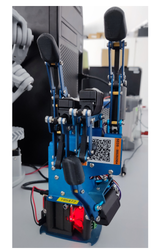
ROBT310 Final Project: Robotic hand control by hand gesture tracking in different light conditions and noise reduction with preprocessing algorithms

Rikhat Akizhanov, Sanzhar Kabitkanov, Saparkhan Kassymbekov,
Kamila Zhumakhanova and Zhanat Kappassov *Member, IEEE*

***Abstract*—**
In this paper the control of a robotic hand using hand gesture recognition is described. The recognition of the hand palm is aimed to be performed in different light conditions and with different types of noise. In particular, we added Gaussian, salt and pepper and sinusoidal noise. To get rid of these noises filtering in spatial and frequency domain were applied. Also we added laplacian to sharpen the edges for improving the conditions for hand palm recognition. To control the hand, hand landmarks from the input video from a web camera are detected and tracked, and the angles are then calculated using the coordinated of the landmarks. Then, these angles are sent to the robotic hand.

***Index Terms*—** **Hand gesture tracking, Robot hand control, OpenCV, MediaPipe, Noise Reduction**



I. Introduction

The hand gesture tracking for robotics applications is not a new idea and a number of attempts to implement this is observed in the literature. An important salient point is that majority of those hand tracking attempts had some limitations in terms of light conditions and noise processing. That is why this paper is aimed to improve existing hand tracking implementations with image processing algorithms that allows to track the hand in different light conditions and noise levels. In particular, we want to write a hand tracker program that uses hand landmark identification using MediaPipe and OpenCV libraries and is advanced with image processing techniques.

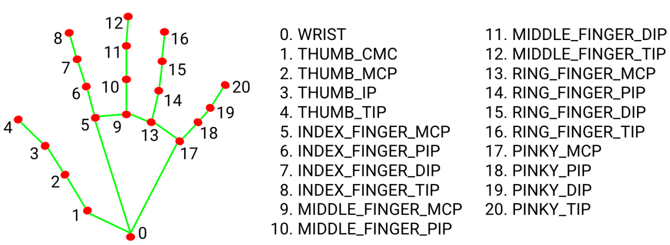


Fig. 1: 21 Hand landmark identification using MediaPipe library [1]

Then this hand tracker will be used to control a robotic arm. We plan to replicate the gestures from the camera output on the robotic arm.

Submitted April 2022 and revised April 2022.
Corresponding authors: Rikhat Akizhanov, Sanzhar Kabitkanov, Saparkhan Kassymbekov, Kamila Zhumakhanova
Rikhat Akizhanov with Nazarbayev University, Nur-Sultan, Kazakhstan (e-mail:rikhat.akizhanov@nu.edu.kz).
Sanzhar Kabitkanov with Nazarbayev University, Nur-Sultan, Kazakhstan (e-mail: sanzhar.kabitkanov@nu.edu.kz).
Saparkhan Kassymbekov with Nazarbayev University, Nur-Sultan, Kazakhstan (e-mail:saparkhan.kassymbekov@nu.edu.kz).
Kamila Zhumakhanova with Nazarbayev University, Nur-Sultan, Kazakhstan (e-mail:kamila.zhumakhanova@nu.edu.kz).
Z. Kappassov is with Nazarbayev University, Nur-Sultan, Kazakhstan (e-mail: kappassov@isir.upmc.fr).

II. Feasibility Analysis

Tracking the hand and then using position to control the robotic hand is a demanding task that might have broad range of applications. There exists numerous works and projects that proposed different ways to track and estimate hand positions from camera. Consequently, exists many ways to control the robotic hand knowing the positions. Our project propose to use several already implemented computer vision algorithms for hand pose estimation, however most of the project are in done in normal conditions and using high quality camera. Our project will try to utilize image processing algorithms in order make tracking possible in different light conditions and with different types of the noise present. Some of the examples are denoising with Gaussian filters, sharpening by reduction of low frequencies in frequency domain, histogram normalization and others. In addition, Kalman filters can be applied in order to reduce shaking noise of the hand between frames. Such image manipulations can make tracking in worse light conditions better and also add smoothness to the movement. Also very inexpensive cameras can be used. Next, from landmarks angles of fingers can be computed and this data used for robotic hand manipulation. All this parts were implemented and studied by other researches and our plan is to bring it all together and compare different algorithms. Summing all together, it can be concluded that this project is feasible.

III. Related Robot Conrol frameworks

There are a number of various attempts to control a robot hand with the visual data of the hand motion presented in the literature. Some of them are using deep learning classification models, pose estimation by hand landmarks, etc. Some present successful results, while others show the weaknesses of certain approaches.

For example, the research done by the Chittagong University of Engineering and Technology of Bangladesh has performed

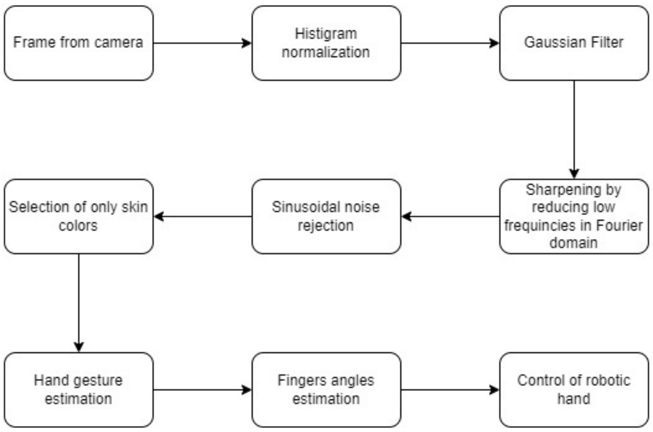


Fig. 2: Pipeline of proposed algorithm

the robot hand control by using pose estimation. It was done by detecting fingertips contour and cropping them from the background image using state of the art OpenCV library. Then, the detected fingertips were marked by landmarks, which is used for the control of the robot hand [2] . This research suggests that the use of landmarks is an effective method for our research.

Some papers has done the same set of operations for extracting the features from the image but has done a gesture recognition without hand landmarks. One of the approach shown in the literature is calculating the centroids of the fingers in the region of interest [3]. One of the observed limitations is that it is suitable for gesture recognition but not for real-time hand motion recognition. That is why, it is not chose as the method for this paper.

Another research performed by the Istanbul Gelisim University also cropped the hand image from the background within the region of interest with OpenCV library but used another evaluation method. Their program worked in such a way that it measured and counted the empty spaces between the fingers, which was then analyzed by the deep learning algorithm to classify the gestures. Although the approach gives promising expectations, it turned out that the approach is not effective. The reason is that the evaluation by the empty spaces between the fingers had several disadvantages. For example, it cannot understand the gesture if the distance between the camera and the had is to large or of the angle between the fingers is too small [4].

[5] In this paper authors implemented model that has several layers: camera module that retrieves image frames, subtraction of background, removal of face, canny edge segmentation, skin color extraction, morphology operations and image smoothing, depth segmentation, detection module, contour extraction and polygon approximation, palm center and radius determination, Setting Region of Interest (ROI) and finding min enclosing circle, Convex hull and convexity defects extraction, Determining hand/fingertips location and direction, Simple hand gesture recognition, Simple finite state machine, Kalman filter, Interfacing module, Translating hand and fingers into functional inputs. Framework utilizing Visual Studio 2008 with C as the programming language was used. EmguCV library (OpenCV wrappers for C) for the picture processing portion is the main library. CL-Eye Stage Driver and OpenNI system were utilized for interfacing the PS3 Eye Camera and Kinect camera. TUIO system was utilized for sending TUIO messages between applications.

[6] The authors implemented Gesture Replicating Arm. Design consists of the webcam input, the software that is working through OpenCV library, the AtMega 168 microcontroller. Also authors analysed algorithms for background subtraction, color detection and contour identification. Some of them are Codebook, Skin Detection, Color co-occurence, and Mixture of Gaussians models for background subtraction. Meanwhile for wrist and elbow detection they considered calculations for position and moment. Finally, for contour detection techniques covered are Convex hull, Quickhull, Graham Scan, Monotone Chain, and Prunning. All of this expands our understanding of how we can implement our own gesture replicating arm.

[7] Team of Gaarg, Aggarwal and Sofat made a secondary study on visual recognition of hand. Section of interest for our research is the section about available algorithms for hand posture and gesture recognition. Here authors discussed advantages and disadvantages of a various model-based hand tracking systems such as DigitEyes, construction of a mesh through PCA from training examples, a sequential Monte Carlo tracking algorithm and Unscented Kalman filter based models. Also Bayesian approaches were considered and compared with Kalman fiter approaches. Different approaches based on appearance that authors studied include skin colored regions detection, AdaBoost learning algorithm, as well as eigenspace and statistical approaches. For our research we decided to work with appearance based approaches.

[8] Authors of this paper take the input from single camera and convert it to binary image that contains information about contours of the hand. This is done with the help of YCbCr colour space instead of RGB. Latter convex-full contour is obtained. All this data, especially convexity defects, are used in order to make fingertips and palm detection. Authors propose that overall system have low computational cost.

[9] Reviewing this work, it is close to what we aiming to do in our project. Authors took low-cost ordinary camera and with the use of OpenCV recognized hand gestures that will control 3D-printed robotic arm. They removed the background and noise by using technique of background subtraction. Also, the Gaussian blur was applied in order to remove white noise and then image is converted to HSV colour space that helps to separate only skin colour. Location of unobserved parts of the hand is estimated using Hidden Markov method based on previously observed locations.

[10] In this paper frames are taken from the camera with 3 FPS. Then area of hand gesture is obtained by using differences between frames. Gesture is determined by by using pattern matching with PCA algorithm. This method is used instead of machine learning because it is much faster in running time. Results are then used to create control instructions for specified robotic system.

We did the literature review on the subject of motion tracking using Opencv in general. For example, Garofalaki et al. [11] worked on the motion tracking of an object of a predefined color. The image captured from an Android device underwent several image processing procedures such as color space conversion, color filtering, noise removal and largest contour estimation and as result the contour of an object was drawn and then tracked. Their project was aimed on the moving of a robot, in particular the task of the robot was to follow the object that is tracked by the color.

Some of the research that we have read focused on the creation of real time hand gesture recognizer using machine learning techniques, such as the work of Gurav and Kadbe [12]. The procedure that they used is creating the large dataset of images with predefined hand gestures and training the AdaBoost neural network on it. With Adaboost and rectangular bounding box drawing the maximum accuracy was 70 percent. However the authors found a way to improve the results by using a convex hull algorithm, which finds the location of fingertips and then draws the contour of a hand palm using the points on fingertips. The accuracy that was achieved by adding convex hull algorithm is 92 percent.

Review of the available literature suggests that the pose estimation by using hand landmarks is the most feasible solution for our task. It provides fast and real-time transmission of data about the position of fingers to the robot hand.

IV. Methodology

Implementation of the project has three main parts such as pixel to angle conversion, robotic hand control, and hand gesture recognition with denoising.

A. Pixel to Angle Conversion

For calculating angles which joints of the robotic hand have to replicate dot product between two vectors was used. Firstly, coordinates of joints were found. Then two vectors from coordinates were obtained as the difference between them. These vectors shared the same point of intersection which is a coordinate of hand's joint. After dot product between two vectors was calculated by the formula (1), desired angle was obtained by the equation (2).

$$a \cdot b = a_x b_x + a_y b_y + a_z b_z \tag{1}$$

$$cos\theta = \frac{a \cdot b}{|a||b|} \tag{2}$$

Then conversion from radians to degrees and handling angles larger than 180 degrees was done. For more smooth angle appearances in the video frames their mean value from 5 frames is calculated. Finally, the function *putText* in *CV2* library was used for displaying joint angles . The final results are in Figure 3.

B. Robotic hand control

After having angles of fingers position, this data will be used to control position of the robotic hand. For our project we used AR10 robotic hand which has 10 degrees of freedom (DOF). Human hand, in comparison has 27 DOFs [13]. MediaPipe provide 21 DOFs. Thus, robotic hand will have reduced information about hand real position, but it is still enough to estimate hand pose.

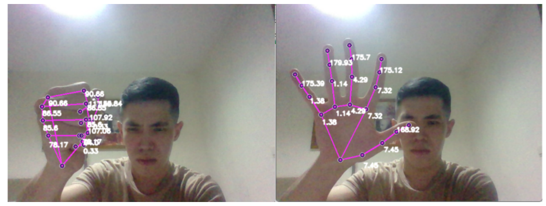


Fig. 3: Demonstration of calculated angles

C. Denoising

To make the project be suitable for different light conditions and resistive for different types of noise, different types of noise were applied to the frames of the video and then were removed as much as possible. Regarding random noise, we added Gaussian and salt and pepper noise to the RGB frames that we receive from the web camera. To filter Gaussian noise, a Gaussian filter was applied. For the purpose of filtering salt and pepper noise a median filter was used. The amount of noise was controlled using trackbars.

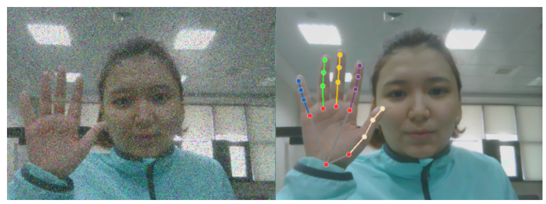


Fig. 4: Illustration of adding and filtering salt and pepper noise

As it can be seen from figures 4 and 5, the filtered images are of a reasonable quality for the hand palm and landmarks to be detected and tracked. As it can be seen from figure 4, median filter performs quite well with large amount of salt and pepper noise. However, from figure 5 a poorer, but still reasonable performance of Gaussian filter is evident.

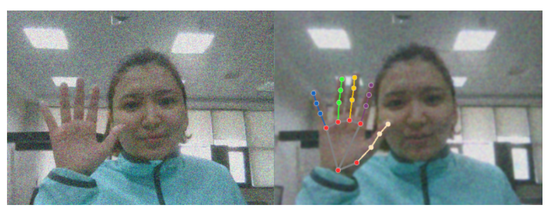


Fig. 5: Illustration of adding and filtering Gaussian noise

Another technique that is added in this project is removing sinusoidal nose using Fast Fourier Transform (FFT). In order to demonstrate this, the sinusoidal noise was added to the image stream from the camera. Then, it was removed from the FFT of the image. The algorithm of adjustable filter for sinusoidal noise is the application of the mask that finds the coordinates of harmonics of sinusoids on the Fourier transform of an image and sets the values of pixels in these coordinates and their surroundings equal to 0. The result of adding and removing sinusoidal noise is depicted in figure 6. As it is evident from figure 6, the filter succeeds in its task.

D. Sharpening of edges

Another image processing feature that was added to the project is sharpening of edges in spatial domain using Lapla-

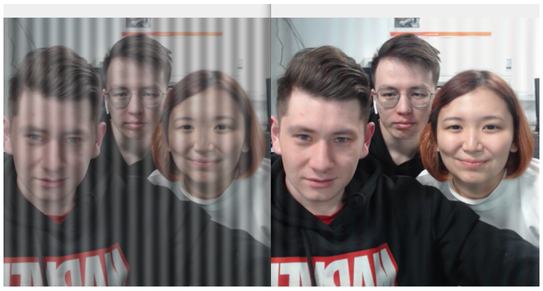


Fig. 6: Addition of sinusoidal noise and the result of its removal

cian. A demonstration of using Laplacians can be seen in figure 7. From figure 7 the effect of sharpening is evident.

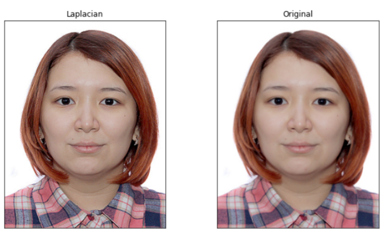


Fig. 7: The comparison of original image and an image with laplacian applied

V. Experimental Platform

For our experiments we used a AR 10 robotic hand [14]. AR10 has 10 degrees of freedom actuated by 10 servo motors which are controlled by Pololu Maestro board. With the use of serial port, position, speed and acceleration of each servo motor can be controlled from python script. Each joint has a range of motion close to 0-90 degrees, position settings of servos has range from 4000 to 8000. Angles converted to positions by linear relation. The restriction of motion for thumb is added in order to avoid collisions of the fingers. Another approach that can be used is to read the current value for each servo and if current starts to exceed some threshold, it means collision happened and motion must be stopped.

VI. Experiments and Results

Since our project consists of image processing and hand control parts, it seems reasonable to divide this section into 2 parts. To make the image processing part more visually attractive and easy-to-understand, we made the window with the trackbars where the user can decide which type of noise to add, how much noise to add, which type of filter to apply and whether to add laplacian to sharpen the edges or not. As it can be seen from figures 4,5 and 6, the filters fulfill their task and filter the image so that the hand palm and fingers are detected and tracked. However, there is a constraint that Gaussian filter variance and variance of the Gaussian noise should coincide for the filter to work properly. Median filter, in turns, works properly for reasonably large and small amounts of salt and pepper noise. Also we made the frequency of sinusoidal noise and the filtering mask adjustable. Apart from that, the Laplacian noticeably sharpens the edges and thus improves the conditions for the hand recognition. Regarding hand control, the robotic hand was controlled using the angles derived from the Mediapipe hand palm. To avoid the negative effect of shaking of the hand and coordinates of the hand landmarks, the averaging of last 20 values of angles was applied. As a result, the hand works properly, in particular it resembles the hand gestures quite quickly and as closely as it can.

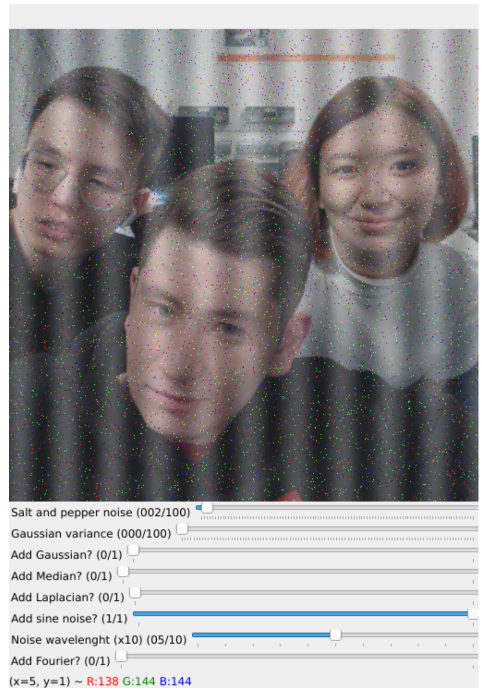


Fig. 8: Interface of the program

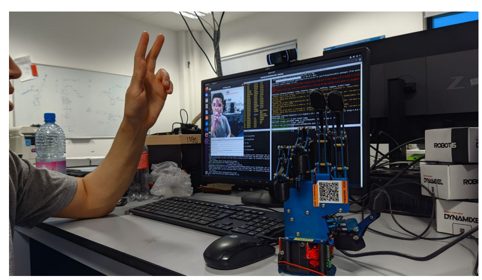


Fig. 9: Demo of the project

VII. Discussion and Conclusion

In conclusion, the overall purpose of the project was achieved. The robot hand repeats the movements of the real hand in harsh environments. The noises such as Salt and Paper, Gaussian noises were removed by median and Gaussian filters respectively with the program being able to transmit the hand gestures to the robot hand. Another type of noise that was applied and removed was the sinusoidal noise. It was removed by the use of Fast Fourier Transform. The experiments has shown that the the program works after the noise removal. Some additional enhancements such as Laplacian were applied.

Some ways to improve the project were identified during carrying out the research. First of all, it is suggested to use the Kalman filter instead of average filter in the calculations of the hand angles from the frame. It is advised to use Kalman filter in future works. Second, it was found out that it would be better to create some constraints on the robot hand movements to not allow the fingers of the robot hand to collide. The thumb and the rest of the fingers collide during repeating some gestures. This is not a good practice of repeating the hand gestures. That is why it should be fixed.

The research has faced several limitations. One of them is that the robot hand is pretty outdated and has only 10 degrees of freedom (while human hand has 27), which substantially decreased the overall number of gestures that the program may repeat. Moreover, the robot hand receives the data about the hand in 2D frame, while the robot hand moves in the 3D frame. Although some good mathematical calculations were made to replicate 2D movements in 3D, it is not enough to reach high accuracy. Another limitation is that the program was run in the computer with limited CPU capabilities. When the program is executed with all the noise removal programs, signal transmission as well as the frame rate, decreases dramatically. Better CPU performance would allow to avoid this.

References

[1] "Hands," 2020. [Online]. Available: https://google.github.io/mediapipe/solutions/hands.html

[2] J. Hossain Gourob, S. Raxit, and A. Hasan, "A robotic hand: Controlled with vision based hand gesture recognition system," in *2021 International Conference on Automation, Control and Mechatronics for Industry 4.0 (ACMI)*, 2021, pp. 1–4.

[3] S. M. Achari, S. G. Mirji, C. P. Desai, M. S. Hulasogi, and S. P. Awari, "Gesture based wireless control of robotic hand using image processing," *International Research Journal of Engineering and Technology (IRJET)*, vol. 5, no. 5, p. 3339–3345, May 2018.

[4] A. Çetinkaya, "Controlling a robotic arm using hand recognition software," *International Journal of Engineering Technologies IJET*, vol. 5, pp. 151 – 174, 05 2019.

[5] H.-S. Yeo, B.-G. Lee, and H. Lim, "Hand tracking and gesture recognition system for human-computer interaction using low-cost hardware," *Multimed Tools Appl*, vol. 74, p. 2687–2715, 2015.

[6] S. Karamchandani, S. Sinari, A. Aurora, and D. Ruparel, "The gesture replicating robotic arm," in *2013 International Symposium on Computational and Business Intelligence*, 2013, pp. 15–19.

[7] P. Garg, N. Aggarwal, and S. Sofat, "Vision based hand gesture recognition," *World Academy of Science, Engineering and Technology*, vol. 49, p. 972–977, 2009.

[8] S. U. Rahman, Z. Afroze, and M. Tareq, "Hand gesture recognition techniques for human computer interaction using opencv," *International Journal of Scientific and Research Publications*, vol. 4, no. 12, p. 1–6, Dec 2014.

[9] J. Paterson and A. Aldabbagh, "Gesture-controlled robotic arm utilizing opencv," *2021 3rd International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)*, 2021.

[10] J. L. Raheja, R. Shyam, U. Kumar, and P. B. Prasad, *2010 Second International Conference on Machine Learning and Computing*. Central Electronics Engineering Research Institute (CEERI)/Council of Scientific amp; Industrial Research (CSIR), 2010, p. 12–16.

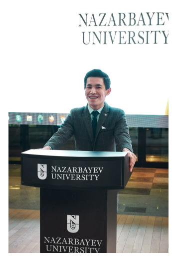
[11] Z. I. Garofalaki, J. T. Amorginos, and J. N. Ellinas, "Object motion tracking based on color detection for android devices," *World Academy of Science, Engineering and Technology International Journal of Computer and Information Engineering*, vol. 9, no. 4, p. 999–1002, 2015.

[12] R. M. Gurav and P. K. Kadbe, *2015 International Conference on Industrial Instrumentation and Control (ICIC)*. College of Engineering Pune, 2015, p. 974–977.

[13] G. ElKoura and K. Singh, "Handrix: Animating the human hand," *Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pp. 110–119, 01 2003.

[14]

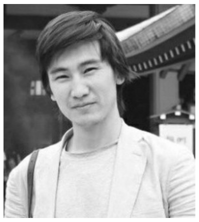
Rikhat Akizhanov Nazarbayev University Junior Robotics student. He currently works as a research assistant in ALARIS lab.



Sanzhar Kabitkanov Nazarbayev University Junior Robotics student. He currently works as a research assistant in ALARIS lab.

Saparkan Kassymbekov Nazarbayev University Junior Robotics student. He currently works as a research assistant in ALARIS lab.

Kamila Zhumakhanova Nazarbayev University Junior Robotics student. She currently works as a research assistant in ALARIS lab.



Zhanat Kappassov (M'16), received the Specialist degree in radioengineering from the Tomsk State of Control Systems and Radioelectronics (TUSUR), Tomsk, Russia. Afterward, he worked in the Industrial Technology Research Institute (ITRI), Taiwan. He received his Ph.D. in robotics from the Institute of Intelligent Systems and Robotics (ISIR), Sorbonne Université (formerly Université Pierre et Marie Curie), Paris, France, in 2017. He is currently an assistant professor of robotics at Nazarbayev University, Nur-Sultan, Kazakhstan. His current research interests focus on tactile sensing for robot physical interaction and dexterous manipulation. He serves regularly as a reviewer for the IEEE Transactions on Robotics and IEEE conferences. His PhD thesis was nominated as the Best PhD Thesis 2017 by the GDR Robotique association in France.