	5(4–5)	-90	0	0	β4
5	Flexion/Extension $-90 < \beta 5 < 90$	6(5–6)	90	0	0	β5
6	Flexion/Extension $-90 < \beta 6 < 90$	7(6-е)	0	0	20.29	β6

Table 2. Degree of joints in D-H coordinates.

3.2. Automatic Control of the Manipulator

The 3D camera obtains the 2D image and the depth information of the manipulator at each position. The manipulator completes the automatic assembling task according to the teaching process after being instructed and trained. During the teaching process, the workpiece is held by hand to complete the migration from the assembly initial posture to the assembly termination posture, as shown in Figure 6.



Figure 6. Teaching the automatic control process of the manipulator.

Firstly, the coordinate system of the manipulator and the 3D camera is calibrated, and the transition matrix from the coordinate system of the 3D camera to the coordinate system of the manipulator is calculated.

Secondly, the 3D camera takes a picture of the working scene, with the workpiece in the original pose at this moment, and the SIFT (scale-invariant feature transform) [23,24] feature of the scene image is calculated. Then the position of the workpiece is identified by the color of the workpiece, and the SIFT characteristic of the workpiece is obtained by the position. The two feature points A and B are picked up, which are the two points farthest away from each other; the vector \overrightarrow{AB} is taken as the original pose of the workpiece, and point A is taken as original position of the workpiece. The feature point A, the feature point B, and the vector \overrightarrow{AB} are all under the coordinate system of the 3D camera.

Thirdly, the workpiece is adjusted from the original pose to the initial pose of assembly, and the SIFT feature is calculated again in the initial pose of the assembly. Then the SIFT feature is matched with the SIFT feature A and the SIFT feature B of the original pose, and the SIFT features A_1 and B_1 in the initial pose are found. The vector $\overrightarrow{A_1B_1}$ is the initial attitude of the assembly and point A_1 is taken as the initial assembly position of the workpiece. The feature points A_1 , B_1 , and the vector \overrightarrow{AB} are all in the coordinate system of the 3D camera (because of the feature-scale invariance of the SIFT feature, the feature points A_1 and B_1 can still be found from the initial pose of the assembly, even if the workpiece rotates or the light changes from the original pose to the initial pose of assembly).

Fourthly, the workpiece is adjusted from the initial pose to the final pose, and the SIFT feature is calculated again in the final pose of the assembly and matched with the SIFT features A and B of the original pose to find the SIFT features A_2 and B_2 in the initial pose. The vector $\overrightarrow{A_2B_2}$ is the end gesture of assembly, and point A_2 is taken as the end assembly position of the workpiece. The feature points A_2 , B_2 , and the vector \overrightarrow{AB} are all in the coordinate system of the 3D camera.

Lastly, the original attitude AB and position A are transformed into control information of the industrial manipulator gripping the workpiece, which is in the initial position and orientation. The initial attitude A_1B_1 and position of assembly A_1 are transformed into control information of the industrial manipulator when it moves the workpiece to the initial pose of assembly. The end attitude A_2B_2 and position of assembly A_2 are transformed into control information of the industrial manipulator when it moves the workpiece to the end pose of assembly. Each pose contains the workpiece's position coordinates and attitude vector.

3.3. Object Recognition with YOLO

Due to the end-to-end characteristic of deep convolutional neural networks, YOLO directly predicts the locations and the classes of objects in a given input image. The model of YOLO consists of many layers. The first several layers (typically convolutional layers) are used for feature extraction and the last several layers are used for the prediction of the final detection results.

In detail, YOLO divides an input image into $S \times S$ grids. Each grid is responsible for outputting B bounding boxes, each of which has four location information variables (x, y, l, w), where (x, y) means the predicted center coordinates and (l, w) means the length and width, as well as one confidence score represented by Pr(Object) which predicts whether this bounding box has object or not. Additionally, C conditional probabilities represented by Pr(Class | Object) are also generated for each grid to indicate the class label for the object. So, the final output layer should have $S \times S \times (5B + C)$ units [20]. We can calculate the class label for each bounding box as:

$$Pr(Class) \times IOU = Pr(Class | Object) \times Pr(Object) \times IOU$$
 (4)

If the predicted and bounding box of the object and its corresponding ground truth are denoted as B_p and B_g , respectively, which are both rectangles, the IOU (Intersection Over Union) is defined as the ratio of the area of the intersection of B_p and B_g to that of their union.

Notably, the loss function should be designed carefully in the training process. The objective of the loss function is to achieve a good balance among the three terms, namely, the location of objects, the prediction of object existence, and the classification of object class. This function can be expressed as follows in Equation (5) and Table 3.

$$J = E + F + G + H + I \tag{5}$$

Symbol	Math Expression	Meaning		
Е	$\lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^{B} l_{ij}^{obj} (x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2$	Loss of center coordinates of the bounding boxes		
F	$\lambda_{coord} \sum_{\mathrm{i}=0}^{\mathrm{S}^2} \sum_{j=0}^{B} l_{ij}^{obj} \left(\sqrt{\mathrm{w}_i} - \sqrt{\widehat{w}_i} ight)^2 + \left(\sqrt{h_i} - \sqrt{\widehat{h}_i} ight)^2$	Prediction loss of length and width of the bounding boxes		
G	$\sum\limits_{\mathrm{i}=0}^{\mathrm{S}^2}\sum\limits_{j=0}^{B}l_{ij}^{obj}(\mathrm{C}_i-\hat{\mathrm{C}}_i)^2$	Confidence prediction loss of the bounding boxes with object		
Н	$\lambda_{ ext{noobj}} \sum_{i=0}^{\mathbf{S}^2} \sum_{j=0}^{B} l_{ij}^{ ext{noobj}} (\mathbf{C}_i - \hat{\mathbf{C}}_i)^2$	Confidence prediction loss of bounding boxes without object		
Ι	$\sum\limits_{ ext{i}=0}^{ ext{S}^2} l_{ij}^{obj} \sum\limits_{ ext{C} \in classes} (p_i(c) - \hat{p}_i(c))^2$	Conditional class prediction		

Table 3. Definition of the loss function.

Equation (5) is the loss function, which has five parts. The first is the loss of center coordinates of the bounding box. To all real objects, square loss is taken between the predicted values and its ground truth. The second is the prediction loss of length and width of the bounding box. The square root is chosen, since the big deviations are usually much bigger than small ones, and the square root could avoid deviations of small bounding boxes. The third and the fourth parts represent the loss functions of confidence scores of all bounding boxes. And the last one is the class prediction of recognition.

 λ coord is the weight of the importance of losses related to coordinate predictions. Increasing this weight can help improve the average accuracy. λ noobj can decrease the penalty loss of a bounding box without an object, as many bounding boxes do not contain objects. Here, the default values λ coord = 5 and λ noobj = 0.5 are set according to Reference [20].

YOLO is first pretrained on Pascal VOC 2007 and 2012 datasets and then finetuned on our own datasets to make it adapt to our scenarios. Our training images are collected by the camera of the system and labeled using labelImg (https://github.com/tzutalin/labelImg), which is a widely used annotation tool for object detection. YOLO is run with CUDA (Compute Unified Device Architecture) to ensure its real-time performance. The whole application process consists of six steps, as Figure 7 shows:



Figure 7. Detection of the target object.

The aim is to detect and identify many common objects, such as the person, bird, and plane in the images. Firstly, training datasets should be prepared in advance. The locations and class labels of all possible objects should be labeled manually. Secondly, the hyperparameters should be changed according to the meta information of our datasets. Then, the number of categories and categories' names are modified in src/yolo.c; relevant induced changes should be made to the penultimate layer of YOLO (with parameters modified in src/yolo_kernels.cu). The hyperparameter relevant to the output and the number of classes are also modified in the configuration file cfg/yolo.cfg. The initial parameters of the network are set as the pretrained values trained on Pascal VOC 2007 and 2012 datasets. Finally, the YOLO is trained on our dataset. We identified our target objects in rectangles of different colors by using the trained model.

4. Experimental

The proposed demonstration-driven system was tested experimentally, as shown in Figure 8. There were three workpieces (cubes with a side length of 1 cm) colored red, green, and blue. One key step of the assembly task is taking these workpieces from an initial position to a target position quickly and accurately. As a demonstration, we placed each workpiece by hand from the initial position to the target position, which was captured by the camera. Then the manipulator was controlled to repeat this process automatically. In the target position, the three cubes were stacked in order stably. The position error was within 5% of the side length of the cube. The whole process is shown in Figure 9.

Two additional experiments were performed to demonstrate that our system can quickly adapt to multiple new tasks. In the experiment on workpiece placement shown in Figure 10a, the three workpieces were moved from the initial position (side by side) to the target position (45° oblique angle) in a certain order. The first two images show the human's demonstration and the last two show how the manipulator repeating the task. The position error of workpiece placement was less than 5%. In the experiment on coin manipulation shown in Figure 10b, the manipulator grabs the coin from the initial position and quickly drops it to a box through a narrow gap. The thickness of coin is 1.9 mm and the gap width of box is 2 mm, which means that the position error of the manipulator is within 0.1 mm.

We also have a graphic user interface for the usage of the manipulator. The vision-guided manipulators were controlled to take an object (a blue man toy) to its target position (white circle), as shown in Figure 11.







Figure 9. The whole imitation process of the manipulator in the experiments.



(a)



Figure 10. Two experiments on workpiece placement and coin manipulation. (**a**) Experiment on workpiece placement. (**b**) Experiment on coin manipulation.



Figure 11. Testing whole process of target object detection.

5. Discussion and Conclusions

This paper introduces a demonstration-driven approach for a vision-guided automatic assembly system to make it easily adapt to performing various different tasks. The manipulator seems to be the "hand" of the system and camera seems to be the "eyes". Thus, the manipulator has the ability to learn the motion trajectory demonstrated by humans and reproduce it quickly with processed visual information and the generated control signals. In flexible manufacturing, demonstration-driven manipulators are able to perform more complex tasks under the guidance of vision, and potentially may cooperate with other manipulators. At the same time, the system can also be applied to intelligent sorting. The accuracy and efficiency of cargo picking may be improved by using the proposed approach. The application and extension of the proposed approach and system in the future will be an important research direction and will greatly help to make such robots more intelligent.

Author Contributions: Q.W. and C.Y. conceived and designed the experiments; Q.W., W.F. and Y.Z. performed the experiments; Q.W. wrote the paper.

Acknowledgments: Thanks to Xiaolong Bai, Ding Zhao, Xiaoyu Tan, Shujian Sun, Shangyang Wu and Ping Jiang for their help in the discussion and writing of the paper.

Conflicts of Interest: The authors declare no conflict of interest.

References

- 1. Brogårdh, T. Present and future robot control development—An industrial perspective. *Annu. Rev. Control* **2007**, *31*, 69–79. [CrossRef]
- Luo, Y.; Wei, W.; Wang, X. Research for Hole-Peg Assembly Based on Machine Vision and Flexible Mechanism. In Proceedings of the 2010 International Conference on Mechanic Automation and Control Engineering, Wuhan, China, 26–28 June 2010; pp. 5326–5330.
- Song, H.C.; Kim, Y.L.; Song, J.B. Automated Guidance of Peg-in-Hole Assembly Tasks for Complex-Shaped Parts. In Proceedings of the 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems, Chicago, IL, USA, 14–18 September 2014; pp. 4517–4522.
- 4. Qiao, H.; Wang, M.; Su, J.; Jia, S.; Li, R. The Concept of "Attractive Region in Environment" and its Application in High-Precision Tasks With Low-Precision Systems. *IEEE/ASME Trans. Mechatron.* **2015**, *19*, 2311–2327. [CrossRef]

- Michalos, G.; Makris, S.; Papakostas, N.; Mourtzis, D.; Chryssolouris, G. Automotive assembly technologies review: Challenges and outlook for a flexible and adaptive approach. *CIRP J. Manuf. Sci. Technol.* 2010, 2, 81–91. [CrossRef]
- 6. Liu, C.; Qiao, H.; Su, J.; Zhang, P. Vision-Based 3-D Grasping of 3-D Objects With a Simple 2-D Gripper. *IEEE Trans. Syst. Man Cybern. Syst.* **2014**, 44, 605–620. [CrossRef]
- 7. Ren, S.; Yang, X.; Xu, J.; Wang, G.; Xie, Y.; Chen, K. Determination of the base position and working area for mobile manipulators. *Assem. Autom.* **2015**, *36*, 80–88. [CrossRef]
- 8. Yamataka, M.; Kuga, T.; Takayama, T.; Furukawa, M.; Ishida, J. Robot Assembly System for LCD TV Using Cooperative Force Control. In Proceedings of the 2008 34th Annual Conference of IEEE Industrial Electronics, Orlando, FL, USA, 2008; pp. 3443–3448.
- 9. Allen, P.K.; Timcenko, A.; Yoshimi, B.; Michelman, P. Automated tracking and grasping of a moving object with a robotic hand-eye system. *IEEE Trans. Robot. Autom.* **1993**, *9*, 152–165. [CrossRef]
- 10. Fontanelli, D.; Moro, F.; Rizano, T.; Palopoli, L. Vision-Based Robust Path Reconstruction for Robot Control. *IEEE Trans. Instrum. Meas.* **2014**, *63*, 826–837. [CrossRef]
- 11. Laptev, I. Improving Object Detection with Boosted Histograms. *Image Vis. Comput.* **2009**, *27*, 535–544. [CrossRef]
- Aivaliotis, P.; Zampetis, A.; Michalos, G.; Makris, S. A Machine Learning Approach for Visual Recognition of Complex Parts in Robotic Manipulation. In Proceedings of the 27th International Conference on Flexible Automation and Intelligent Manufacturing (FAIM 2017), Modena, Italy, 27–30 June 2017; Volume 11, pp. 423–430.
- 13. Zhang, C.; Xue, Z.; Zhu, X.; Wang, H.; Huang, Q.; Tiane, Q. Boosted random contextual semantic space based representation for visual recognition. *Inf. Sci.* **2016**, *369*, 160–170. [CrossRef]
- 14. Tsarouchi, P.; Matthaiakis, S.A.; Michalos, G.; Makris, S.; Chryssolouris, G. A method for detection of randomly placed objects for robotic handling. *Int. CIRP J. Manuf. Sci. Technol.* **2016**, *14*, 20–27. [CrossRef]
- Balabantaray, B.K.; Biswal, B.B. Part Identification in Robotic Assembly Using Vision System. In Proceedings of the Sixth International Conference on Machine Vision (ICMV 2013), London, UK, 16–17 April 2013. [CrossRef]
- Sahu, O.P.; Balabantaray, B.K.; Patle, B.; Biswal, B.B. Part Recognition Using Vision and Ultrasonic Sensor for Robotic Assembly System. In Proceedings of the 2015 IEEE Student Conference on Research and Development (SCOReD), Kuala Lumpur, Malaysia, 13–14 December 2015; pp. 145–149.
- 17. Natsagdorj, S.; Chiang, J.Y.; Su, C.-H.; Lin, C.; Chen, C. Vision-based Assembly and Inspection System for Golf Club Heads. *Robot. Comput. Integr. Manuf.* **2015**, *32*, 83–92. [CrossRef]
- 18. Qin, Z.; Wang, P.; Sun, J.; Lu, J.; Qiao, H. Precise Robotic Assembly for Large-Scale Objects Based on Automatic Guidance and Alignment. *IEEE Trans. Instrum. Meas.* **2016**, *65*, 1398–1411. [CrossRef]
- Al-masni, M.A.; Al-antari, M.A.; Park, J.M.; Gi, G.; Kim, T.Y.; Rivera, P.; Valarezo, E.; Choi, M.T.; Han, S.M.; Kim, T.S. Simultaneous detection and classification of breast masses in digital mammograms via a deep learning YOLO-based CAD system. *Comput. Methods Programs Biomed.* 2018, 157, 85–94. [CrossRef] [PubMed]
- Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
- 21. Singh, A.; Singla, A.; Soni, S. Extension of D-H parameter method to hybrid manipulators used in robot-assisted surgery. *Proc. Inst. Mech. Eng. H* **2015**, *229*, 703–712. [CrossRef] [PubMed]
- 22. He, B.; Han, L.; Wang, Y.; Huang, S.; Liu, L. Kinematics analysis and numerical simulation of a manipulator based on virtual prototyping. *Int. J. Adv. Manuf. Technol.* **2014**, *71*, 943–963. [CrossRef]
- 23. Zhong, S.; Liu, Y.; Chen, Q. Visual orientation inhomogeneity based scale-invariant feature transform. *Expert Syst. Appl.* **2015**, *42*, 5658–5667. [CrossRef]
- 24. Al-khafaji, S.L.; Zhou, J.; Zia, A.; Liew, A.W.C. Spectral-Spatial Scale Invariant Feature Transform for Hyperspectral Images. *IEEE Trans. Image Process.* **2018**, *27*, 837–850. [CrossRef] [PubMed]



© 2018 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).