DogBOT – An Interactive Robot Dog for Entertainment

Kuan-Ting Yu, Ping-Che Hsiao, Wei-Hao Mou, Yi-Shu Li
Department of Computer Science & Information Engineering, National Taiwan University

***Abstract*—This paper describes a robot dog. First, by using the Kinect sensor the user can use more natural 3D dynamic hand gestures to interact. Unlike previous work, either the user must wear some device to sense the human motion or the gestures are limited to 2D or static poses. Second, the DogBOT can follow the user with the laser range finder by utilizing an existing human tracking algorithm. Third, with a color camera on board, the DogBOT can chase ball smoothly by applying hue histogram back projection. The ball detection system can tolerate moderate illumination change. Lastly, we have designed the facial expression to let the DogBOT react in a more natural way.**

I. INTRODUCTION

With the rapid population growth in modern cities, people are becoming physically busy but psychologically lonely. A loyal companion, such as a pet, can be a good playmate for all ages. However, a real pet need extra care to feed it, to walk it, or to cope with its excrement, which are problems that people doesn't think of at the first moment. In contrast, a robot dog only needs charging, and its behaviors are controlled by engineers' design (i.e. won't spontaneously bark in the midnight).

There have been several successful robot pet products or prototypes. The ability to recognize faces and receive voice commands are the most common communication channel. However, few of them have utilized the gesture recognition, one of the most natural communication techniques between human and pets. The hand gesture recognition depends on a highly reliable hand tracking method, on which Kinect does a pretty good job.

In this project, we built a robot dog for entertainment which can recognize human's hand gesture. It can act like a pet dog which have facial expressions and voices and play with the master. Besides, the robot can actively detect and avoid obstacles, follow the master, and chase ball.

II. RELATED WORK

The most common extrioceptive sensor can be categorized into three kinds: tactile, audition and camera.

For tactile sensors, they could embed in the head, chin and back, so the robot can recognize stroking and patting. Like Paro[7], an advanced interactive robot developed by AIST, he feels being stroked and beaten by tactile sensor, or being held by the posture sensor. Sony's Aibo dog [8] has Head touch sensor and Chin Touch Sensor, it will immediately react when you stroke this touch sensor. Aibo is constantly learning from interacting with you.

Fig. 1. View of the robotic walker we designed.

For audition sensors, it is usually in the head, so it can detect sound and recognize the source of the sound. When Aibo hears something it will analyze the sound and recognize words. The direction of the sound came from is also perceived and Aibo will turn his head toward the source. Paro can also recognize the direction of voice and words such as its name, greetings, and praise with its audio sensor.

For camera, Aibo uses the Color vision camera to interact with you and its environment in a number of ways. Aibo can record movie clips, remember your face, take and send color pictures by email, or play programmed CD tracks by recognizing the CD cover. Within its scope of vision, Necoro[9], a very lifelike Robotic cat, can perceive the direction of moving objects. With the light sensor, Paro can recognize light and dark.

For hand recognition, Zhi Li *et al* proposed to recognize hand position by analyzing the histogram of depth image [11]. In contrast to traditional methods based on color information, hand segmentation is extract by depth image. It assumes that the hand position is the nearest object from range camera, and put the depth data into n bins of histogram. The distance of hand is then the nearest bin which contains enough number of points.

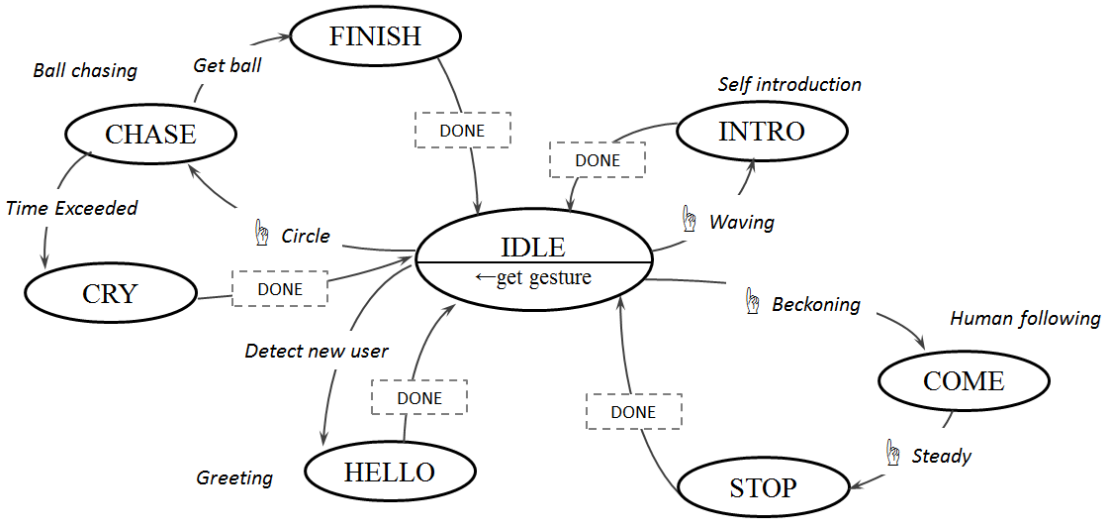


Fig. 3. DogBOT state transition

III. System Overview

A. Physical System Overview

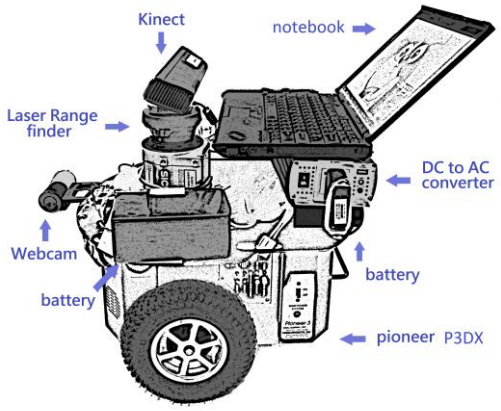


Fig. 2. Side view of DogBOT

B. Software System Overview

We basically construct the entire control system in a FSM design, which mainly consist of five actions. In IDLE state, it does nothing but detect the hand gesture by user; it greets and says hi to user in HELLO state; it introduce itself in INTRO state; it go chasing the ball in CHASE state; finally in COME state, it follows human until "Steady" gesture is detected. Fig. X shows the four gestures we have defined: *circle*, *waving*, *steady* and *beckoning*. We summarize the whole transition finite state machine in fig. 3.

About the transition between states, we communicate with each component by CMU Inter Process Communication (IPC) [10]. IPC provides efficient message passing between processes. When the process responsible for gesture recognition detects new gesture by user, it sends message to IPC server. The other components will receive the message then do the correspondent tasks. E.g., after detecting "waving" gesture, we send a message "INTRO" to IPC server. Another process starts the procedure of self-introduction after receiving "INTRO" message.

IV. Functionality

A. Gesture Recognition

We define how to determine whether a gesture is finished by means of constructing finite state machines (FSM). First, we construct the FSM for each gesture. By these FSMs, we could be aware of which of them is occurred. In the FSM, few steps are defined to finish. If user doesn't finish the hand gesture in time limit or the behavior violates the rule of gesture, it jumps to start state again. That is to say, user has to do the hand gesture from beginning.

For example, "waving" gesture consists of 4 states, including repeat twice of waving the hand from left to right and reverse. If the 4 task has been completed, then we determine that waving gesture is triggered by user.

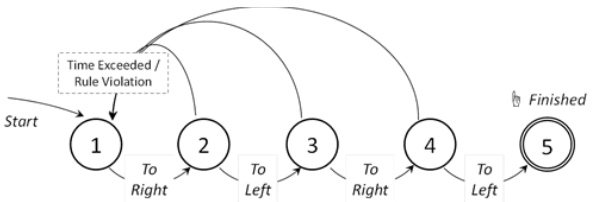


Fig. 4. Implementation example of "waving" gesture. Begin with start state; it is finished after going through all steps sequentially. It needs to do gesture again when it comes to time exceeded or rule violation.

B. *Human Following*

The researches on human's position detection and tracking using only LRF sensor are increasing [2][3][4]. Horiuchi *et al.* [5] have proposed a pedestrian tracking system based on sensing from LRF mounted on a mobile robot to identify potential moving pedestrian targets in the environment. Since our system is made to follow the nearest human, we filter out humans that showing too close or too far away from the dog robot.

Our final project build a hybrid approach to a Laser Range Finder (LRF) based human leg detection system that returns not only "true" or "false" type of answer but also a probability.

We first obtain the geometric information from measurements made by the laser range finder, and this set of measurement data is further decomposed into several sectors using segmentation. And then we apply a probabilistic model to compare these sectors with leg patterns to check if they belong to the set of human leg patterns or not. Moreover, we also use motion detector to check if these objects move or not as an enhancement of the detection.

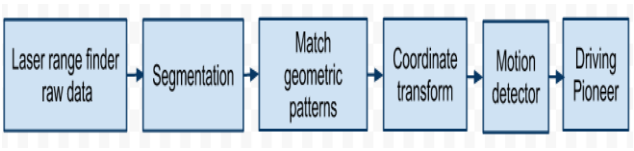


Fig. 5. Human following system flowchart.

Figure. 5. is the system flowchart of human following system in our Robotics final project. First, we mounted the laser range finder 40cm above ground level to get the distance of front of pioneer, the laser we use is SICK-LMS 100. The range of LMS 100 is 270 degrees and 20 meters, resolution is 0.5 degree, the sampling rate of LMS100 is $50Hz$. After a sequence of scanning points is collected by the laser range finder, we segment the laser points as shown in figure 6. Then, we do the geometric patterns matching to find out that whether the segment possible to be the human or not. In figure 7, if

$$d_{ab} = \sqrt{\left(\overline{OA}\right)^2 + \left(\overline{OB}\right)^2 - 2\overline{OA} \times \overline{OB} \times \cos\Delta\alpha}$$

is in the threshold we defined, where angle $\Delta\alpha$ is the constant value, 0.5°, we can define the segment as a human candidate.

Then we match the human candidates to geometric patterns to see if the segment like legs or not, the leg patterns is shown in figure 8, where a, b and c are following standard of normal human's leg size. And the forth step, we change the human position coordinate to pioneer's coordinate. The fifth step, we use a motion detector to examine if the detected object moves or not, we use this formula to do motion examination,

$$\sqrt{(x_i^t - x_i^{t-1})^2 + (y_i^t - y_i^{t-1})^2} \qquad (1)$$

Where x_i^t is an array of human position of X-axis at time t, y_i^t is an array of human position of Y-axis at time t, x_i^{t-1} is an array of human position of X-axis at time t-1, y_i^{t-1} is an array of human position of Y-axis at time t-1, if formula (1) is less than a threshold we defined, we will take it out of human candidate. After we change the position to the pioneer coordinate, we can driving the pioneer to follow human, since we only see the human candidate front of pioneer by 80 cm to 180 cm as the object of we want to follow, we can avoid pioneer to hit the human. Also during the walk, our obstacle avoiding system can work on the first priority, our obstacle avoiding is laser based, front of pioneer from 45° to 135°, the distance is 60cm.

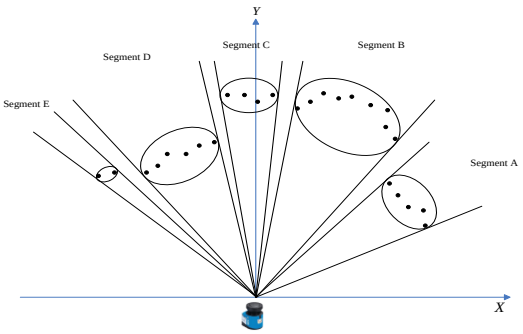


Fig. 6. Segments are composed of several laser points.

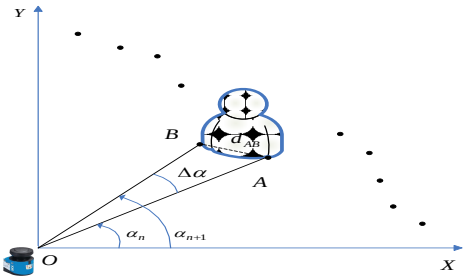


Fig. 7. Representation of the point-distance-based-method segmentation.

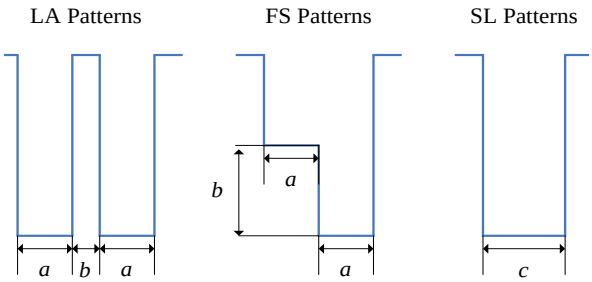


Fig. 8. Schematic representation of the leg patterns

C. *Ball Chasing*

Our ball chasing system includes the following four steps:

1) Calculate hue histogram of the ball
2) Back projection of the histogram on the current image
3) Thresholding and finding contour
4) Coordinate transform from image to base

To achieve better navigation ability (i.e. Nearness Diagram), precise target localization is required.

$$^{base}p = {}^{base}T_{cam}\ {}^{cam}T_{img}\ {}^{img}p\ ,$$

where ^{base}p and ^{img}p are the coordinate of the ball w.r.t. base and image respectively, $^{base}T_{cam}$ is the

transformation from camera's coordinate to pioneer p3dx's, and ${}^{cam}T_{img}$ is the transformation from image to camera.

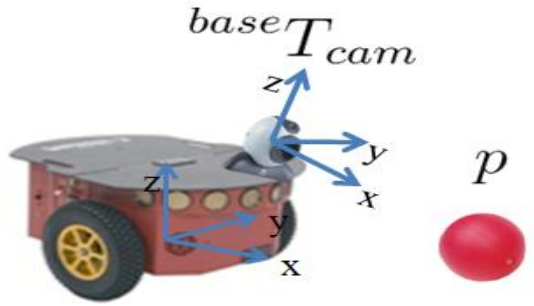


Fig. 8. Representation of the ball coordinates transformation from image to base.

We can calculate ${}^{base}T_{cam}$ by using camera calibration tool with the equation:

$$ {}^{base}T_{cam}\,{}^{cam}T_{board} = {}^{base}T_{board}, $$

where ${}^{cam}T_{board}$ can be measured as extrinsic parameter and ${}^{base}T_{board}$ can be measured by hand. Here *board* means the checkerboard used for camera calibration.

${}^{cam}T_{img}$, known as the inverse perspective transform, can be solved by combing the constraint of the camera matrix (2) and a general plane equation of the ground (3).

$$ \begin{bmatrix} {}^{img}p_x \\ {}^{img}p_y \\ 1 \end{bmatrix} = \begin{bmatrix} fc1 & 0 & cc1 \\ 0 & fc2 & cc2 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} {}^{cam}p_x / {}^{cam}p_z \\ {}^{cam}p_y / {}^{cam}p_z \\ 1 \end{bmatrix} \quad (2) $$

$$ ({}^{cam}p - {}^{cam}o_{board}) \cdot {}^{cam}\vec{n}_{board} = 0 \quad (3) $$

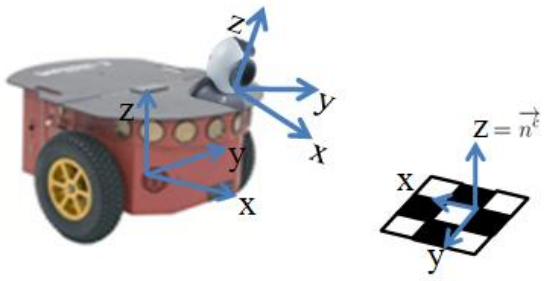
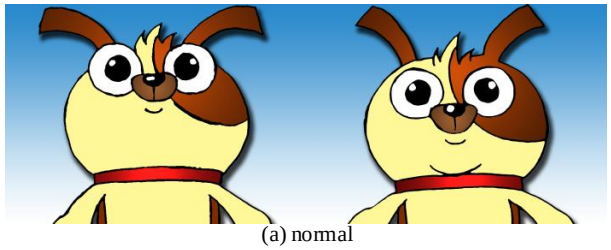
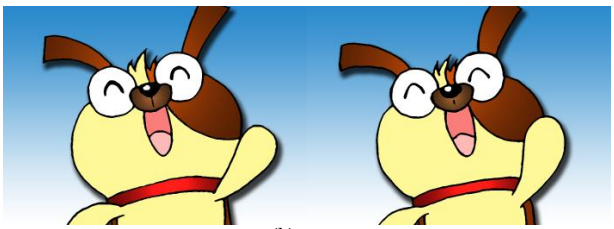


Fig. 9. Solve the transformation matrix with camera calibration tool and a checkerboard.

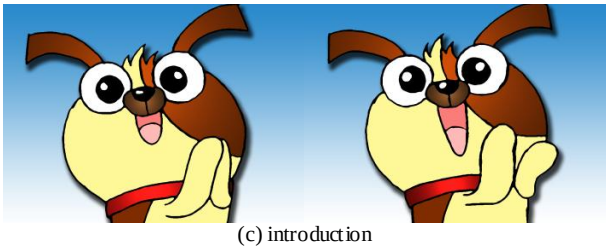
D. Facial Expression



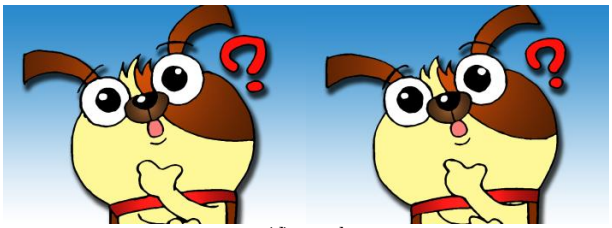
(a) normal



(b) wave

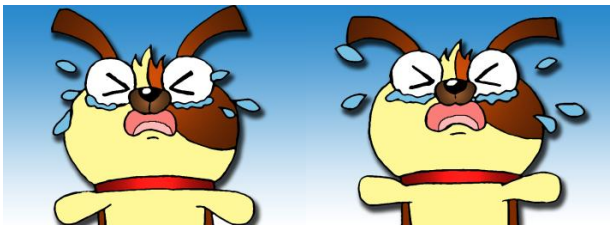


(c) introduction



(d) search

(e) happy



(f) cry

Fig. 10. Our dogbot's face

V. RESULTS

A. Stereo Vision

Stereo vision is a sensor which is used to produce depth image of environment. The principle behind stereo vision is to compare the disparities of two images, and then depth image is produced. However, we did some experiments on it, and the performance is not enough for us to extract hands in our application. Because of the principle of stereo vision, the depth image is rather incomplete when it comes to objects with less texture. Figure 11 shows ranging image comparison between stereo vision (left) and Kinect sensor.

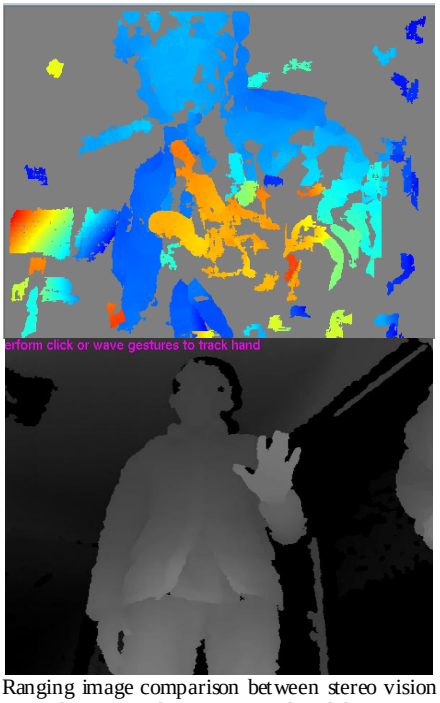


Fig. 11. Ranging image comparison between stereo vision (up) and Kinect sensor (bottom). The image produced by stereo vision is obviously rather broken.

B. Kinect Hand Tracking Capability

The distance can be detected by Kinect sensor ranges widely. It ranges from 0.5 meter to 4 meter above. And it costs 3 seconds to localize the hand of user in the first time use. And it just costs a second to re-catch the position of hand if the hand is going back to the range of Kinect again.

C. Human Following

Our human following system can demonstrated in the open space, with lots human crossing by, as it will only chase the nearest human, it won't follow the human who far away from it. Although it presented well in Demo the last course of Robotics, it may falsely detect some chairs as humans, so we have to put some boards to avoid this situation. The figure 12 is the demonstration of human following.

Fig.12 Human Following

D. Ball Chasing

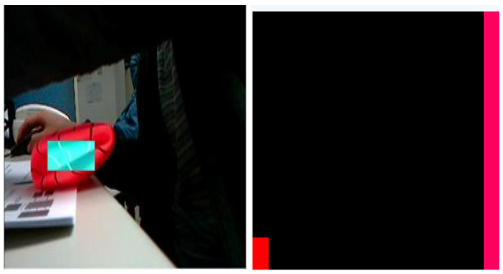


Fig.13 Hue histogram

Calculate the ${}^{base}T_{cam}$:

$$ {}^{base}T_{cam} = \begin{bmatrix} 0.043 & -0.26 & 0.98 & 130.97 \\ -0.998 & -0.057 & 0.032 & -9.31 \\ 0.049 & -0.97 & -0.22 & 242.42 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

To perform ${}^{cam}T_{img}$, the inverse perspective transform, we combine the (1) and (2) with the following measured parameters to obtain ${}^{cam}p$.

$fc = [5.27 \;\; 5.20]$, $cc = [3.21 \; 2.43]$

$ {}^{cam}o_{board} = [-0.9601 \;\; 134.67 \;\; 510.60] $

$ {}^{cam}\vec{n}_{board} = [0.0492 \;\; -0.9748 \;\; -0.2176] $

VI. DIVISION OF LABOR

Gesture recognition: Ping-Che Hsiao
Human following: Wei-Hao Mou
Ball chasing: Kuan-Ting Yu
Facial expression and voice: Yi-Shu Li

VII. CONCLUSION

We have implemented gesture recognition, the human following system on our Robotics final project, and successfully demonstrate it on the last course, although it may falsely detect some chairs as human, but after using some boards to mask the chairs out, the human following result is quite satisfied. The future work may be fusing webcam or Kinect to enhance the accuracy of human correctness.

REFERENCES

[1] C.-T. Chou *et al*, "Multi-robot Cooperation Based Human Tracking", in *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, 2011.

[2] E.A. Topp, H.I. Christensen, "Tracking for Following and Passing Persons," IEEE Int. Conf. on Intelligent Robots and Systems, pp. 2321-2327, 2005.

[3] A. Fod, A. Howard, M. J. Mataric, "Laser-Based People Tracking," Proceedings of IEEE Int. Conf. on Robotics and Automation, vol. 3, pp. 3024-2029, 2002.

[4] Z. Huijing, S. Ryosuke, I. Nobuaki, "A Novel System for Tracking Pedestrians Using Multiple Single-Row Laser-Range Scanners," IEEE Int. Trans. Systems, Man and Cybernetics, vol. 35, pp. 283-291, 2005.

[5] T. Horiuchi, S. Thompson, S. Kagami, Y. Ehara, "Pedestrian Tracking From a Mobile Robot Using a Laser Range Finder," IEEE Int. Conf. on Systems, Man and Cybernetics, pp. 931-936, 2007.

[6] http://www.vision.caltech.edu/bouguetj/calib_doc/

[7] http://www.parorobots.com/

[8] http://www.robotmatrix.org/Sonyaiborobot.htm

[9] http://www.megadroid.com/Robots/necoro.htm

[10] Inter Process Communication, http://www.cs.cmu.edu/~IPC/

[11] Zhi Li, Ray Jarvis, "Real time Hand Gesture Recognition using a Range Camera", Australasian Conference on Robotics and Automation (ACRA), Dec 2009.