Nadine : a social robot that can localize objects and grasp them in a human way

Thalmann, Nadia Magnenat; Tian, Li; Yao, Fupin

2017

Thalmann, N. M., Tian, L., & Yao, F. (2017). Nadine : a social robot that can localize objects and grasp them in a human way. Frontiers in Electronic Technologies, 1-23. doi:10.1007/978-981-10-4235-5_1

https://hdl.handle.net/10356/139003

https://doi.org/10.1007/978-981-10-4235-5_1

Nadine: A Social Robot that Can Localize Objects and Grasp Them in a Human Way

Nadia Magnenat Thalmann, Li Tian and Fupin Yao

Abstract What makes a social humanoid robot behave like a human? It needs to understand and show emotions, has a chat box, a memory and also a decision-making process. However, more than that, it needs to recognize objects and be able to grasp them in a human way. To become an intimate companion, social robots need to behave the same way as real humans in all areas and understand real situations in order they can react properly. In this chapter, we describe our ongoing research on social robotics. It includes the making of articulated hands of Nadine Robot, the recognition of objects and their signification, as well as how to grasp them in a human way. State of the art is presented as well as some early results.

Keywords Robotic hand · 3D printing · Object recognition · Trajectories planning · Natural grasp

1 Introduction

In robotics, the uncanny valley [1] is the hypothesis that human replicas that appear almost but not exactly like real human beings elicit eeriness and revulsion among some observers. Unfortunately, most of the humanoid machines fall into the uncanny valley as they are not exactly like a human (Fig. 1). The uncanny valley is the region of negative emotional response towards robots that seem "almost" human. Movement amplifies the emotional response.

Humanlike Behaviors, such as grasp are vital to solving this problem. Some researchers have done much work on robot grasp, but they usually focus their attention on grasp for industrial robot hands. Most of the hands designed for industrial robots are very different from real human hands. Industrial robot hands

N.M. Thalmann (✉) · L. Tian · F. Yao
Institute for Media Innovation (IMI), Nanyang Technological University (NTU), Singapore, Singapore
e-mail: nadiathalmann@ntu.edu.sg

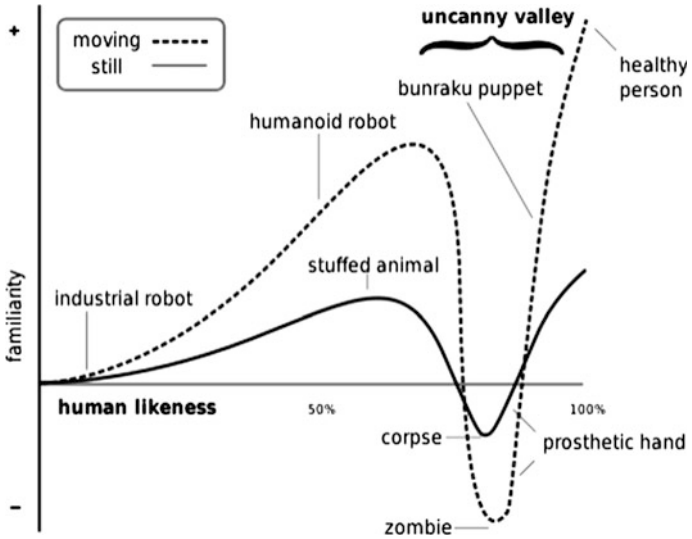


Fig. 1 Uncanny valley [1] Proper hypothesized emotional response of subjects is plotted against anthropomorphism of a robot, following Mori's statements

are usually simpler and may have only two or three fingers, but real humans interact with objects by all fingers most of the time. Therefore, grasping area detection and human-like hands for social robots are more complicated and difficult. In our chapter, we focus on two areas, the making of a proper human-robotic hand and grasping area detection (localization) for pinch grasp. We take pinch grasp as the first example which is most widely used in our common life. Pinch grasp means we hold objects with our thumbs and the opposed fingers.

Grasping is one of the most important abilities for a human being when interacting with objects around them. For humans, it seems quite easy to know what kind of objects there are and grasp them. It is almost immediate. However, for robots, it is quite difficult to grasp something. First, the robots have to know what objects are in front of them. Second, they need to know the location of the objects. Even if the robots know the above information, when interacting with large objects and performing some ordinary tasks, such as picking up a cup full of water, it is difficult to do it well. Robots should neither put their fingers in the water nor cover the top of the cup with their palm. Therefore, the robots should know the location of the proper grasping area and orientation for grasping. In the physical world, the categories and location of objects and grasping area location are unknown. Thus vision is desirable because it provides the information which is more accurate than other information resources. Information based on vision is very necessary. In this chapter, we solve the above mentioned problem with information captured through vision. The entire problem can be divided into two tasks: object recognition, grasping area detection and localization. We will construct a shared neural network for these two tasks. The input is images captured by the Kinect, and the output is the

Fig. 2 Proper grasping area: handle [2]

Fig. 3 Nadine robot at IMI, NTU

categories of objects detected, location and orientation of the proper grasping area (Fig. 2).

A robotic hand is a type of mechanical hand, usually programmable, with similar functions as a human hand. The links of such a manipulator are connected by joints allowing either rotational motion (such as in an articulated robot) or translational (linear) displacement. In this chapter, we describe how we have designed a robotic hand for our social robot Nadine (Fig. 3) [3]. When designing the hand, we considered the weight of the robotic hand, the compatibility with Nadine robot and the

cost of time and money. Therefore, we made our robotic hand look and work like a real hand as much as possible. After creating the new hand, we tested and modified it to get better grasping force and fingers' trajectories. In the end, the hand was then tested with the Nadine robot to verify our new algorithm of natural grasping approach.

The contributions of this chapter can be summarized as follows:

- A framework which detects object localization and the grasping area in a shared neural network. It is quicker than two single networks and real time.
- A new design of robotic hand for Nadine robot.

The layout of this chapter is as follows. Next section is the related work. Section 3 shows details about object localization and grasping area detection. The design of robotics hand is presented in Sect. 4. Section 5 describes our grasping experiments. Future work and conclusions are in Sect. 6.

2 Related Work

2.1 Humanoid Robot

There are lots of advanced humanoid robots in the world. The Atlas robot [4] developed by Boston Dynamics, the ASIMO [5] robot developed by Honda, the iCub [6] created by the RobotCub Consortium, etc., are all fantastic robots with various kinds of powerful functions. However, they are shelled with plastic or metal which make them look like a robot more than a human. If people stand in front of them, they may be afraid of them. To overcome this feeling, we have chosen a robot with deformable skin which is similar to a real human. Nadine is a realistic female humanoid social robot designed by the Institute for Media Innovation of Nanyang Technological University and manufactured by Kokoro Company in Japan. The robot has strong human-likeness with a natural-looking skin and hair. Kokoro Company has already produced quite a few well-known humanoid robots as the "Actroid" [7] serials. Robot "JiaJia" created by the University of Science and Technology of China is also a good sample of realistic humanoid robots [8]. Hong Kong designer Ricky Ma is also successfully constructing a life-sized robot looking like Hollywood star Scarlett Johansson [9]. They look like human beings, but they are not able to recognize objects and grasp them in a human way. To let Nadine grasp in a human way, we designed two dexterous hands and gave them the ability to grasp like a human being. In this chapter, we show our solution which includes grasp approach using vision-based information. We also first designed our 3D virtual hand to be printed later on (Fig. 4).

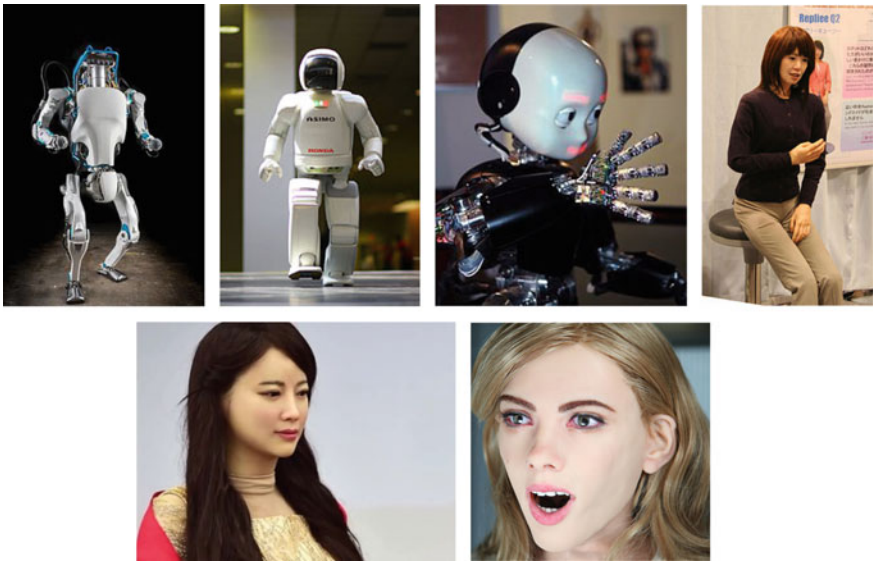


Fig. 4 Humanoid robots From *left* to *right*: Atlas robot [4], ASIMO robot [5], iCub robot [6], Actroid robot [7], JiaJia robot [8], Ricky Ma [9]

2.2 Object Recognition and Localization

Object localization, or object detection, means finding a predefined object in a set of images. One popular way to detect objects is to use regions with convolutional neural network features (R-CNN) [10] and fast versions [11, 12]. The authors divide object localization problem into two steps: generating object proposals from an image (where they are), and then classifying each proposal into different object categories (object recognition). The major drawback is repeated computation and fast R-CNN that is not fast enough. To improve the speed, we chose first YOLO [13] to eliminate bounding box proposals, but this solution did not perform very well with small objects. We then chose SSD [14] similar to YOLO, but it uses a small convolutional filter to predict object categories and bounding box locations. It allows multiple features maps to perform detection at multiple scales. It increases accuracy, especially for small objects (Figs. 5 and 6).

The second task, grasping area detection using visual information has arose people's interest during the last ten years. Saxena et al. [15] first use visual information to predict grasping region with a probabilistic model (Fig. 7). In [16], the authors try to get multiple grasping points rather than one grasping region (Fig. 8). Nearly all use local hand-designed features to learn good grasping areas. Local features only reflect part of the information, and hand-designed features usually require expert knowledge which makes it difficult to design good features. Therefore, some researchers started to use deep learning to detect grasping area.

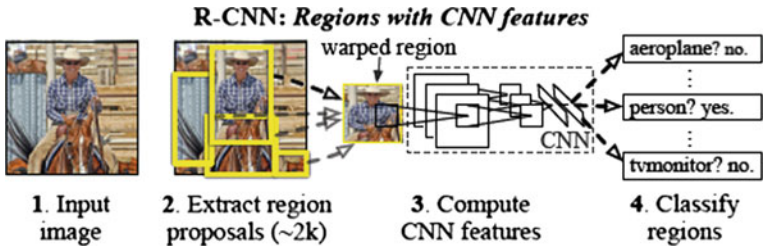


Fig. 5 RCNN workflow [10]

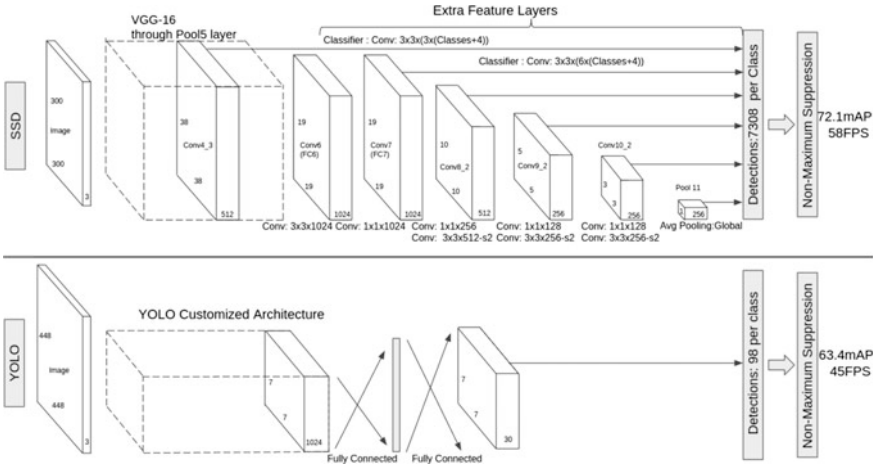
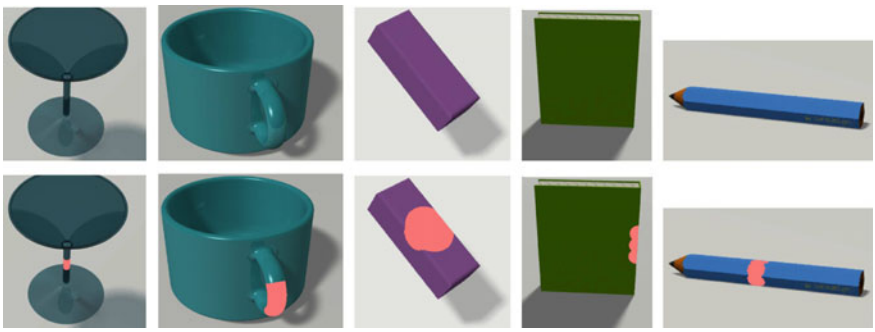


Fig. 6 YOLO [13] and SSD [14] architecture



(a) Martini glass (b) Mug (c) Eraser (d) Book (e) Pencil

Fig. 7 The images (*top row*) with the corresponding labels (shown in *red* in the *bottom row*) of the object classes used for training [15]

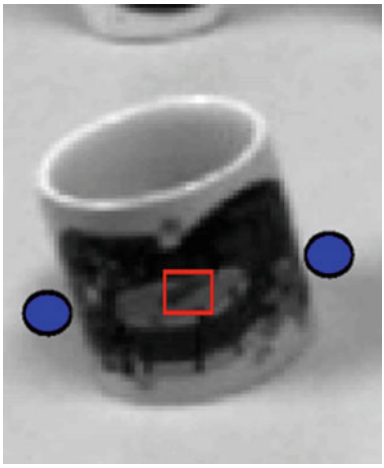


Fig. 8 Finger contact points (*blue circles*) [16]

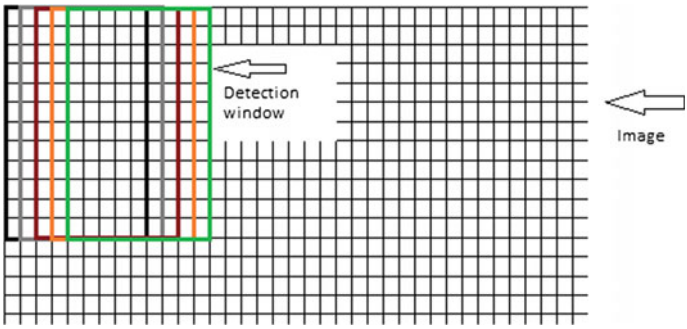


Fig. 9 Sliding window [21]

Lenz et al. [17] use convolutional networks as a classifier for grasping area detection in sliding windows. The sliding window means getting image patches when a window slides from left to right and from top to bottom and then independently classifying all image patches as being good grasping areas or not (Fig. 9). Detecting good grasping area in sliding windows is time-consuming because there are lots of image patches in an image. The authors trained their neural network model on Cornell Grasp Dataset [18] which is the most widely used datasets. Redmon et al. [19] also use the same dataset and deep convolutional network for this task. The solution is real time but not accurate, especially for small objects since it divides one image into several grids and predicts one grasp per grid. Guo et al. [20] uses similar neural network architecture which could also detect objects and grasping areas and create a new dataset. However, their solution only works when objects are put on a horizontal plane. Their dataset only contains fruits so their solution may not detect other objects in our daily life. We build a shared neural network which is similar. It can perform object recognition and grasping area

Table 1 Approaches comparison for grasping area detection

Approach	Contribution	Limitation
Robotic grasping of novel objects using vision [15]	Can get grasping points	Grasping points are not enough, more information needed, such as orientation
Deep learning for detecting robotic grasps [17]	Can get grasping parts	No object recognition
Real-time grasp detection using convolutional neural networks [19]	Combine object recognition and grasping area detection	Perform not well for small objects
Object discovery and grasp detection with a shared convolutional neural network [20]	Combine object recognition and grasping area detection	Could only grasp objects on horizontal planes
		Could only work for fruits grasp
Our approach	1. Combine object recognition and grasping area detection in a shared CNN	
	2. Perform well even for small objects	
	3. Could work for any graspable objects	
	4. Could work for objects placed everywhere	

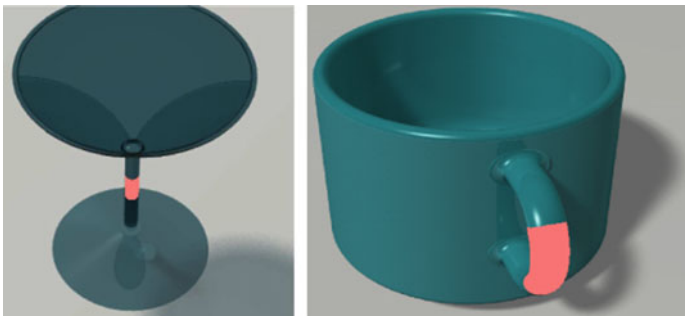


Fig. 10 Grasping region in Saxena et al. [15]

detection simultaneously and works well with any graspable objects placed on any plane, even for small objects. A detailed comparison is shown in Table 1 (Figs. 10, 11 and 12).

2.3 Humanoid Robotic Hands

For the design of robotic hands, most existing robotic arms or hands can only perform pre-programed actions, especially for industrial robotic arms. Only a few

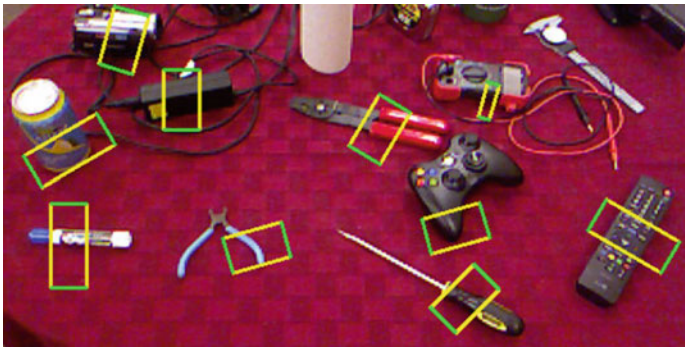


Fig. 11 Grasping *rectangle* in Lenz et al. [17]

Fig. 12 Cornell Grasp Dataset [18]

ones can do autonomous grasping, but the action is inflexible regarding handling the complex objects. Some of them show wonderful high speed and precise movement. The KUKA robot [22] can beat famous table tennis player Timo Boll (Fig. 13). Motoman-MH24 [23] is a Sword-wielding robot which beat Japanese master samurai (Fig. 13). However, these robotic hands are very different from real human hands regarding shape and behavior. Bebionic V3, Dextrus and Tact (Fig. 14) are 3 leading anthropomorphic hands. In this chapter, Inmoov hand, i-LIMB hand (Fig. 14), are also reviewed to compare with Nadine's hand as well in Sect. 4.

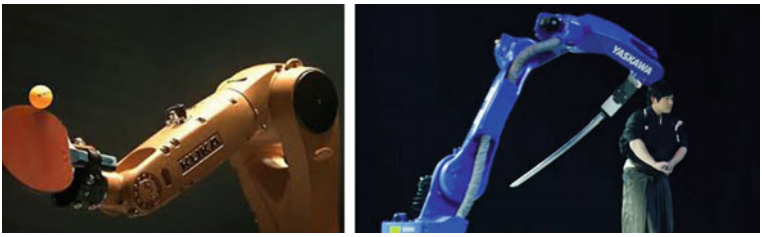


Fig. 13 The KUKA robot and Motoman-MH24 [22, 23]

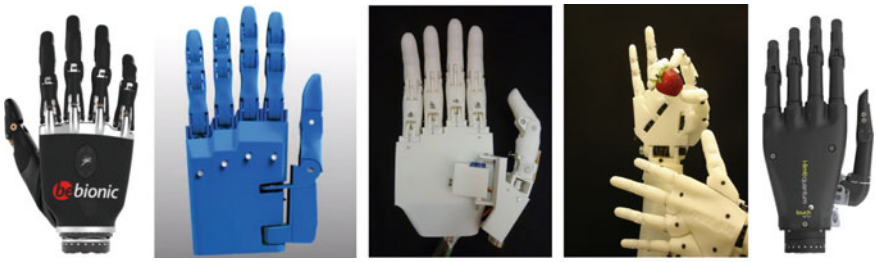


Fig. 14 (From *left* to *right*) Bebionic, Dextrus, Tact, Inmoov, and i-Limb hand [24–26]

3 Object Localization and Grasping Area Detection

3.1 Grasp Representation and Accuracy Metric

Using vision-based information, we want to detect all objects and proper grasping area. The vision nsdrf information includes images and points cloud data. We use color images and depth images from Kinect because it is accessible and easy to process. In our chapter, we focus on pinch grasp as pinch is most widely used by our human being in common life, and it is easier to study compared to other complicated grasp types which may involve multiple fingers. For pinch grasp, five-dimension representation (Fig. 15) was proposed by Lenz et al. [17].

This representation can be expressed as {x, y, θ, w, h}. x, y here are the center coordinates of the grasping rectangle. θ is the orientation angle of this rectangle. W and h are width and height of the rectangle. Figure 15 is an example of this grasp representation.

To evaluate the accuracy of our predicted grasping area, we adopted a rectangle metric used by Lenz et al. [17] and Redmon et al. [19]. The metric classifies a grasp rectangle as correct and valid if:

- The angle between the ground truth rectangle and predicted rectangle is less than 30°

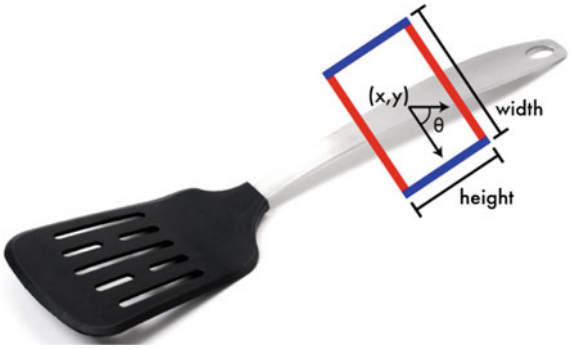


Fig. 15 Pin representation [19]. The *blue edges* are parallel to pinching fingers and width is the open width of the two fingers. The *blue lines* tell the size and orientation of the two pinching fingers. The *red lines* show the distance between thumbs and the opposed fingers

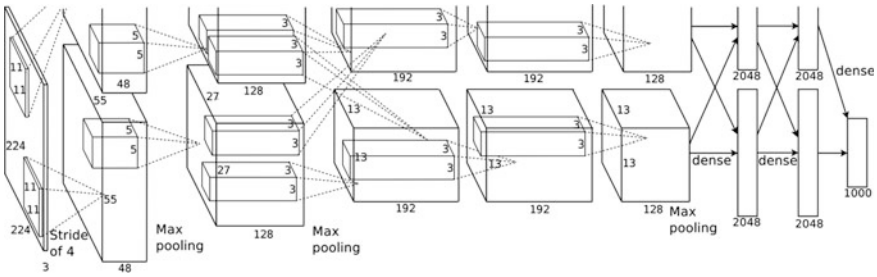


Fig. 16 AlexNet network architecture [27]

- The Jaccard index $\frac{|A \cap B|}{|A \cup B|}$ is greater than 0.25. A and B are ground truth rectangle and predicted rectangle
- The metric requires that a correct grasp rectangle should have large common areas with ground truth rectangle and their orientation rectangle should be small.

3.2 Model Architecture

Our convolutional neural network is a direct solution from color and depth images to object categories and grasping areas. We do not need object detection, and we predict proper grasping areas and object categories directly. Our network outputs a grasping rectangle which represents the proper grasping area and categories of objects on which rectangles are located. It is a combined convolutional neural network which could finish two tasks: object recognition and grasping area

detection. This idea is derived from [19]. The combination decreases the training and testing time of the convolutional neural network and makes it real time without sacrificing accuracy.

The first five layers derived from AlexNet extract common features for object classification and grasping area regression. AlexNet (Fig. 16), one well known and widely used convolutional neural network, is first proposed in [27]. AlexNet is designed for object recognition and gets very good scores in the ImageNet competition. We adopted the AlexNet because we could easy fine-tune our convolutional neural network and fine-tuning could greatly improve the accuracy of results and eliminate overfitting. The architecture can be found in Fig. 16. They are five convolutional layers and two connected hidden layers in AlexNet. We copy the first convolutional layers and add two connected hidden layers with 2048 neurons each. At the end of the network, there is a softmax loss layer [28] which outputs object categories and a regressor, a Euclidean loss layer which outputs predicted grasping area.

3.3 Data Preprocessing and Training

Only a few datasets are suitable for our work. We chose Cornell Grasping Dataset [18]. There are 885 images of 240 different objects. For each image, there are one color and one depth image and 5 or more labeled grasping rectangle. It is originally designed for grippers, but it is also suitable for our use. We divide the images into about 16 categories, such as "bottle," "shoes" and so on.

To make full use of depth information, we substituted the blue channel with the depth channel. Then, we took a center crop of 340 * 340 for each image and did data augmentation for them. We translated them with a random number (between 0 and 50) of pixels in x and y direction and rotated them up to 360°.

We used our convolutional neural network showed in Fig. 3. We fine-tuned our model with weight file from AlexNet. As the first five layers in our network are the same as that in AlexNet, so we could initialize the first five layers with AlexNet weights. Then we will freeze the first five layers and train the network with the Cornell grasping dataset.

3.4 Software Development

We first developed our software using Theano [29], a deep learning python package. We implemented full network architecture from the beginning, preprocessing our data, defining different layers, training and testing the network and visualize our result. However, the network performance was too slow for training, which took about one day for training once. We then implemented our whole

program with Caffe [30], a well-known deep learning platform. It provides lots of pre-trained neural network model which could be used for fine-tuning, and it is also much faster than Theano.

4 Nadine Hand Design

In this section, we identify the biomechanical features that affect the functionality of human hand from the aspects of the bones and the joints. Furthermore, we introduce the model of Nadine hand, which drastically reduces the degrees of freedom (DOF) of the real human hand.

4.1 Human Hand Features

Figure 17 displays the skeleton of a real human hand which is made of 27 pieces of bones. The five fingers are constituted by eight small wrist bones, five metacarpal bones (in palm) and 14 finger bones. The bones are connected by five distal interphalangeal (DIP) joints, four proximal interphalangeal (PIP) joints and five metacarpophalangeal (MCP) joints in hand. The thumb is a special case which has no DIP or PIP but an Interphalangeal (IP) joint, a trapezio-metacarpal (TM) joint and an MCP joint. Thus, the real human hand has 15 movable joints in total and each finger have 3 of them.

A human hand totally has 27 DOF in total. There are 5 DOF in the thumb, 4 DOF in each other finger and 6 DOF in the wrist [31]. The 6 DOF in the wrist have the property of "global motion" which refers that all the palm and fingers will move at the same time, while the other 21 DOF in hand has the property of "local motion." In most prior studies of the motion or the design of robotic hands, only the "local motion" of the hand is considered.

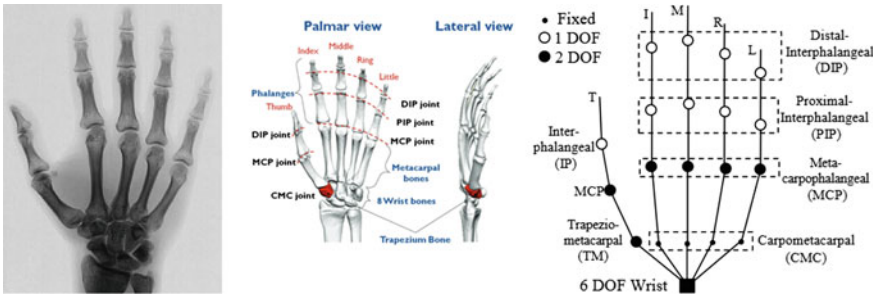


Fig. 17 Bones in human hand and DOF in hand [31]

4.2 Human Hand Constraints

Based on our observation, we find that the human hand and fingers have the following constraints:

- Fingers cannot bend backward without external force.
- The movement of the pinky finger will affect the ring finger or even the middle finger. Also, all the other four fingers are affected if the middle finger is bent.
- The DIP joint of each finger cannot be moved alone.

A formal representation of the constraints of human hands is proposed by Lin et al. [32], in which the constraints are divided into three types:

Type I has static constraints that limit the finger motion as a result of the hand anatomy. For example, the DIP joints cannot be moved more than 90°.

Type II has dynamic constraints that limit the joints in motion. For instance, when we bend a DIP joint, the corresponding PIP joint must be bent too.

Type III has the constraints that exist in certain human hand gestures. Sometimes they are not hard constraints, but most of the human beings follow it. For example, to make a fist, most people bend all the fingers together instead of bending fingers one by one.

4.3 Reduction of the Hand DOF

In this section, we propose several heuristics to reduce the DOF of the hand model. Firstly, we can reduce 5 DOF of the human hand by ignoring the subtle abduction/adduction motion of the fingers' MCP joints. So we reduce the DOF of the hand's local motion to 16. Moreover, only the thumb's TM joint has 2 DOF. It is difficult to simulate 2 DOF on one joint due to mechanical limitations. In Nadine's hand, a new joint is added to represent the TM joint of the thumb and simulate the abduction/adduction moving of the thumb. The final DOF is 16 and showed in Fig. 18.

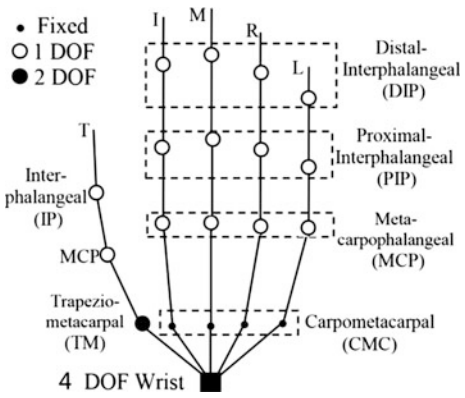


Fig. 18 Mechanical DOF in Nadine's hand

For the wrist part, with the help of Nadine robot's arm motion, Nadine can move hand up/down, side to side, forward/backward which contribute 3 DOF. However, as the design limitation of Nadine robot, it has only one actuator to simulate flexion/extension between the carpals and radius. So the abduction/adduction and supination/pronation motion from the wrist bone are omitted, and there are 4 DOF to represent the global motion of the hand. This omission will not affect the hand global motion much as arm will assist the hand to make the global motion. Together with the local motion, there are totally 20 DOF for Nadine's hand.

4.4 Simulate the Local Motion of the Human Hand/Finger

There are many ways to simulate the local motion of the hand. The difficult part is how to reproduce the DOF and the motion angles of each finger. It is also a tradeoff between performance, ability, and efficiency. One simple way is to use 16 actuators to control 16 DOF local motion. It will get direct control of each DOF separately. However, so far none of the commercial anthropomorphic hand chooses this way as the cost and design complexity are extremely high. The thing goes the same for Nadine's hand. So the first problem is to simulate the local motion with a minimal number of actuators (Fig. 19).

Nadine's hand has 6 actuators to control 16 joints and 6 DOF. The actuate way of Nadine's hand is similar like Dextrus [26] or Inmoov hand [24]. For the thumb part, the Inmoov hand's thumb has three joints but only have 1 DOF in flexion/extension. The Dextrus hand's thumb has 2 DOF in flexion/extension and abduction/adduction, but its flexion/extension has only two joints (IP and MCP). Nadine's hand combines the advantage of both hands. Its thumb has three joints and all three joints controlled by 1 DOF in flexion/extension. It has one more joint which split from the TM joint of the thumb so that it can make the

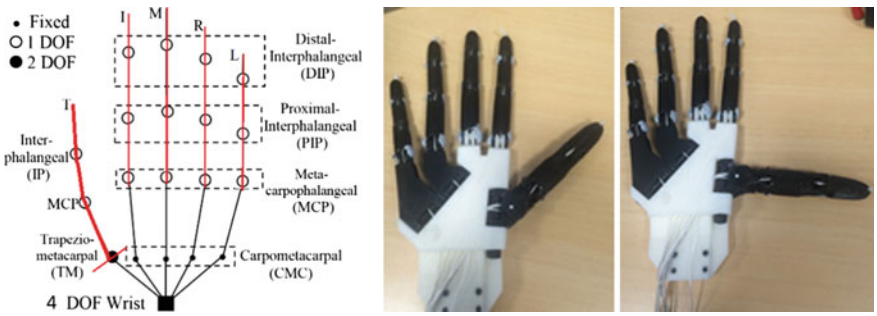


Fig. 19 Actuate DOF in Nadine's hand

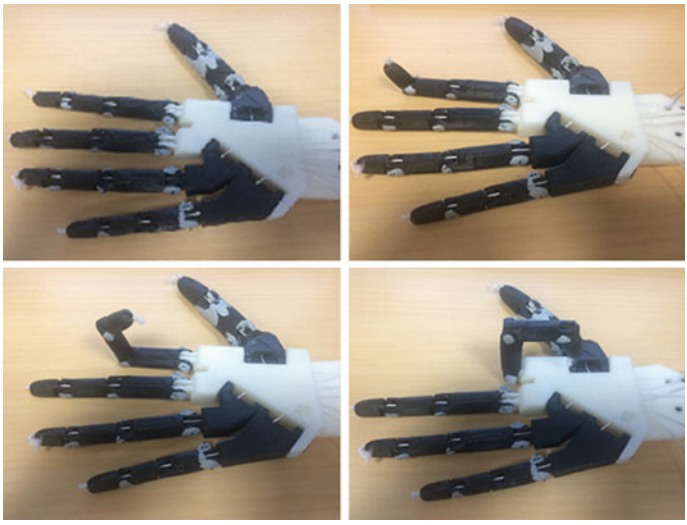


Fig. 20 Actuated fingers of Nadine's hand

abduction/adduction move. These joints can work under the artificial skin without any abnormal in the appearance of the hand.

In Fig. 20, we show the internal design of Nadine's hand and how it works. For each finger, it can be assembled by three pieces of 3D printable parts. The joint part can be linked together by a metal screw or a piece of solid wire like iron wire. A thread like a fish line or cotton wire goes through the inner part of the whole finger to control the movement. When pulling the control thread, the finger will move starting from the joint with minimal resistance force. Different weights in these three 3D printed parts represent the distal, middle and proximal phalanges. The distal part should move first as it is the lightest part and it has the minimal resistance force. Then if the pulling continues, the distal part will continue to move until it reaches the maximum moving point. After that, the middle phalange should start to move and stop at the limit angle. The proximal phalange should move last as it has the maximum resistance force. Nadine's hand only has one thread for each finger for flexion. As Nadine's hand has artificial skin, it provides the elastic force to make the finger extension after the release of the thread. It makes the design of the actuator simpler. Table 2 shows the Nadine hand's joint moving angles compare to other robotic hands.

As Nadine robot uses the air motor for its actuator, the weight of the new hand should be as light as possible to avoid making a heavy load to the joint of the wrist. It is the reason to choose plastic rather than metal. The technology of 3D printing gives an easy way to design and try errors. As most of the 3D printed hand has the same size as a human, it is impossible to put inside the artificial hand skin like

Table 2 Hand joint moving angle

Hand	Metacarpophalangeal joints (°)	Proximal interphalangeal joints (°)	Distal interphalangeal joints (°)	Thumb flexion (°)	Thumb circumduction
Nadine hand	0–90	0–110	0–90	0–90	0–90
Tack [25]	0–90	23–90	20	0–90	0–105
Dextrus [26]	0–90	0–90	0–90	0–90	0–120
I-Limb pulse [26]	0–90	0–90	20	0–60	0–95
Bebionic V2 [26]	0–90	0–90	20	–	0–68

"InMoov hand" [24] and "Tack hand" [25]. Based on the study of the state-of-art 3D-printed hand, the new created Nadine's hand is smaller and slimmer and does fit the artificial hand skin. With the help of 3D modeling software like 3DS Max or AutoCAD, the size of the hand can be easily changed to different scales, as well as the length of each bone. In the future, we could design software to automatically generate the 3D printed part of customized size hand by giving some basic size of the finger like each finger's and palm's dimensions.

For the weight, Nadine's hand is 200 g and much lighter than any of the existing hand. The three main reasons are first the 3D printed parts are not heavy as the metal parts in the commercial hand. The second reason is the servo motor of Nadine's hand used for the actuator (HITEC HS-5070MH) is only 12.7 g each. The third one is that Nadine's hand can use the external power. There is then no need to have a battery inside the hand.

5 Grasping Experiments

In this section, the characters of human grasping are listed and applied to virtual Nadine. After that, the grasp experiment is tested on Nadine robot with new robotic hand.

5.1 Human Grasps Motion Study

The human hand can move up to 50 m/s. However, most of daily motion's speed is within the range of 0.5–5.0 m/s. It will be abnormal if the hand move too slow or

too fast without a reasonable purpose. Nadine robot's forearm can at most move roughly 3 m/s under current air pressure. Moreover, it is easy to slow down the moving speed of Nadine by software method. We can also try to move Nadine's forearm and arm simultaneously to get the faster speed for moving the hand.

When humans want to move the hand to a certain position, he/she will use the joint of the shoulder, elbow or maybe wrist at the same time. Furthermore, the separate joints' motor will take almost the same time. For example, if a person wants to move his/her hand in a position which requires moving elbow up to 120° and shoulder up to 10°, he/she will not move them one by one, nor move them at the same speed. In that case, the shoulder will finish the movement much earlier than the elbow. The elbow and shoulder will cooperate with each other and try to have the same start and end time. Nadine's motion parameters are provided for each piece and each piece is performed in sequence.

When humans try to grasp something, their sight is most likely focused all the time on the target during the action. Nadine robot has its vision system. The first step of grasping action is to turn the head and eyes to locate the target. At this stage, the position of the target is fixed on the table. The first motion of Nadine is to low down the head and put the sight focus on the target and continue with the hand's motion.

Muscles and tendons support human to complete the grasp action. They are linked together, and they also constrain our motions. So a human cannot reach or hold some position too long as it will stretch or curl muscles or tendons too much. It is also needed to avoid the motion beyond the human anatomy limitation.

Humans usually show different face emotions when doing some action. As Nadine robot has seven movable points in the face, it adds to the human grasp motion to simulate some emotions.

5.2 Motion Design on Virtual Nadine

The Nadine robot has 27 air motors, and each of them can be set to different values to achieve various postures. Up to 30 posture frames can be set within one second. Then smooth actions can be produced if these postures are playing in sequence. To make the action look more human like, moving speed, motion continuity, motion simultaneity, sight focus and coordinate body movement are considered. The first action indicates the default position of Nadine robot with two hands in front of the body. The second action is for "eye on the target." Nadine robot will head down and look at the table. The third action is for "approach to the target," Nadine robot will forward her right hand higher and forward to the target object. The fourth action is for "touch the target," and Nadine robot will put her hands to the suitable position before grasp. Additional head, waist, and left arm/shoulder motions are

Fig. 21 Four intermediaries' actions of Virtual Nadine

added to third action to make the whole process more coordinated and human like. Some facial emotions are also added.

Four intermediaries' actions are showed in Fig. 21.

5.3 Grasp Experiments of Nadine

Nadine's hand is tested inside Nadine's artificial hand skin. A 3 × 3 × 5 CM sponge toy is used as grasp target. From Fig. 23, Nadine can move the hand toward the object, grasp it and move it to a new predefined location. These actions are similar to the virtual Nadine showed in Fig. 21. Four intermediaries' actions are showed in Fig. 22 for the real Nadine robot.

Fig. 22 Four intermediaries' actions of Nadine robot

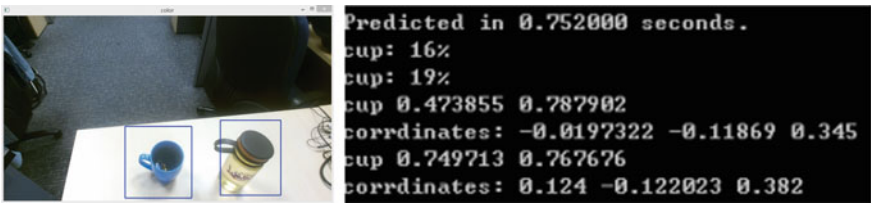


Fig. 23 Object detection's result

5.4 Grasp Experiments with Visual Information

Our experiments about grasping area detection are still on development. We are using visual information to guide our grasp. We first set up a Kinect v2 on a desktop

in front of Nadine. After our neural network predicts the object categories and 2d position, we will convert the 2d position to 3d position and orientation according to surface normal vectors obtained from depth images. The robot then will reach out her hands to the position and grasp the object.

The following is the object detection's results. It is real time and can achieve high accuracy.

After 15000 iterations for training our model and testing it on 300 images, we find that it can predict the location of the object, but the orientation and location of the grasping area are not accurate enough. These experiments are still under development.

6 Conclusion and Future Work

We presented our ongoing research on human-like grasp for social robots. It includes our state-of-the-art methods for the design of a robotic hand, the object recognition and grasping area detection. With a large number of experiments, we successfully validated our methods. The robot Nadine can achieve the grasping with a high accuracy as well as behave human-like when grasping. There is still a lot of work to do in this area. For example, how to make the grasp motion more human like rather than reaching directly to the destination with a constant speed? It is still a problem to solve.

In summary, our future work will mainly focus on the following areas:

1. Improve the robotic hand. Our current hand is not powerful and precise enough for wide and long-time use in common life.
2. Integrate visual information with a robotic hand. The robot should grasp any graspable objects once they come into Nadine's field of view.
3. Improve the way Nadine grasps. There are still a lot to do to make the grasping process completely humanlike.

Acknowledgements This research is supported by the BeingTogether Centre, a collaboration between Nanyang Technological University (NTU) Singapore and University of North Carolina (UNC) at Chapel Hill. The BeingTogether Centre is supported by the National Research Foundation, Prime Minister's Office, Singapore under its International Research Centres in Singapore Funding Initiative.

References

1. Uncanny valley, in Wikipedia. https://en.wikipedia.org/wiki/Uncanny_valley. Accessed 28 Nov 2016
2. https://funwithluka.files.wordpress.com/2015/03/img2885.jpg. Accessed 13 Nov 2016

3. Nadine social robot, in Wikipedia. https://en.wikipedia.org/wiki/Nadine_Social_Robot. Accessed 28 Nov 2016
4. Atlas (robot), in Wikipedia. https://en.wikipedia.org/wiki/Atlas_(robot). Accessed 30 Nov 2016
5. Y. Sakagami, R. Watanabe, C. Aoyama et al., The intelligent ASIMO: system overview and integration, in *IEEE/RSJ International Conference on Intelligent Robots and Systems, 2002,* vol. 3, (IEEE, 2002), pp. 2478-2483
6. G. Metta, G. Sandini, D. Vernon, L. Natale, F. Nori, The iCub humanoid robot: an open platform for research in embodied cognition, in *PerMIS'08 Proceedings of the 8th Workshop on Performance Metrics for Intelligent Systems,* pp. 50–56
7. Actroid, in Wikipedia. https://en.wikipedia.org/wiki/Actroid. Accessed 28 Nov 2016
8. T. Times, Meet Jiajia, china's new interactive robot. http://www.techtimes.com/articles/150827/20160416/meet-jiajia-chinas-new-interactive-robot.htm. Accessed 30 Nov 2016
9. Home, http://www.rickyma.hk/. Accessed 30 Nov 2016
10. R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, in *CVPR* (2014)
11. R. Girshick, Fast r-cnn, in *Proceedings of the IEEE International Conference on Computer Vision* (2015), pp. 1440–1448
12. S. Ren, K. He, R. Girshick, J. Sun, Faster r-cnn: towards real-time object detection with region proposal networks, in *Advances in Neural Information Processing Systems* (2015), pp. 91–99
13. J. Redmon, S. Divvala, R. Girshick, A. Farhadi, You only look once: unified, real-time object detection, in *CVPR* (2016)
14. W. Liu, D. Anguelov, D. Erhan et al., SSD: single shot multibox detector (2015). arXiv:1512.02325
15. A. Saxena, J. Driemeyer, A.Y. Ng, Robotic grasping of novel objects using vision. Int. J. Rob. Res. **27**(2), 157–173 (2008)
16. Q.V. Le, D. Kamm, A.F. Kara, A.Y. Ng, Learning to grasp objects with multiple contact points, in *IEEE International Conference on Robotics and Automation (ICRA)* (2010), pp. 5062–5069
17. Ian Lenz, Honglak Lee, Ashutosh Saxena, Deep learning for detecting robotic grasps. Int. J. Rob. Res. **34**(4–5), 705–724 (2015)
18. Y. Jiang, S. Moseson, A. Saxena, Efficient grasping from rgbd images: learning using a new rectangle representation, in *IEEE International Conference on Robotics and Automation (ICRA)* (2011), pp. 3304–3311
19. J. Redmon, A. Angelova, Real-time grasp detection using convolutional neural networks, in *IEEE International Conference on Robotics and Automation (ICRA)* (2015), pp. 1316–1322
20. D. Guo, T. Kong, F. Sun et al., Object discovery and grasp detection with a shared convolutional neural network, in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. (IEEE, 2016), pp. 2038–2043
21. Adioshun, Results matching, https://adioshun.gitbooks.io/learning-opencv-3-computer-vision-with-python/content/learning_python_opencv_ch07.html. Accessed 21 Dec 2016
22. G. Schreiber, A. Stemmer, R. Bischoff, The fast research interface for the kuka lightweight robot, in *IEEE Workshop on Innovative Robot Control Architectures for Demanding (Research) Applications How to Modify and Enhance Commercial Controllers (ICRA 2010)* (2010), pp. 15–21
23. S. Rueckhaus, MH24. http://www.motoman.com/industrial-robots/mh24 Accessed 28 Nov 2016
24. Inmoov project, http://inmoov.fr/. Accessed 1 Dec 2016
25. P. Slade et al., Tack: design an performance of an open-source, affordable, myoelectric prosthetic hand, in *2015 IEEE ICRA*
26. J.T. Belter et al., Mechanical design and performance specifications of anthropomorphic prosthetic hands: a review. J. Rehabil. Res. Dev. **50**(5) 2013

27. A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, in *Advances in Neural Information Processing Systems* (2012), pp. 1097–1105
28. Layer catalogue, http://caffe.berkeleyvision.org/tutorial/layers.html. Accessed 28 Dec 2016
29. Theano 0.8.2 documentation (2008), http://deeplearning.net/software/theano/. Accessed 21 Dec 2016
30. Deep learning framework (no date), http://caffe.berkeleyvision.org/. Accessed 21 Dec 2016
31. A.M.R Agur, M.J. Lee, *Grant's Atlas of Anatomy 10th ed* (1999)
32. J. Lin, Y. Wu, T.S. Huang, Modeling the constraints of human hand motion, in *Workshop on Human Motion, 2000*. Proceedings. (IEEE, 2000), pp. 121–126