SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

May 2019, Vol. 62 052205:1–052205:23
https://doi.org/10.1007/s11432-018-9681-3

Formation building and collision avoidance for a fleet of NAOs based on optical sensor with local positions and minimum communication

Xiaomin WANG[1,2,3*], Lorenzo BENOZZI[2,3,4], Benoit ZERR[2,3], Zexiao XIE[1], Hélène THOMAS[3] & Benoit CLEMENT[2,3]

[1]*Engineering college, Ocean University of China, Qingdao* 266100, *China;*
[2]*Lab-STICC, UMR CNRS* 6285, *Brest* 29806, *France;*
[3]*Ecole National Supérieure de Techniques Avancées (ENSTA) Bretagne, Brest* 29806, *France;*
[4]*Department of Industrial Engineering, University of Florence, Florence* 50121, *Italy*

Received 5 March 2018/Revised 5 September 2018/Accepted 31 October 2018/Published online 3 April 2019

Abstract Multi-robot system has become a research hotspot because of low demand on the sensors' accuracy, high reliability, and high efficiency. To put all the robots together, formation control is a crucial problem. In this paper, we propose a local position-based method to plan trajectories and build a pyramid pattern for a fleet of NAOs in the obstacle-free environment by refering to the position-based method and giving an $\mathcal{O}(n \log n)$ collision avoidance strategy inspired from one graph theory, where the local positions are estimated from optical sensors. To get the local positions, an integrated image processing method is developped. Firstly a mask-base is generated to store the features of NAOs, and a cross-correlation method is introduced to recognize the NAO. Subsequently, the distance and angle models are proposed to get the local information from a single image. Then, a visual compass is introduced to obtain the orientation of one NAO. After the local information exchange by the WiFi communication, a neighbor-check method is put foward to distinguish the homogeneous NAOs (all the NAOs look like the same). Further, a common frame is constructed as an artificial global frame, and straight non-intercrossing trajectories are planned according to the $\mathcal{O}(n \log n)$ collision avoidance strategy. At last, the performance of our proposed local position-based method is verified by the simulations with up to 15 robots and the indoor experiments with 3 NAOs in a real environment. The convergence of the method has been demonstrated in both obstacle-free and static obstacle environments.

Keywords formation building, a fleet of NAOs, optical sensor, visual compass, local positions

Citation Wang X M, Benozzi L, Zerr B, et al. Formation building and collision avoidance for a fleet of NAOs based on optical sensor with local positions and minimum communication. Sci China Inf Sci, 2019, 62(5): 052205, https://doi.org/10.1007/s11432-018-9681-3

1 Introduction

Robots have been widely used in many fields, such as search and rescue in the dangerous environments, tracking and arresting targets [1, 2], environment monitoring [3], and surveillance. Now, the studies about multi-robot systems are also increasing greatly because their cooperation can reduce the demand for sensors' accuracy, improve the reliability and can expand their working range to save time [4]. To make them cooperate and accomplish the missions well, a pattern formation needs to be considered and built to connect all the robots together. The existed formations mainly belong to four categories [5]:

* Corresponding author (email: wangxiaomin88@126.com)

virtual structure [6], behavior-based method [7], artificial potential field technique [8], and leader-follower structure [9].

To form the formations, formation control methods play essential roles, which include position-based, displacement-based and distance-based methods, according to the capacity of sensors and the topology of robots [10]. Position-based methods are the most popular and the easiest way to build one formation by updating the global positions online [11–13], while displacement-based methods need all the local robot frames to be aligned to the same orientation [14]. Distance-based methods have no special demand for each local robot frame, whereas it is a big challenge to propose the satisfying algorithms [15].

To get the positions, distances or other geometric information needed by the formation control methods, global or local sensors are needed, such as GPS, radio frequency (RF), laser, and vision system. In some cases, it is difficult to get the global positions in real-time, such as in some underground or underwater environment. As a result, building a pattern with local positions should be solved. An on-board vision system can be used to sense the local information for each robot. Except detecting the neighboring robots (neighbors) and getting their local positions, the vision system can also be used to avoid obstacles along the trajectories [16].

To cooperate with each other, communication should be considered. Usually, RF (or acoustic) communication is the main way to communicate on land (or in the water), but packet loss problem is still exiting. Now optical communication has been considered, such as [17], consequently the vision system can be used as part of communication to reduce the communication volume.

Now, NAO humanoid robot, as shown in Figure 1, is very popular, especially in the humanoid robot football game. And they are portable and affordable with acceptable quality and performance. They provide a platform to do the studies and to develop more functions [18]. In this paper, we focus on the problem of building a pattern with local positions got from the on-board cameras for the multi-NAO system in a 2D obstacle-free environment with minimum communication. The characteristics of the multi-NAO system include:

(1) The outline shape of NAO robot is complex with abundant features;

(2) All the NAOs are the same and share the same algorithms (homogeneous structure);

(3) The information acquisition (local position and orientation) depends on the forehead camera of each NAO robot.

Considering that the forehead camera looks forward, a leader-follower pattern, the planar pyramid pattern (pyramid pattern for short in the rest part), is given to put all the NAOs together [19], because the internal robots can consolidate the connections among the robots and reduce the communication length, which meets the short working range of the video cameras. To achieve the pyramid pattern, a local position-based control method is proposed to plan trajectories and build the pattern with minimum WiFi communication by refering to the position-based method and giving an $\mathcal{O}(n \log n)$ complexity collision avoidance strategy. To make the method feasible, the image processing is a key part to get the local positions of NAOs from a single image. Then an integrated image processing method, composed by the recognition method of NAO, the corresponding relative distance and angle models, a visual compass to get the global orientation, is developped. After sharing the local positions, a neighbor-check method is given to distinguish the homogeneous NAOs' IDs and to check the real neighbors. Further, a common frame is built to get the initial distribution refering the position-based method. Finally, the $\mathcal{O}(n \log n)$ complexity collision avoidance strategy inspired from a graph theory is put forward to optimize the pyramid distribution and plan the non-intercrossed straight trajectories to prevent the collision between NAOs.

The rest of this paper is organized as follows. In Section 2, the problem to be solved is described in detail. The recognition of NAOs and extraction of neighbors' information are presented in Section 3. The proposed local position-based control method considering the collision avoidance strategy with the local positions got from optical sensors is stated in Section 4. In Section 5, simulations with up to 15 robots and real experiments done with 3 NAOs are displayed. And the conclusion is given in Section 6.

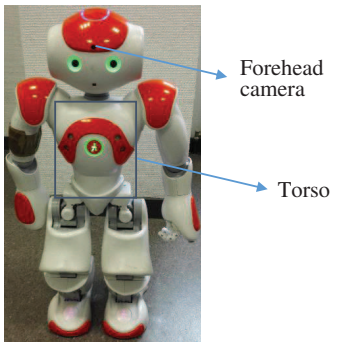


Figure 1 (Color online) One NAO humanoid robot.

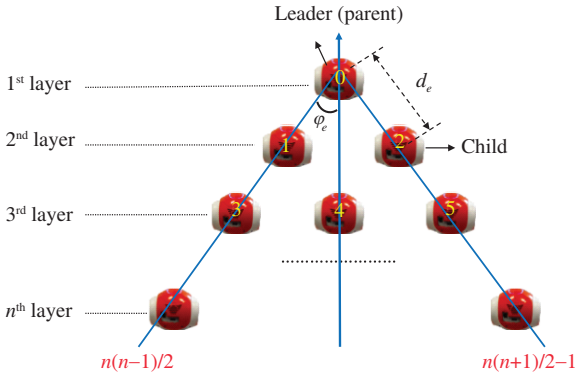


Figure 2 (Color online) Sketch of a planar pyramid pattern.

2 Problem description

In this paper, there are some assumptions: (1) each NAO moves in a 2D plane without obstacles; (2) there are enough features in the surroundings; (3) all the NAOs are located in a visual range, each NAO can see at least one neighbor, and they can be connected with each other through their neighbors. Otherwise, the NAOs will do the searching part again until they meet the condition (3).

The pattern to be built is a pyramid pattern shown in Figure 2, which is a leader-follower model, designed in [19] based on the second smallest eigenvalue of Laplacian matrix [20] and other literals [21,22]. Except the leader, each of the other NAOs (noted as a child) followers one NAO located in their front layer (noted as a parent) with the expected distance d_e and angle φ_e, respectively.

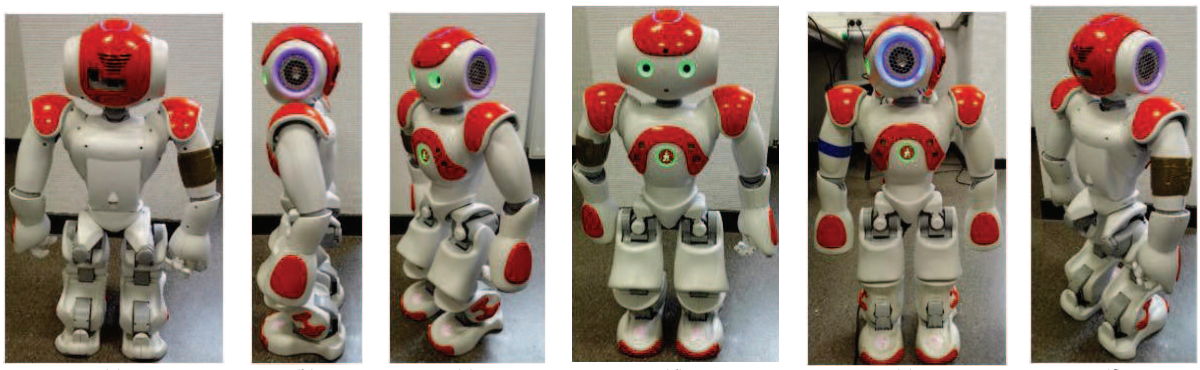
The purpose of this paper is to build a pyramid pattern with homogeneous NAOs depending on the local positions obtained by the visual system (the forehead camera) from a limited random origin distribution. In an obstacle-free environment, the position-based method can be referred. But without updating global positions online, the local position-based method should plan trajectories for all NAOs to arrive at their destinations as well as no collision among NAOs only depending on relative positions.

To make this kind of control method be valid, the NAOs should be recognized from the images. Subsequently, the relative positions among NAOs should be obtained, which include two elements: relative distance and relative angle. Both elements should be calculated immediately once a neighboring NAO is detected from a single image. To achieve this, a distance model and an angle model should be constructed beforehand. Moreover, as NAOs have no compasses but cameras, a visual-based method to measure NAOs' orientations should be introduced to provide feedback to make each NAO rotate to an acceptable orientation. Further, with a homogeneous structure, a distinction method should be given to distinguish each NAO before planning trajectories with our control method.

Therefore, in this paper, we need to solve these problems: (1) giving an image processing method to recognize NAOs from images, to get the relative distances and angles, and to estimate the NAOs' orientations; (2) proposing a kind of local position-based control method considering the collision avoidance strategy to build a pattern with the local positions got from the optical sensor without the assistance of global sensors, at the same time, without collision among NAOs.

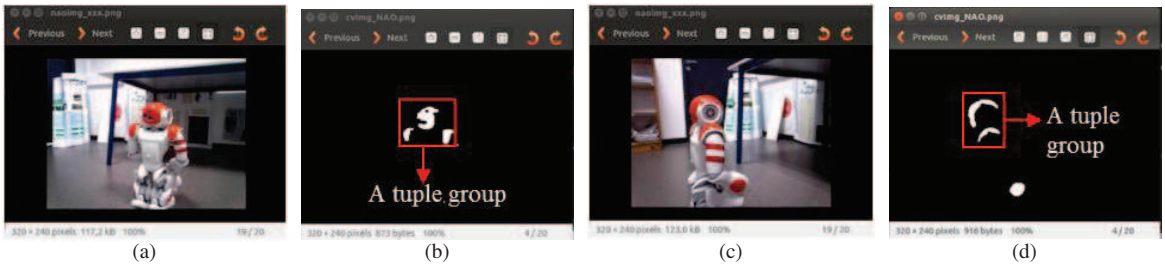
3 Image processing method for NAOs

In this paper, information acquisition mainly depends on the forehead camera of each NAO, as shown in Figure 1. Consequently, image processing becomes a significant part to compute the NAOs' local positions.



(a) (b) (c) (d) (e) (f)

Figure 3 (Color online) The examples of the same NAO's visual features captured with different relative orientations θ. (a) $\theta = 0$, $\varphi = 0$; (b) $\theta = 90$, $\varphi = 0$; (c) $\theta = 120$, $\varphi = 0$; (d) $\theta = 180$, $\varphi = 0$; (e) $\theta = 180$, $\varphi = 90$; (f) $\theta = 320$, $\varphi = 0$. θ is the orientation, the angle between the torso and the north directions, while φ is the heading, the angle of head yaw.



(a) (b) (c) (d)

Figure 4 (Color online) Image segmentation for recognizing an NAO: detection of the orange tuples. (a), (c) The origin color images; (b), (d) the corresponding segmented images.

3.1 Recognition of NAOs

Different visual features will be extracted when an NAO is located at different orientations θ (the angle between torso and north orientations) with different headings φ (the angle of head yaw), the visual features of the same NAO with different orientations are exampled in Figure 3. To recognize the NAOs correctly, a mask-base (M-base) is created to store these different features. Then a 2D cross-correlation is used to select the mask in M-base, which is the most similar to the visual features extracted from the current image [23].

3.1.1 *Generation of a mask*

As shown in Figure 3, the orange tuples of an NAO can be detected and distinguished from the suroundings by being segmented from the perceived image in the HSV color space with the orange range (2, 170, 50) (Hmin, Smin, Vmin) and (10, 255, 255) (Hmax, Smax, Vmax): except the orange regions are transformed into the white color, the other colors are transformed into the black color, as shown in Figure 4. Then the head and shoulders constitute one mask.

3.1.2 *Construction of the M-base*

Ideally, this M-base should consist of all the features of an NAO got from different distances d, different headings φ and different orientations θ. To save time, the masks in this M-base are got from the fixed distance, $d_s = 0.6$ m with the head yaw keeping to zero degree $\varphi = 0$, as the available distance interval is [0.5 m, 1.45 m] [23]. To get an accurate M-base, the masks are generated in a simulation software "V-rep", as shown in Figure 5. Then we generate masks according to the following criteria.

(1) Construct two NAOs (NAO-i and NAO-j) totally according to the parameters of the real NAOs [24]. Both NAOs maintain the same posture: "StandZero".

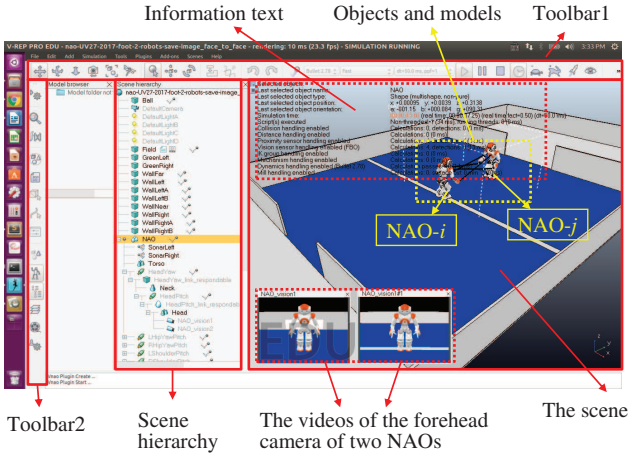


Figure 5 (Color online) The generation of the M-base in the software "V-rep".

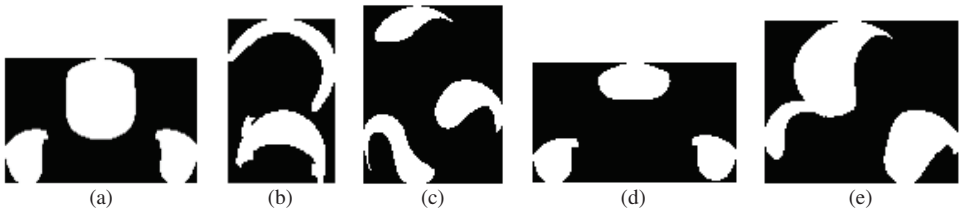


Figure 6 Some masks of the M-base corresponding to the (a)–(d), (f) of Figure 3. (a) $\theta = 0$; (b) $\theta = 90$; (c) $\theta = 120$; (d) $\theta = 180$; (e) $\theta = 320$.

(2) NAO-i, NAO-j both stand on the y axis of the "V-rep" frame with the position (0,0) and (0, 0.6 m) respectively.

(3) The orientation of NAO-i is fixed, always along the y axis of the "V-rep" frame ($\theta_i = 0°$), as shown in Figure 5; while the NAO-j turns around continuously with a step length: $\theta_{\text{stp}} = 5°$ (in Figure 5, the orientation of NAO-j is $\theta_j = 180°$).

(4) At each orientation of NAO-j, the NAO-i captures one image of NAO-j, and generates one mask with the features located in the center part of the image. After the NAO-j turns around, 72 ($\frac{360°}{5°}$) masks are generated, which make up the M-base. Some examples are shown in Figure 6.

3.1.3 *Recognition of NAO*

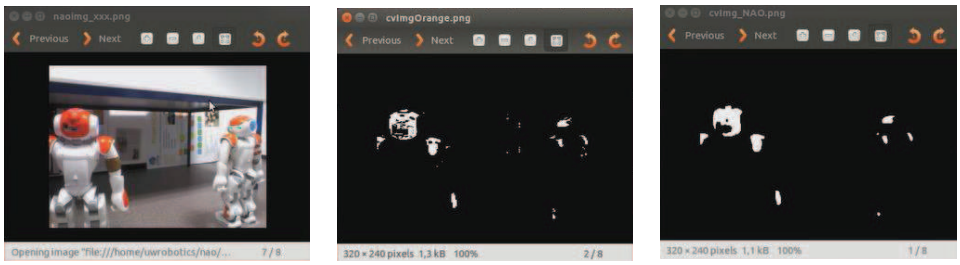
To recognize the pose and the distance to another NAO, the 2D discrete cross-correllation (Eq. (1)) is used to select the mask in M-base that is the most similar to the observed tuple.

$$(f * g)[x, y] = \sum_{m=0}^{w} \sum_{n=0}^{h} f[m, n] g[x + m, y + n]. \tag{1}$$

Here, $f[m, n]$ stands for one mask in the M-base, while $g[x, y]$ represents a tuple group extracted from one image. w and h are the width and height of the tuple group, respectively.

The larger the value of $(f * g)$ is, the higher the similarity between $f[m, n]$ and $g[x, y]$ will be. Then we define a confidence $\lambda = \sqrt{\frac{a}{b}}$ (the possibility to be an NAO) to judge whether $g[x, y]$ is an NAO or not. Here, $a = (f * g)[x, y]$, and $b = (f * f)[x, y]$.

However, usually, the images segmented by the orange color include more than just the NAOs' features because of other orange objects or light noises in each image, as shown in Figure 7(b). To eliminate the noises, the qualities of the images should be improved. In this work, the noises are filtered by one Gaussion filter combined with some morphological operations (closing and opening), and one improved image is shown in Figure 7(c). And like the Figure 7(c), there are usually more than 3 tuples in the images because that more than one NAOs somtimes are shown in the pictures and light noises sometimes



(a) (b) (c)

Figure 7 (Color online) The tuples extracted from one image. (a) One image got from the forehead camera; (b) the binary image after segmentation of image with orange color in the HSV space; (c) enhanced image with one Gaussian filter and the closing and opening operations of morphological operations.

cannot be eliminated totally for reserving the NAOs' features as much as possible. In this case, to find the real tuple group, every 3 tuples (or 2 tuples if in the "side" postures) will construct one group $g_j[x, y]$, and all the tuple groups in one image will assemble to a set $S = g_j[x, y]$ $(j \in 1, 2, \ldots, C_n^3(\text{or } C_n^2))$, where n is the number of tuples detected from one image.

Algorithm 1 determines which mask in the M-base corresponds to the visual features extracted from the current image. It consists an exhaustive search on every tuple group in the image, every pose and every distance. Each distance corresponds to a scale factor r_g on the masks computed at a fix distance of 0.6 m. The association of the mask in M-base and the tuple group $g_j[x, y]$ in the image showing the highest confidence $\lambda_{\max}$ is selected.

Algorithm 1 Recognition of NAOs with all the tuple groups in one image

for $g_j[x, y]$ in tuple groups **do**
  for mask in M-base **do**
    for scale (r_{g_j}) in range $[0.01, 1.5]$ with a step 0.01 **do**
      dim = size of mask $\times$ r_{g_j};
      if the size of dim is smaller than the size of the tuple group $g_j[x, y]$ **then**
        Break;
      end if
    end for
    mask_new = mask $\times$ dim, to change the size of mask to become smaller than the size of $g_j[x, y]$;
    λ_{g_j} is calculated by comparing the mask_new with the tuple group $g_j[x, y]$, and is stored in the list L_λ;
  end for
  $\lambda_{g_j} = \max\{L_\lambda\}$ is the evaluation of $g_j[x, y]$ and is stored in the list $L_{\lambda_{\max}}$;
end for
$\lambda_{\max} = \max\{L_{\lambda_{\max}}\}$ is the evaluation of an NAO existence or not in this image.

If $\lambda_{\max}$ is larger than a given threshold ($\lambda_{\text{th}} = 0.6$ in our study), one NAO is recognized, and its position in the image is that of the tuple group $g_j[x, y]$; meanwhile, the associated scale r_{g_j} will also be stored.

Remarks. As all the masks in the M-base are generated in the image center part, all the tuple groups are valid only when they are located in the center part of an image.

3.2 Extraction of neighbors' information

With the neighboring NAOs recognized from the images, the neighbors' information (relative positions) should be calculated, which includes the relative distances and angles, and the orientation of NAO itself. As there is only one on-board camera, to obtain the relative distances and angles, a distance model and an angle model should be put forward and calibrated. To get the orientation of NAO, a visual compass is introduced to make up for the lack of a compass.

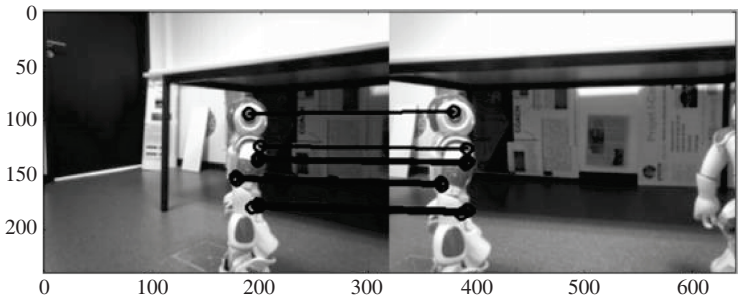


Figure 8 The corresponding feature points in two adjacent images. Left image stands for the current image, while the right one is the latest image.

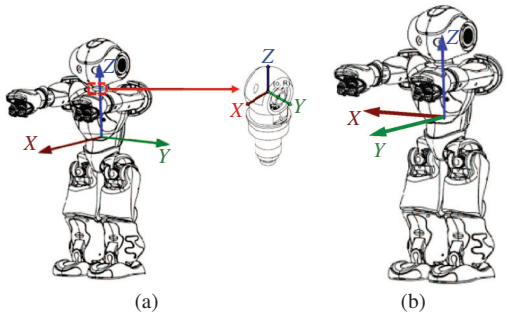


Figure 9 (Color online) The NAO frame (NAO-i), the neck frame (neck-i) (a) and the local frame (local-i) (b). In the NAO-i, the x axis is looking forward in the horizon plane, y axis is perpendicular to the x axis in the horizon plane, the origin point is the mass center, and z axis is perpendicular to the xy plane. The neck-i is parallel to the NAO frame. In the local-i, the origin point and its y axis coincide with the origin point and x axis of the NAO-i respectively, and its x axis is along the inverse direction of the y axis of the NAO-i.

3.2.1 *Identification of the orientation*

The visual compass uses the forehead camera to get the offset of orientation ($\Delta\theta$) at each rotation step by combining the comparative results of two adjacent images and the values measured by accelerometer and gyroscope [25]. To guarantee the validity of visual compass, there should be enough corresponding features in the two adjacent images. It means that the surroundings should include abundant features, and each rotation step cannot be too large. Figure 8 shows one comparison result of two adjacent images. Then the current orientation will become

$$\theta_c = \theta_l + \Delta\theta. \tag{2}$$

Here, θ_c and θ_l are the current and latest orientations of the rotating NAO relative to the initial orientation. It implies that the orientation is a relative angle, whose value depends on the initial orientation given.

3.2.2 *Establishment of the angle model*

Before establishing the angle model, the NAO frame (noted as NAO-i) and the neck frame (noted as neck-i) should be described first. As shown in Figure 9(a), the NAO-i is relatively fixed to its torso, which is changeable following the movement of the NAO. The axes of the neck-i are totally parallel to those of the NAO-i. The head yaw φ_{Nij} in the NAO-i will be equal to φ_{nij} in the neck-i.

Associated with each NAO-i, a local frame is constructed, shown in Figure 9(b), noted as the local-i. Like the NAO-i, the local-i is also relatively fixed with the NAO's torso.

The angle model expresses the relative angle between one neighbor NAO-j and NAO-i in the local-i, φ_{lij}. The heading φ_{nij} (the angle of head yaw between the neck joint and x axis of neck-i) can be read from the neck joint. Considering that only the tuple groups at the center part of one image will be used to recognize the NAO in our study, the heading of NAO-i, φ_{Nij}, can be seen as the relative angle between

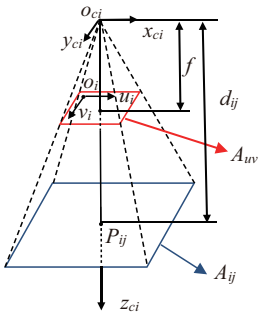


Figure 10 (Color online) Camera model. $o_iu_iv_i$ is the image frame, u axis and v axis are along the width and height of the image respectively. $o_{ci}x_{ci}y_{ci}z_{ci}$ is the camera frame of forehead camera of the NAO-i, where o_{ci} is the optic center. Its z axis is along the direction of optic axis and the z axis is parallel to the x axis of the NAO-i. x axis and y axis are parallel to the u axis and v axis, respectively. P_{ij} is the mass center of NAO-j in the NAO-i. f is the focus of forehead camera of NAO-i. A_{ij} and A_{uv} are the projections of one aspect of NAO-j in the plane $o_{ci}x_{ci}y_{ci}$ and in the CCD $o_iu_iv_i$. The distance between NAO-i and NAO-j is d_{ij}.

one neighbor NAO-j and NAO-i in the NAO-i according to the method in Subsection 3.1. Then the angle model can be expressed as

$$\varphi_{lij} = \varphi_{Nij} + 90^\circ. \tag{3}$$

3.2.3 *Establishment of the distance model*

As mentioned in Subsection 3.1.3, each extracted NAO is associated with a scale r_g to get the maximum similarity value. Figure 10 shows the camera model. Assuming that A_{ij} and A_{uv} are the projection's areas of NAO-j in the $o_{ci}x_{ci}y_{ci}$ (translated to the position of NAO-j, P_{ij}, the blue area shown in Figure 10) and in the $o_iu_iv_i$, we can get an ideal relationship between the distance d_{ij} and the area A_{uv} of NAO-j in the image, shown in (4), and Eq. (5) describes the relationship between d_{ij} and d_s.

$$\begin{cases} \dfrac{d_{ij}^2}{(f+\delta_{ij})^2} = \dfrac{A_{uv}}{A_{ij}} = \dfrac{r_{ij}^2 wh}{wh}, \\ \dfrac{d_s^2}{(f+\delta_s)^2} = \dfrac{A_{uv2}}{A_{ij}} = \dfrac{r_s^2 wh}{wh}, \end{cases} \tag{4}$$

$$\frac{d_{ij}}{d_s} = \frac{f+\delta_s}{f+\delta_{ij}} \cdot \frac{r_{ij}}{r_s} = \frac{f+\delta_s}{f+\delta_{ij}} \cdot \frac{1}{r_g}, \tag{5}$$

where δ_{ij} and δ_s are the uncertain factors because the $g_j[x, y]$ cannot exactly be at the center of one image. r_{ij} and r_s are the scales between the size of A_{uv}, A_{uv2} (the projection area in the image when $d_s = 0.6$ m) and A_{ij}, respectively. And $d_s = 0.6$ m is the distance for the generation of M-base. Then the distance d_{ij} is deduced to have an inverse proportion relationship with the scale r_g of NAO-j, expressed by (6) got from (5):

$$d_{ij} = \frac{a}{r_g} + b, \tag{6}$$

where a is fluctuated around 0.6, standing for the distance corresponding to the M-base, while b represents uncertain factors. a and b compose a coefficient pair. Because the NAO-j is recognized according to the different masks of the M-base, a and b associated to each mask are not the same. As a result, a distance base (D-base) is built to store these coefficient pairs, which will be presented in experiments.

4 Local position-based control method to build a pyramid pattern

Though the local positions got from the video cameras are the only information, considering the time expenditure, the popular position-based method can also be used without updating the trajectories in obstacle-free environment. Based on this idea, we propose our local position-based method by combining

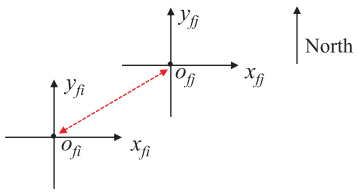


Figure 11 (Color online) The relationship between the origin points of the frames fixed-i ($o_{fi}x_{fi}y_{fi}$) and fixed-j ($o_{fj}x_{fj}y_{fj}$).

the position-based method and a collision avoidance strategy together. In this method, a common frame will be established with the local positions as an artificial global frame. Then the initial distribution will be translated into this frame. Subsequently, the pyramid pattern will be assigned and the non-intercrossing straight trajectories will be planned depending on the collision avoidance strategy. Finally, the NAOs will move to their destination with a simple kinematical control.

4.1 Establishment of the common frame

Inspired by the displacement-based method, a local fixed frame for each NAO is constructed, and all the fixed frames are aligned to be parallel with each other. The relative positions of neighbors will be computed in the fixed frames. Then the IDs of all the neighbors will be identified as all the NAOs look like the same. After that, a common frame will be established.

4.1.1 *Estimation of the relative positions*

Without the compass, NAOs cannot adjust themselves to the north direction automatically. Consequently, at the beginning, we manually put all the NAOs with the same initial orientation as a north direction but with random positions in a limited area. The original posture of local-i is regarded a fixed frame for each NAO (fixed-i), which indicates all the fixed frames are parallel. Then the relative positions can be calculated.

According to the distance and angle models, the relative positions of NAO-j in the local-i can be got by

$$\begin{cases} x_{lij} = d_{ij} \cdot \cos\varphi_{lij}, \\ y_{lij} = d_{ij} \cdot \sin\varphi_{lij}. \end{cases} \tag{7}$$

Then with the visual compass, the θ_{ij} can be got depending on (2). The neighbors' information of each NAO will be calculated in the fixed-i:

$$\begin{pmatrix} x_{ij} \\ y_{ij} \end{pmatrix} = \begin{pmatrix} \cos\theta_{ij} & -\sin\theta_{ij} \\ \sin\theta_{ij} & \cos\theta_{ij} \end{pmatrix} \begin{pmatrix} x_{lij} \\ y_{lij} \end{pmatrix}. \tag{8}$$

4.1.2 *Distinction of each neighbor's ID*

After all the NAOs get their neighbors' information, the communication occurs for exchanging the neighbors' information among NAOs by WiFi with the below format:

$$\begin{aligned} &\{\text{ID of NAO-}i\text{ itself: } i; \\ &\text{number of neighbors detected by the image processing step: } N_i; \\ &\text{relative positions of the detected neighbors: } (x_{ij}, y_{ij}),\ j \in [1, 2, \ldots, N_i]\}. \end{aligned}$$

As the multi-NAO system has a homogeneous structure, a neighbor-check method is given to distinguish their neighbors. Because all of the fixed frames are parallel, if NAO-i and NAO-j can find each other, ideally, the position values of $o_{fi}(x_{ji}, y_{ji})$ in the frame $o_{fj}x_{fj}y_{fj}$ and the position values of $o_{fj}(x_{ij}, y_{ij})$ in the frame $o_{fi}x_{fi}y_{fi}$ are opposite to each other, shown in Figure 11. The sum of their relative positions

in each frame should be equal to 0. However, taking into account the position errors, the sum of positions should be in a small range

$$\begin{cases} \varepsilon_{xij} = x_{ij} + x_{ji} < \varepsilon, \\ \varepsilon_{yij} = y_{ij} + y_{ji} < \varepsilon, \\ \varepsilon_{dij} = \sqrt{\varepsilon_{xij}^2 + \varepsilon_{yij}^2} < \varepsilon. \end{cases} \tag{9}$$

According to the given threshold ε, the IDs and the positions of neighbors can be determined. If there are more than one relative positions meeting (9), the one with minimum distance error $\varepsilon_{d_{\min}} = \min\{\varepsilon_{d_{ij}}\}$, $j \in 0, 1, \ldots, m$ (m is the number of neighbors of NAO-i whose positions meeting (9)) is the right neighbor.

As a result, if one neighbor's position of NAO-i and one neighbor's position of NAO-j can meet (9), and the error, $\varepsilon_{d_{ij}}$, is the minimum value $\varepsilon_{d_{\min}}$, we believe this neighbor of NAO-i is NAO-j, the neighbor of NAO-j is NAO-i equally. And NAO-i and NAO-j are noted as a neighbor pair. The distinction is done by going through all the neighbors of all the NAOs, as shown in Algorithm 2.

Algorithm 2 Identification of the IDs of neighbors with the neighbor-check method

```
for i in all the NAOs do
  for j in all the neighbors of NAO-i do
    for k in all the NAOs do
      if k! = i then
        for n in neighbors of NAO-k do
          if ε_xjn, ε_yjn, ε_djn meet (9) and ε_djn is the minimum value then
            ID of neighbor-j of NAO-i is k;
            ID of neighbor-n of NAO-k is i;
          end if
        end for
      end if
    end for
  end for
end for
```

Similar to Algorithm 2, we check the results of distinction by re-going through all the neighbors with their IDs and delete those neighbor pairs whose errors do not meet (9). The correct neighbors will be obtained.

4.1.3 *Establishment of the common frame*

After Subsection 4.1.2, each NAO knows the neighbors' information of all the NAOs (IDs, number of neighbors, and relative positions). As all the local frames are parallel, each NAO can build a common frame independently by the critera below.

(1) If each NAO finds at least one neighbors, the fixed-n with maximum neighbors is chosen as the common frame; otherwise, the NAOs run the second step.

(2) If at least one NAO has no neighbor, these NAOs will restart searching neighbors again, and repeat the first step.

Then the positions of the other NAOs can be translated into the common frame step by step. For example, NAO-i is one neighbor of NAO-n, and then the relative position of NAO-i in the fixed-n can be computed by (8), noted as (x_{ni}, y_{ni}). After fixed-n is chosen as the common frame $(o_c x_c y_c)$, the position of NAO-i in the common frame (x_{ci}, y_{ci}) equal to (x_{ni}, y_{ni}). Because there is only translation between every fixed frame and the common frame, the position of NAO-j (one neighbor of NAO-i) in the common frame will become

$$\begin{cases} x_{cj} = x_{ci} + x_{ij}, \\ y_{cj} = y_{ci} + y_{ij}. \end{cases} \tag{10}$$

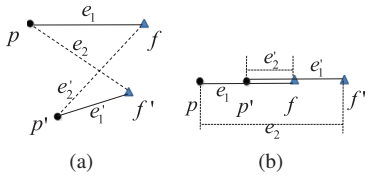


Figure 12 (Color online) The illustration of Lemma 1. If the points choose the short distances as the edges, the edges will be never intercrossing. (a) The general situation; (b) the situation with all the points are alligned. In this case, $d_{e_1} + d_{e'_1} = d_{e_2} + d_{e'_2}$, but if $\max(d_{e_1}, d_{e'_1}) < \max(d_{e_2}, d_{e'_2})$, we regard the e_1 and e'_1 are not intercrossing.

4.2 Collision avoidance strategy

With the initial distribution in the common frame, a collision avoidance strategy should be put forward to identify and optimize the pyramid distribution to be closed to the initial distribution, and plan the trajectories without collision.

Assuming that the net constituted by the NAOs' positions is one graph G, the points, p, p' are the positions of NAO-i and NAO-j in the initial distribution, while f, f' are the destinations of NAO-i and NAO-j in the pyramid pattern. Before giving the collision avoidance strategy, Lemma 1 of graph theory is presented.

Lemma 1 ([26]). Let $e_1 = (p, f)$ and $e'_1 = (p', f')$ be two edges in the graph G, $e_2 = (p, f')$ and $e'_2 = (p', f)$ be the other two edges in the graph G, and assume that p, p', f, and f' are distinct, as shown in Figure 12. Then e_1 and e'_1 never cross, although they can overlap ($e_1 \cap e'_1 ! = \emptyset$, shown in Figure 12(b)), if $d_{e_1} + d_{e'_1} < d_{e_2} + d_{e'_2}$ and $\max(d_{e_1}, d_{e'_1}) < \max(d_{e_2}, d_{e'_2})$. The proof for Lemma 1 has been given in [14].

After the optimized pyramid distribution is identified, the NAOs will never collide with each other, if the positions in both distributions match up according to Lemma 1, which can be expressed by

$$w(G) = \operatorname{argmin}\left(\sum_{p,f\in G} \operatorname{dis}(p, f)\right), \tag{11}$$

where $\operatorname{dis}(p, f)$ is the distance between the initial position p and the final postion f, and both p and f participate to constitute one and only one edge.

Considering that getting $w(G)$ by the mathematical method is complex, a collision avoidance strategy with $\mathcal{O}(n \log n)$ complexity is proposed to determine the optimized pyramid distribution and plan trajectories according to Lemma 1.

(1) Identification of the optimized pyramid distribution. As the pyramid pattern is symmetric, the origin distribution is divided into two groups as evenly as possible by the principal axis based on the principal component analysis (PCA). The principal axis is the vector associated with the larger eigenvalue of covariance matrix of positions (shown in Eq. (12)). Then a pyramid frame $o_p x_p y_p$ will be established, as the origin point $o_p(x_{op}, y_{op})$ is the average value of all the positions of NAOs in the $o_c x_c y_c$, the y axis is the principal axis (the pyramid orientation), and x axis is perpendicular to y axis, as shown in Figure 13. Subsequently, the pyramid distribution in the pyramid frame $o_p x_p y_p$ can be identified according to its definition after giving the leader position first [19].

$$\mathrm{Cov} = \begin{pmatrix} \dfrac{\sum_{i=i}^{N} x_{c_i}^2}{N} - x_{op}^2 & \dfrac{\sum_{i=i}^{N} x_{c_i} y_{c_i}}{N} - x_{op} y_{op} \\ \dfrac{\sum_{i=i}^{N} x_{c_i} y_{c_i}}{N} - x_{op} y_{op} & \dfrac{\sum_{i=i}^{N} y_{c_i}^2}{N} - y_{op}^2 \end{pmatrix}. \tag{12}$$

(2) Planning the non-intercrossing straight trajectories. The trajectories planning happens in the pyramid frame $o_p x_p y_p$, because both distributions can be translated into the pyramid frame according to the relationship between the pyramid frame and common frame, as shown in Eq. (13). And the key points are to separate the positions into groups evenly and generate the combined sub-groups meeting

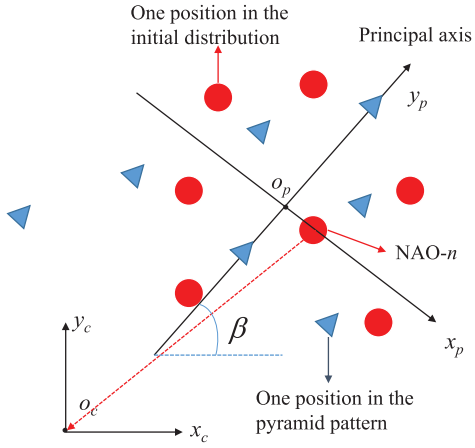


Figure 13 (Color online) Common frame ($o_cx_cy_c$) and pyramid frame ($o_px_py_p$). NAO-n is assumed to be the robot with the maximum number of neighbors. $o_cx_cy_c$ is the frame fixed-n. Here, $o_cx_cy_c$ presents the orientation but not the real position, while $o_px_py_p$ is located at the correct position. Its y axis o_py_p is the principal axis. Origin point o_p is the average value of all the positions in the $o_cx_cy_c$. β is the angle between the o_py_p and the o_cx_c.

the application conditions of Lemma 1. Then the positions in the initial and pyramid distributions can be matched up according to Lemma 1.

$$\begin{pmatrix} x_c \\ y_c \end{pmatrix} = \begin{pmatrix} \sin\beta & \cos\beta \\ -\cos\beta & \sin\beta \end{pmatrix} \begin{pmatrix} x_p \\ y_p \end{pmatrix} + \begin{pmatrix} x_{op} \\ y_{op} \end{pmatrix}, \tag{13}$$

where (x_p, y_p) and (x_c, y_c) are the positions in the pyramid frame and in the common frame respectively. β is the angle between the o_py_p and the o_cx_c.

(a) The generation of the combined sub-groups. To assign the positions in both distributions appropriately to constitute commbined sub-groups disjointedly as possible as they can, the positions should be sorted according to the ranking criteria. (i) The positions in both distributions should be separated into two groups equally. (ii) The positions in each group should be sorted according to the order of top to bottom and left to righ.

According to the ranking criteria (i), except the leader, the principal axis separates the initial and the pyramid distributions into two groups as evenly as possible respectively: L_I, R_I (the red ellipses shown in Figure 14(a)) and L_P, R_P (the blue ellipses shown in Figure 14(b)). The groups should satisfy the conditions $|N_{L_I} - N_{R_I}| \leqslant 1$, $|N_{L_P} - N_{R_P}| \leqslant 1$, and $N_{L_I} = N_{L_P}$, $N_{R_I} = N_{R_P}$ (N_{L_I}, N_{R_I}, N_{L_P} and N_{R_P} are the numbers of positions in the left and right groups of both distributions). If not, the positions near the principal axis in both distributions will be re-assigned to the left and right groups.

Then according to the the ranking criteria (ii), the positions in each group (L_I, R_I, L_P, R_P) will be sorted well. Subsequently, the sub-groups with one or two positions are assigned respectively (the red dashed ellipses such as R_{I1}, R_{I2}, and the blue dashed ellipses such as R_{P1}, R_{P2} in Figures 14(a) and (b)). In the next step, the sub-groups with the same order in both distributions combine together to make up one combined sub-group (such as R_{IP1} composed by R_{I1} and R_{P1}), as shown in Figure 14(c).

(b) Non-intercrossing straight trajectories planning. In each combined sub-group, the positions will be matched up according to Lemma 1. Then the matched-up positions in the pyramid distribution are the destinations of robots, and the position relationship[1)] among robots can be got. The edges of the match-up positions are non-intercrossing and straight, which are the planned trajectories including two elements: angle and distance, noted as (η_i, d_i) in the pyramid frame, shown in Figure 14(c). The details and the performance proof have been given in [19].

1) After the matching up part, each robot will know the order of its destination and orders of the others. The orders connected with IDs of robots are noted as position relationship. According to the position relationship, each robot can compute the expected local information between each robot and each of its neighbors.

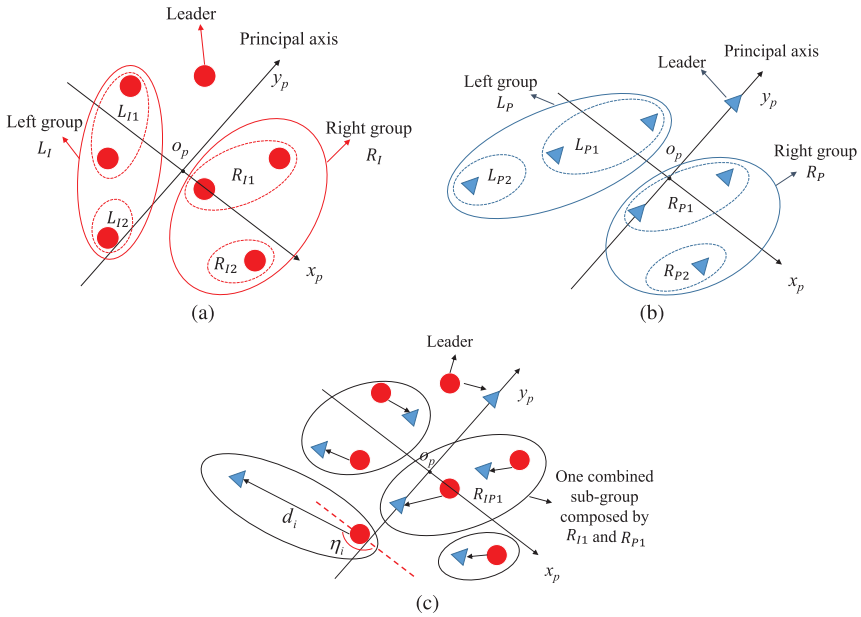


Figure 14 (Color online) The generation of straight and non-intercrossing trajectories in the pyramid frame. Except the leader (the NAO nearest to the first place in the pyramid pattern), the positions in the initial distribution and in the pyramid pattern are divided into the combined sub-groups with one or two positions respectively according to their subscriptions ordered by their coordinations (a), (b). Then in each combined sub-group, they match up depending on Lemma 1. The straight and non-intercrossing trajectories are planned, noted as (η_i, d_i) (c).

With the trajectories, each NAO moves to its destination along its trajectory (η_i, d_i). The movement of each NAO mainly depends on itself. To keep the NAOs moving straight as possible as they can, a simple kinematical control, which consists of a rotation control and a distance control, is given. In the rotation control, the PID control is introduced to correct its orientation η_i with the estimation of the visual compass as the feedback; while in the distance control, d_i is divided into several steps to be arrived at with an open-loop control, and at each step, the orientation is corrected with the rotation control. The explanation is given in Subsection 5.2.2.

5 Experiments

To evaluate the performances of this local position-based control method, the simulations and real experiments are both done. The simulations with up to 15 robots are performed in Matlab, while 3 NAOs are used to build a pyramid pattern (actually a triangle) in an indoor environment.

5.1 Simulation

In Matlab, a red point stands for a robot with radius as $r = 0.2$ m. The expected relative distance and the relative angle between each robot and its follower are set as $d_e = 2.0$ m and $\varphi_e = 30°$, respectively. In the simulation, we assume that a common frame has been constructed and all the initial distribution has been translated into the common frame. Then all the robots plan their trajectories with the proposed collision avoidance strategy and build the pyramid pattern in Matlab.

5.1.1 *The test in obstacle-free environment*

To verify the performance of the local position-based method to build the pyramids, up to 15 robots are used to build the pyramid with this method. To estimate the largest tolerable error range of the image processing step and the movement of NAOs to achieve a pyramid pattern, the relative distance errors

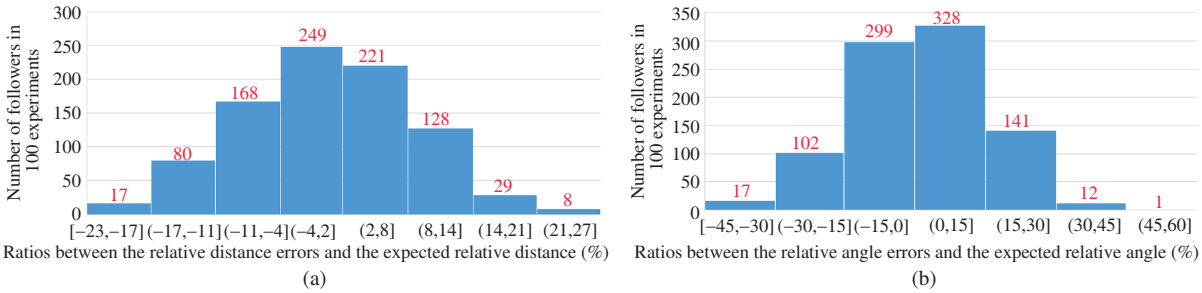


Figure 15 (Color online) The evaluation of the relative distance errors and relative angle errors between each follower and the followed robot at the convergent state of pyramid built by 10 robots with the relative distance noises $\delta_{d_{ij}} \in \pm 0.1 d_e$, and the planned trajectory noises $\delta_{d_i} \in \pm 0.1 d_e$ and $\delta_{\eta_i} \in \pm 0.05\theta_e$. (a) Relative distance errors; (b) relative angle errors.

$e_{d_{ij}} = d_{ij} - d_e$ and relative angle errors $e_{\varphi_{ij}} = \varphi_{ij} - \varphi_e$ between each follower (robot-i) and the followed robot (robot-j) at the convergent state are introduced.

The conditions of the pyramid building successfully are defined by the ratios between relative distance errors and the expected distance $r_{d_{ij}} = \frac{e_{d_{ij}}}{d_e}$ and the ratios between relative angle errors and the expected angle $r_{\varphi_{ij}} = \frac{e_{\varphi_{ij}}}{\varphi_e}$. Considering that the pyramid pattern will be continuously coordinated when moving forward to execute the missions, the errors ($r_{d_{ij}}$ and $r_{\varphi_{ij}}$) are allowed to be large but not larger than 30% and should be distributed evenly.

Then a series of noise values $\delta_{d_{ij}}$, δ_{d_i} and δ_{η_i} are added to the relative distances between neighbors and to the planned trajectories respectively to simulate the relative distance errors with neighbors generated by the image processing step and the errors of trajectories caused by the movement of NAOs. The simulations by 10 robots with the random initial distributions in a limited range are continuously repeated 100 times respectively with different noise values. The details are shown below.

(1) Only considering the relative distance errors caused by image processing step, the largest relative distance errors are set as $\delta_{d_{ij}} \in \pm 0.1 d_e$ according to the tests with NAOs in Subsection 5.2 ($\delta_{d_{ij}} \in \pm 0.1$ m)[2)]. The results show that when $\delta_{d_{ij}} \in \pm 0.1 d_e$, $r_{d_{ij}}$ is mainly within $\pm 13\%$, while the range of $r_{\varphi_{ij}}$ is mainly within $\pm 20\%$.

(2) Keeping the relative distance errors $\delta_{d_{ij}} \in \pm 0.1 d_e$, the planned trajectory noises δ_{d_i} and δ_{η_i} are introduced to the trajectories. Considering that the rotation and distance controls are closed and open control, respectively, the noise values are set as $\delta_{\eta_i} = 0.5\delta_{d_i}$. After simulation, we find that $\delta_{d_i} \in \pm 0.1 d_e$ and $\delta_{\eta_i} \in \pm 0.05\theta_e$ are the largest error range, with the results $r_{d_{ij}}$ and $r_{\varphi_{ij}}$ mainly within $\pm 20\%$ and $\pm 30\%$, respectively.

Figures 15(a) and (b) show the distributions of $r_{d_{ij}}$ and $r_{\varphi_{ij}}$ at the convergent state of pyramid built by 10 robots with the relative distance noises $\delta_{d_{ij}} \in \pm 0.1 d_e$, and the planned trajectory noises $\delta_{d_i} \in \pm 0.1 d_e$ and $\delta_{\eta_i} \in \pm 0.1\theta_e$. Figure 16 shows the trajectories of pyramids built with 6, 10, 15 robots respectively with noises $\delta_{d_{ij}} \in \pm 0.1 d_e$, $\delta_{d_i} \in \pm 0.1 d_e$ and $\delta_{\eta_i} \in \pm 0.05\theta_e$. The slight-crossing trajectories shown in Figures 16(d) and (f) are results from the combined sub-groups intercrossing with each other partially because of the crowded initial distribution.

5.1.2 *The test in static obstacle environment*

Though the local position-based control method is proposed mainly for the obstacle-free environment, it still can be used in the static obstacle environments as long as their relative positions can be got. The process of pyramid building in the environment with obstacles is shown below and one example is presented in Figure 17.

(1) At the beginning, the obstacles (the blue points with the same $r = 0.2$ m, shown in Figure 17(a)) are regarded as robots. The noises are added to the relative distances with neighbors and the planned

2) Relative angle errors are set unlimitedly, which means that the relative position errors $\delta_{x_{ij}}, \delta_{y_{ij}}$ are within an area with the radius r_δ equal to the maximum relative distance error $\max\{\delta d_{ij}\}$: $r_\delta = 0.1 d_e$.

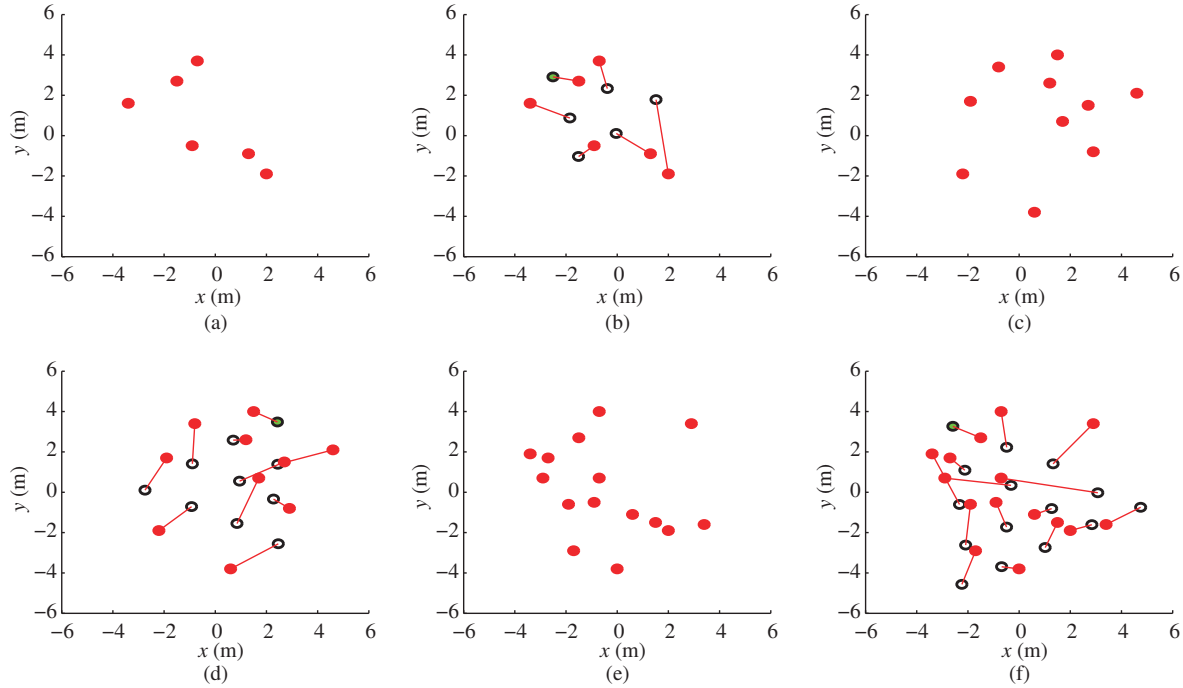


Figure 16 (Color online) The examples of building pyramids with 6, 10, 15 robots respectively with the relative distance noises $\delta_{d_{ij}} \in \pm 0.1 d_e$, and planned trajectory noises $\delta_{d_i} \in \pm 0.1 d_e$ and $\delta_{\eta_i} \in \pm 0.05\theta_e$. (a), (c), (e) are the initial distributions (the red points) with 6, 10, 15 robots respectively, while (b), (d), (f) are the corresponding trajectories (red lines) and the built pyramid patterns (the points with black edges). The only one green point with black edge is the leader in the pyramid patterns.

trajectories: $\delta_{d_{ij}} \in \pm 0.1 d_e$, $\delta_{d_i} \in \pm 0.1 d_e$ and $\delta_{\eta_i} \in \pm 0.05\theta_e$. Then the distribution of pyramid pattern considering the obstacles is optimized, and both the robots and obstacles participate in matching up with the positions of pyramid pattern according to the collision avoidance strategy.

(2) After the match-up process, the real robots move to their destinations in the pyramid pattern along their non-intercrossing straight trajectories without collision with the obtacles, as shown in Figure 17(b).

(3) Then the pyramid pattern and its surroundings are seperated into four areas: the internal, left, right and bottom of the pyramid pattern. After that, the obstacles are judged which one or more areas they belong to. Consequently the real robots will make a decision and choose a direction (left or right) to plan translation trajectories for themselves to move out to a new area without obstacles and should keep the pyramid pattern when they arrive at the new area, as shown in Figures 17(c) and (d).

(4) Then starting from the last layer, the robots correct the pyramid pattern by moving to the match-up positions of obstacles in the pyramid pattern layer by layer, as shown in Figures 17(e) and (f).

Remarks. To avoid collision with obstacles when robots move along their planned translation trajectories, the translation noises ($\delta_{d_{ti}}$, $\delta_{\eta_{ti}}$) cannot be too large. To achieve a pyramid pattern, the simulation results show that the ranges of translation noises are not large: $\delta_{d_{ti}} \in \pm 0.05 d_e$, $\delta_{\eta_{ti}} \in \pm 0.05\theta_e$. The reasons are twofold: (1) the error ranges of $\delta_{d_{ij}}$, δ_{d_i}, δ_{η_i} have been set as the largest ones; (2) the trajectories with big errors may lead to the collision with obstacles or to the broken of pyramid patterns.

The simulation implies that: (1) the local position-based control methods is verified to be valid when the number of robots increasing, even in the static obstacle environment, as long as the obstacles can be detected and their relative positions can be obtained; (2) the pyramid pattern can be achieved if the errors of relative distances meet $\delta_{d_{ij}} \in \pm 0.1 d_e$, and the errors of the planned trajectories are within $\delta_{d_i} \in \pm 0.1 d_e$ and $\delta_{\eta_i} \in \pm 0.05\theta_e$. The $r_{d_{ij}}$, $r_{\varphi_{ij}}$ mainly fluctuate within $\pm 20\%$ and $\pm 30\%$, respectively.

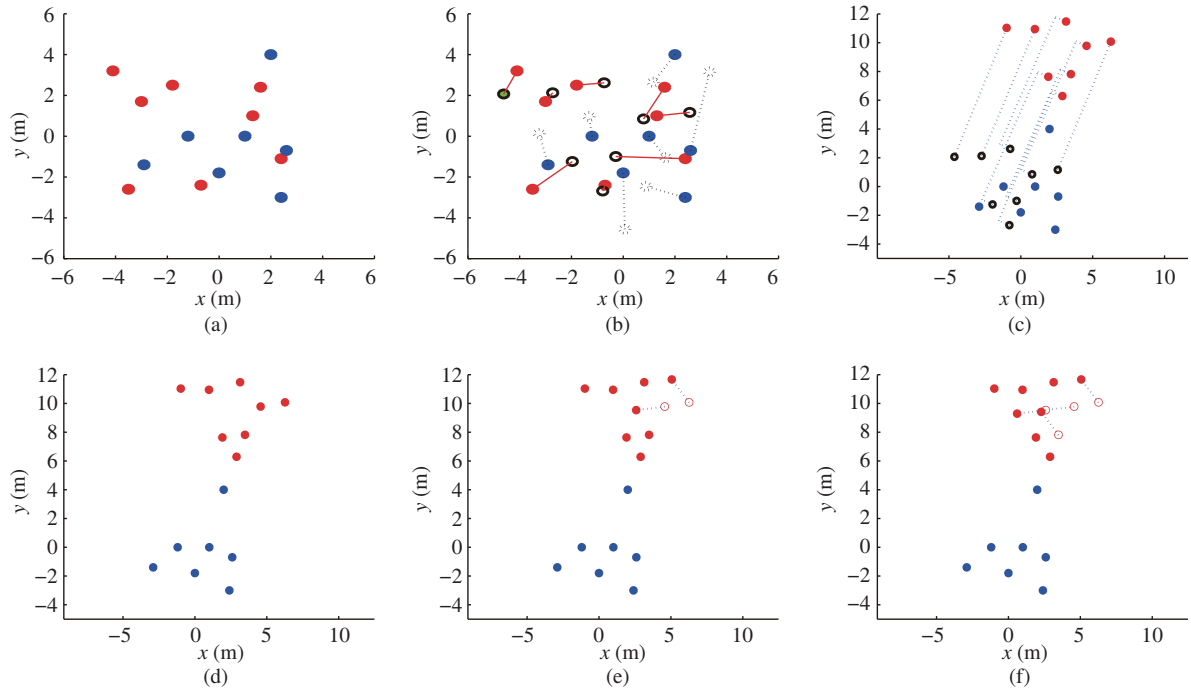


Figure 17 (Color online) One example of building a pyramid pattern in the static obstacle environment (8 robots and 7 obstacles) with relative distance noises $\delta_{d_{ij}} \in \pm 0.1 d_e$, planned trajectories noises $\delta_{d_i} \in \pm 0.1 d_e$, $\delta_{\eta_i} \in \pm 0.05\theta_e$, and translation trajectory noises $\delta_{d_{ti}} \in \pm 0.05 d_e$, $\delta_{\eta_{ti}} \in \pm 0.05\theta_e$. The red points are the robots, while the blue points are the obstacles. (a) The initial distribution of robots and obstacles; (b) the basic pyramid built by considering the obstacles as the robots, where the black edge circles and dotted black edge circles are the positions in the pyramid pattern corresponding to robots and obstacles respectively; (c) the basic pyramid moving out of the obstacle area along the planned obstacle-avoided translation trajectories; (d) the basic pyramid at the obstacle-free area moved out from (c); (e), (f) the process of correcting and building a final pyramid pattern by the followers moving to the positions in the pyramid pattern corresponding to the obstacles layer by layer.

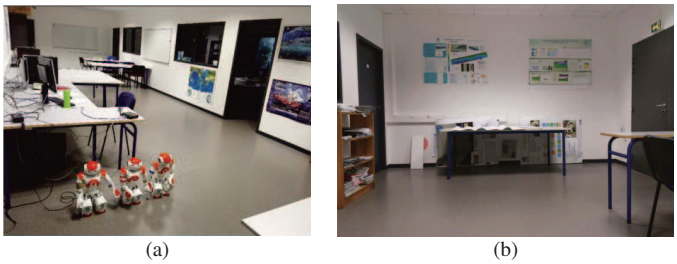


Figure 18 (Color online) Experiments environment with enough space and many features. (a) Experiment environment taken from front to back; (b) experiment environment taken from back to front.

5.2 Real experiments with three NAOs

The real experiments are done in a big lab with many features in surroundings, as shown in Figure 18. The three NAOs are totally the same with V4 version. They communicate with each other by WiFi through a center computer, with an Intel(R) Xeon(R) CPU (4 processors) at 3.1 GHz, to exchange neighbors' information. Before doing the experiments, we should calibrate the distance model and generate the D-base.

5.2.1 *Calibration of the distance model*

To get the accurate calibration results, we calibrate the distance model in the simulation software "V-rep" with the two NAOs built, as shown in Figure 5. The steps to sample the points is given:

(1) NAO-i, NAO-j both stand on the y axis of the "V-rep" frame. NAO-j has a fixed position (0,

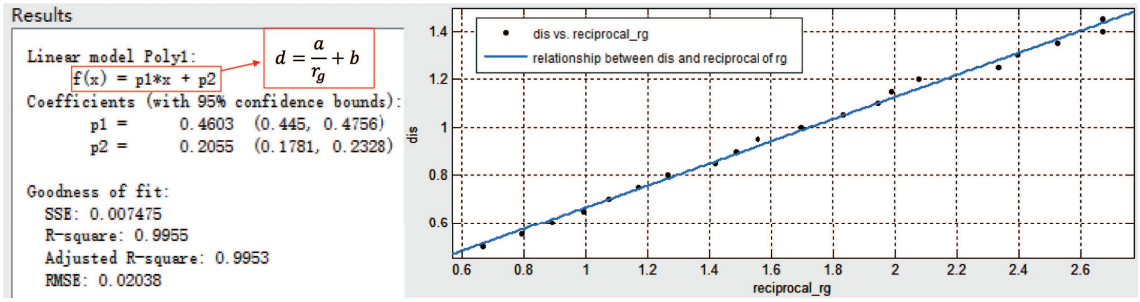


Figure 19 (Color online) An example of fitting curve of distance model with $\theta = 180$. Here, $f(x)$, x, p_1, p_2, represent d, $\frac{1}{r_g}$, a and b, respectively.

Table 1 Part of coefficient pairs of D-base with different orientations

a	b	θ
0.5044	0.1095	0
0.5167	0.09141	5
0.5146	0.09774	10
0.5068	0.1037	15
0.5032	0.1053	20
0.4908	0.1258	25
⋮	⋮	⋮

1.0 m), while NAO-i can change its position between (0, 0.5 m) and (0, −0.45 m) with a back step length 0.05 m along y axis to keep the distance in the range [0.5 m, 1.45 m], totally 20 positions.

(2) NAO-j is set to every orientation associated with the M-base, whereas, NAO-i keeps along the y axis of the "V-rep" frame.

(3) Owing to every orientation of NAO-j, the NAO-i recognizes the NAO-j and gets the scale r_g at each position mentioned in (1). Every distance d_{ij} and the reciprocal of the scale $\frac{1}{r_g}$ compose to a sample pair. Then 20 sample pairs are got at every orientation of NAO-j, called a sample set.

(4) After the NAO-j turns around, 72 sample sets associated 72 masks in the M-base will be generated. The sample pairs in each set are input into Matlab, a curve is fitted with the tool "cftool" (curve fitting toolbox), and one coefficient pair can be got. Finally, 72 coefficient pairs are got and they make up a D-base.

Figure 19 gives an example of fitted curve with $\theta = 180$. Table 1 presents part of coefficient pairs calibrated in the D-base.

5.2.2 *Building of the pyramid pattern*

Different from the simulation, some unexpected problems occur because of the environmental noises, such as the disturbance of objects with the same or similar color with NAOs, the visual features changed because of the light refection, and the joints and sensors' errors caused by the robot aging. To solve these problems, a series of tests are done, and associated solutions are given.

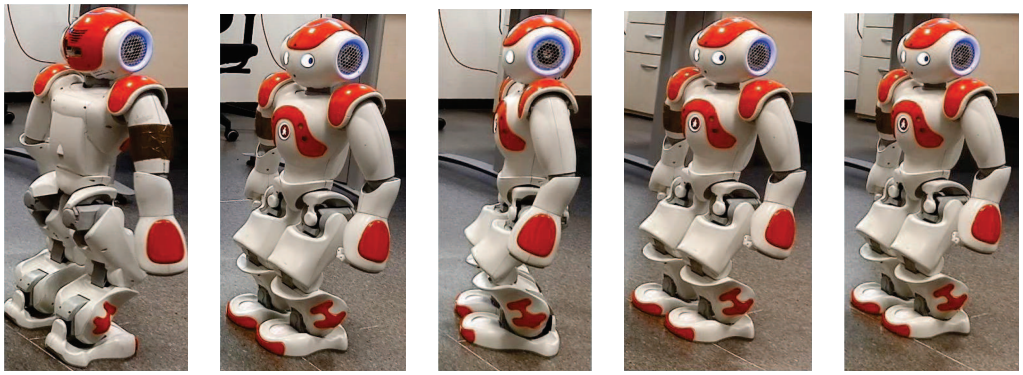
(i) Recognition of NAOs at "side" postures. Different from the other postures, every tuple group in the "side" postures is composed by two tuples. Consequently, we should give a way to distinct the "side" postures and recognize the NAOs at "side" postures, because the "side" postures have fewer features than the other postures.

In Figure 6(c), the width of the tuple group in "side" postures is smaller than those in other postures. As a result, the ratio $r_w = \frac{w_f}{w_i}$ is used to judge the postures, in which, w_f and w_i are the widths of the visual features and the image respectively. In our experiments, if $r_w < 0.225$, we believe the NAO stands sideways.

Because the "side" posture only contains two tuples, sometimes it fails to recognize the NAOs. The

(a) (b)

Figure 20 Two masks with "side" posture got in the real environment under two different kinds of lighting. The posture parameters are: the heading $\varphi = 90°$, the distance $d_s = 0.6$ m, and the orientation $\theta = 270°$.



(a) (b) (c) (d) (e)

Figure 21 (Color online) Comparison of the rotation behavior performed by the NAO itself and by the vision compass. (a) The initial position; (b) and (c) are good and bad rotation results by the NAO itself; (d) and (e) are the rotation results by the visual compass.

reason of failure is that the M-base is generated with $\varphi = 0°$ in an ideal simulation environment without any noises. However, in the real environment, without the influence from the noises, there is a great difference between the features with $\varphi = 0°$ and those with $\varphi = 90°$ in the "side" postures. Then we provide a supplementary "side" M-base got in the real environment with the light noise, which includes 2 masks with the same posture but two different kinds of light noises, as shown in Figure 20. The posture parameters are: the heading $\varphi = 90°$, the distance $d_s = 0.6$ m, and the orientation $\theta = 270°$.

(ii) Evaluation of the visual compass. To evaluate the performance of the visual compass, the behavior of turning around 180° is performed repeatedly by the NAO itself, and by the visual compass providing a feedback at a given position (a line written by a chalk on the ground), as shown in Figure 21(a). The results are also shown in Figure 21. We find that: (1) the bad rotation results are mainly caused by the sways of NAOs; (2) the results with the feedback of visual compass are more accurate than the results performed by the NAO itself, especially when the sways of NAOs have little influence to the rotation. But the visual compass also has some shortcomings, as shown in Figure 22. (1) When there are few features in one image, the orientation offset will be wrong because of the lack of corresponding features, as shown in Figure 22(a). (2) When there are similar features in the adjacent images, such as the corners of a table, wrong results will occur, as shown in Figure 22(b). (3) When one step length is too large, similar to (1), no features will be matched up, which leads to the false, such as Figure 22(c) with one step length 40°.

(iii) Building the pyramid pattern. To build the pyramid pattern by NAOs with the proposed integrated image processing method and the local position-based method, the process includes the following steps.

(1) The origin orientations of all the NAOs are set to be the same manually to keep all the fixed frames being parallel, because of the lack of compass.

(2) Each NAO searches its neighbors around itself independently and gets neighbors' information (the relative positions) with the proposed integrated image processing method.

(3) All of NAOs communicate with each other by the WiFi to exchange their neighbors' information,

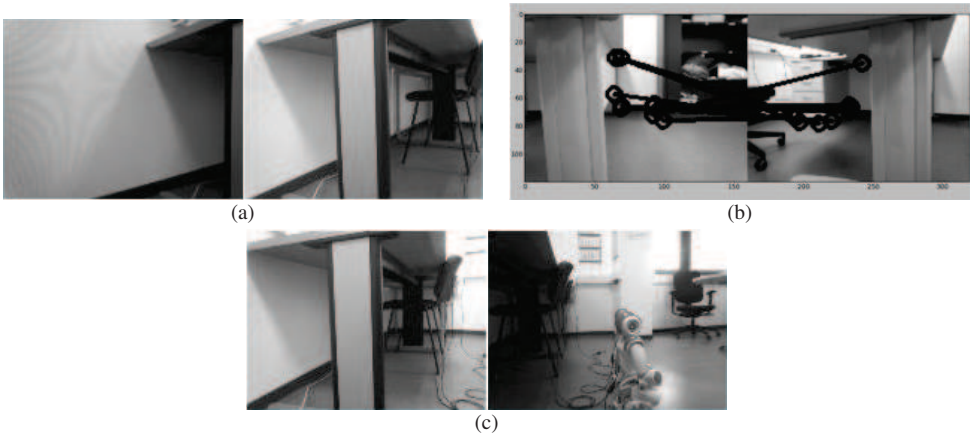


Figure 22 The situations of leading to wrong estimations of orientations by visual compass. (a) Few features in one or two images; (b) similar features in two images; (c) large rotation step length (40°).

Table 2 The moving distances and associated orientation errors

Distance (m)	e_{orien} (radian)	Distance (m)	e_{orien} (radian)
0.2	−0.0211	0.5	0.1698
0.2	0.0957	0.7	0.5292*
0.3	0.3734*	0.7	0.2274
0.3	−0.0379	0.8	0.334
0.4	0.0125	0.8	0.4465
0.4	−0.1579	1.0	0.4493
0.5	0.1374	1.0	0.4062

* The errors are large because of the heave sway of the NAO when it moves.

after all the NAOs finish the step (2).

(4) The NAOs distinguish the IDs of their neighbors according to the neighbor-check method by themselves.

(5) The NAO with the maximum amount of neighbors is chosen as the center, and its fixed frame is considered as the common frame. Then the positions of the other NAOs can be translated into this common frame.

(6) The proposed collision avoidance strategy is used to design an optimized pyramid pattern closed to the initial distribution. The strategy matches up the positions in both the initial and the pyramid distributions in the pyramid frame and generate the non-intercrossing straight trajectories (η_i, d_i).

(7) Finally, with the kinematical control, the leader goes to its destination firstly, and then the followers move along their trajectories and arrive at their destinations. After turning back to the pyramid orientation, the pyramid building is completed.

Remarks. Because the NAOs share the same algorithms, (4)–(7) steps are done in each NAO independently.

In this process, a simple kinematical control is used to correct the movement of NAOs as mentioned in Subsection 4.2, which includes rotation control and distance control.

(1) Rotation control. Considering the properties of visual compass got in Subsection 5.2.2, the step length cannot be larger than 40°. To get enough corresponding points, the step length is set as $\theta_{\text{stp}} = \min\{15°, \theta - \hat{\theta}\}$ (θ and $\hat{\theta}$ are the expected rotation angle and completed rotation angle computed by the visual compass respectively), and the PID coefficients are set as $k_P = 1.0$, $k_I = 0$, $k_D = 0.2$ until the error e_θ ($e_\theta = |\theta - \hat{\theta}|$) is within an allowed error.

(2) Distance control. Because the feedback of the inertial unit is accurate, the open-loop control is used to adjust the movement distance. After a series of distance tests (shown in Table 2), it implies that only running less than 20 cm per step can keep the movement trajectory in a straight line. The distance accuracy can be accepted.

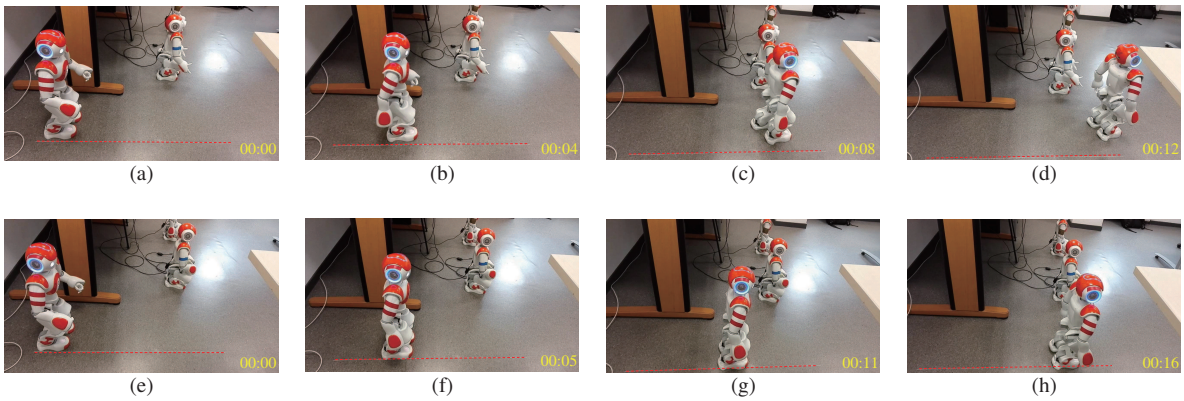


Figure 23 (Color online) The results of NAOs moving 0.8 m along a line (a chalk line, rewritten with red dashed line). (a)–(d) The movement results performed by the NAO itself; (e)–(h) the movement results performed by the distance control. (a) and (e) are the initial postures of both tests; (b) is the posture of moving forward 20 cm; (c) and (d) show the deviation of the movement by the NAO itself; (f) is the posture of moving forward 40 cm after the second step with $d_{\text{stp}} = 20$ cm; (g) presents the deviation of the orientation even moving forward 20 cm; (h) is the correction result of (g) by using the orientation feedback of distance control at the end of this step.

As a result, the step length is set as $d_{\text{stp}} = 20$ cm, and the rotation control is used to correct the orientation (but $k_D = 0$) at each step until the NAOs arrive at the destinations. Figure 23 presents the results of one NAO running 0.8 m along a line by itself (Figures 23(a)–(d)) or by the distance control with $d_{\text{stp}} = 20$ cm (Figures 23(e)–(f)). Generally, with a step $d_{\text{stp}} = 20$ cm, the NAO will move straight, as shown in Figures 23(b) and (f). Sometimes, it will generate an orientation deviation, as shown in Figure 23(g). In this case, the rotation control will be used to correct the orientation, presented in Figure 23(h).

With the preparation work, 3 NAOs build the pyramid pattern according to the trajectories planned by the proposed integrated image processing method and local position-based method with the expected relative distance and relative angle: $d_e = 0.8$ m and $\varphi_e = 30°$. When searching the neighbors, the forehead camera of each NAO needs to scan the surroundings around. As the range of head yaw is $[-119.5°, 119.5°]$ in the NAO frame, the NAOs can get all their neighbors' information with the integrated image processing method after scanning twice with the forehead camera at the orientation of $\theta = 0°$ and $\theta = 180°$ respectively in the fixed frames. The rotation to 180° from 0° is finished by the rotation control.

Then with the shared neighbors' information by WiFi, the NAOs will plan trajectories (η_i, d_i) with the proposed local position-based method. After that, the NAOs move to their destinations along their trajectories. During the process, rotation control is used to rotate η_i to face the destination, and distance control is then adopted to move straight approximately until they arrive at the destination. One example is shown in Figure 24.

5.3 Discussion

The performance of the local position-based method has been tested both in the simulation with up to 15 robots and in the real experiments with 3 NAOs. From a great deal of simulations in Matlab, the largest allowed noise ranges to achieve a pyramid pattern is carried out: the relative distance noises $\delta_{d_{ij}} \in \pm 0.1 d_e$, the planned trajectory noises $\delta_{d_i} \in \pm 0.1 d_e$, and $\delta_{\eta_i} \in \pm 0.05 \theta_e$. And the local position-based method also can be used in the static obstacle environment with certain translation noises.

Except for being used to plan trajectories in the obstacle-free or static obstacle environments, the collision avoidance strategy can also provide the optimized expected formation pattern and the position relationship among robots for the application of distributed methods in the more complex environment (such as unexpected dynamic object environment). With the position relationship, each robot can plan and update its trajectory according to its neighbors' information in the distributed methods.

At the same time, the success of the pyramid building with 3 NAOs in the indoor environment verifies the feasibility and convergence of the proposed integrated image processing method and the local position-

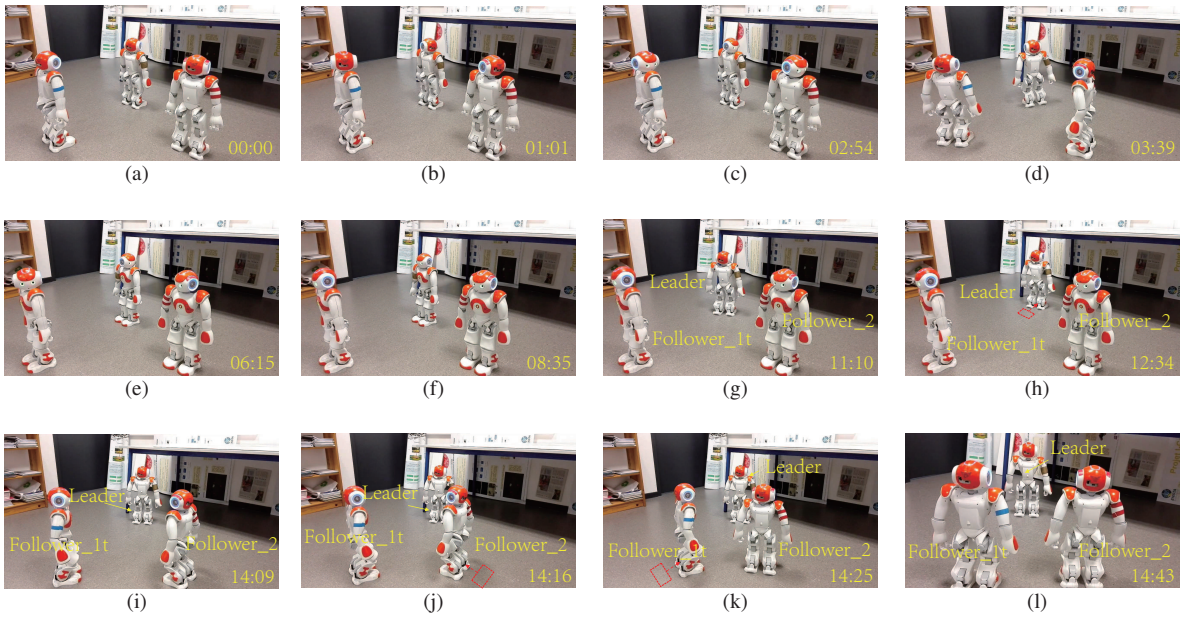


Figure 24 (Color online) An example of pyramid building. (a) Initial distribution of NAOs; (b) the NAOs search neighbors at $\theta = 0°$ with the $\varphi \in [0°, 119°]$; (c) the NAOs search neighbors at $\theta = 0°$ with the $\varphi \in [-119°, 0°]$; (d) the NAOs are rotating to $\theta = 180°$ from $\theta = 0°$ with the rotation control; (e) the NAOs search neighbors at $\theta = 180°$; (f) NAOs are exchanging information among NAOs by WiFi through the PC, the leader and followers are identified, and the trajectories are planned well with the proposed method; (g) the leader is rotating to the destination with the rotation control; (h) the leader has arrived at its destination with the distance control; (i) the leader rotates to the pyramid orientation and the followers begin rotating to face to their destinations with the rotation control; (j) follower_2 arrives at its destination, while follower_1 still rotates to its destination; (k) followers_2 has rotated to the pyramid orientation and follower_1 has arrived at its destination; (l) after follower_1 rotates to the pyramid orientation, the pyramid building is achieved.

based method after repeating experiments. The errors of relative distances between the NAOs are within ± 0.1 m with an expected distance $d_e = 0.8$ m, and the errors of relative angles are in the allowed range with the visual compass providing the feedback.

But sometimes the pyramid building with NAOs fails or has a large error. The reasons are: (1) the failure of recognition of NAOs due to the overlapping of NAOs and light noises in the image; (2) the error of visual compasses because of the dynamic features (the mobile neighboring NAOs) in two adjacent images; (3) the large movement errors of the NAO caused by the NAOs aging. Further, the initial range among NAOs cannot be too large because of the limited recognition range for each NAO ([0.5 m, 1.45 m]). Usually, the initial distances among NAOs are within [0.6 m, 1.2 m].

However, the stability of the pyramid building by NAOs can be improved. (1) The recognition algorithm can be perfected by increasing the features of masks in the M-base (such as colors and contour factors) and by reducing the influence of the light noises in the images. (2) The movement performance of NAOs can be improved by updating the feature extraction method of the visual compass to prevent the emergence of the situation in Figure 22(b), and by adopting the closed-loop control to the distance control with the video cameras providing the feedback (such as the size of features), or using new NAOs.

Moreover, the other NAO sensors can be adopted to detect the obstacles. Usually, the forehead camera can find obstacles, and the bottom camera on the chin can help to observe the ground. But if they fail, the feet bumpers [27] can sense the obstacles when the foots move forward and collide with them (the NAO can stand up by itself if it falls down); and the arm tactile sensors [27] can perceive them when the NAOs touch against the obstacles or other objects. In these cases, based on the foot bumpers or arm tactile sensors, the NAOs can go around the obstacles. In addition, the sonars have a large working range, [0.25 m,2.55 m] [28], as a result, the sonars can be used to detect the surroundings and prejudge whether there are objects in the long distance and measure the distances with the objects. Then the NAOs can update the trajectories to prevent the obstacles using the information got by sonars and cameras.

6 Conclusion

In this paper, we focus on solving a pyramid pattern building problem for a fleet of NAOs based on the forehead camera of each NAO with minimum WiFi communication. The contributions are twofold: (1) an estimation method of the local positions based on automatic recognition of the neighbors and their relative poses is put forward; (2) a local position-based method associated with an $\mathcal{O}(n \log n)$ complexity collision avoidance strategy is proposed to plan trajectories and build a pyramid pattern.

Specifically, the local positions are estimated by successively applying an integrated image processing method: (1) an M-base is generated to store the features of NAOs, and a cross-correlation method is introduced to recognize the NAOs; (2) relative distance and angle models are established to get the local information from a single image; (3) to overcome the lack of actual compass, the orientation of the NAO is estimated from a pair of images with a visual compass. Then in the local position-based method, (1) a neighbor-check method is put forward to distinguish the IDs of NAOs; (2) with a common frame constructed, an $\mathcal{O}(n \log n)$ complexity collision avoidance strategy inspired by one graph theory is given to optimize the pyramid distribution and plan the straight non-intercrossing trajectories.

The feasibility and stability of the $\mathcal{O}(n \log n)$ complexity collision avoidance strategy have been validated with up to 15 robots in the obstacle-free and static obstacle environments respectively in Matlab. The performances of all our methods are verified by doing experiments with 3 NAOs. But sometimes the formation building is not satisfying because of the failure of some NAOs' recognition, the error of visual compass, or large errors of NAOs' movements.

In the future work, we will give another coordination method to build the pyramid pattern in the more complex environment, such as the dynamic obstacle evnironment. Furthermore, the image processing methods will be studied deeply to improve the stability of NAOs' movement. Moreover, the other NAO sensors (e.g., sonars, foot bumpers, and arm tactile sensors) will be considered to be used to detect the obstacles.

Acknowledgements This work was supported by National Natural Science Foundation of China (Grant Nos. 61571478, 61601428, 51709245, 51509229).

References

1 Gu D B. A game theory approach to target tracking in sensor networks. IEEE Trans Syst Man Cybern B, 2011, 41: 2–13
2 Jin Y C, Guo H L, Meng Y. A hierarchical gene regulatory network for adaptive multirobot pattern formation. IEEE Trans Syst Man Cybern B, 2012, 42: 805–816
3 Ahmad M D, Kasim M A A, Mohammed M A, et al. Multi-robot system for real-time sensing and monitoring. In: Proceedings of the 15th International Workshop on Research and Education in Mechatronics (REM), El Gouna, 2014
4 Arai T, Pagello E, Parker L E. Guest editorial advances in multirobot systems. IEEE Trans Robot Autom, 2002, 18: 655–661
5 Li X, Zhu D Q, Qian Y. A survey on formation control algorithms for multi-AUV system. Unmanned Syst, 2014, 2: 351–359
6 Benzerrouk A, Adouane L, Martinet P. Stable navigation in formation for a multi-robot system based on a constrained virtual structure. Robot Auton Syst, 2014, 62: 1806–1815
7 Balch T, Arkin R C. Behavior-based formation control for multirobot teams. IEEE Trans Robot Autom, 1998, 14: 926–939
8 Sun J Y, Tang J, Lao S Y. Collision avoidance for cooperative UAVs with optimized artificial potential field algorithm. IEEE Access, 2017, 5: 18382–18390
9 Loria A, Dasdemir J, Jarquin N A. Leader-follower formation and tracking control of mobile robots along straight paths. IEEE Trans Control Syst Technol, 2016, 24: 727–732
10 Oh K K, Park M C, Ahn H S. A survey of multi-agent formation control. Automatica, 2015, 53: 424–440
11 Alonso-Mora J, Breitenmoser A, Rufli M, et al. Optimal reciprocal collision avoidance for multiple non-holonomic robots. In: Proceedings of the 10th International Symposium on Distributed Autonomous Robotic Systems, Lausanne, 2010. 203–216
12 Paulos J, Eckenstein N, Tosun T, et al. Automated self-assembly of large maritime structures by a team of robotic boats. IEEE Trans Autom Sci Eng, 2015, 12: 958–968
13 Xia Y Q, Na X T, Sun Z Q, et al. Formation control and collision avoidance for multi-agent systems based on position estimation. ISA Trans, 2016, 61: 287–296

14 Nikou A, Verginis C K, Dimarogonas D V. Robust distance-based formation control of multiple rigid bodies with orientation alignment. 2017. ArXiv:1611.01824
15 Oh K K, Ahn H S. Distance-based control of cycle-free persistent formations. In: Proceedings of IEEE International Symposium on Intelligent Control, Denver, 2011. 816–821
16 Bartlett S L, Hampapur A, Huber M J, et al. Vision for mobile robots. In: Image Technology. Berlin: Springer, 1996
17 Fredslund J, Mataric M J. A general algorithm for robot formations using local sensing and minimal communication. IEEE Trans Robot Autom, 2002, 18: 837–846
18 Gouaillier D, Hugel V, Blazevic P, et al. Mechatronic design of NAO humanoid. In: Proceedings of IEEE International Conference on Robotics and Automation, Kobe, 2009. 769–774
19 Wang X M, Zerr B, Thomas H, et al. Pattern formation for a fleet of auvs based on optical sensor. In: Proceedings of OCEANS-17, Aberdeen, 2017
20 Kim Y, Mesbahi M. On maximizing the second smallest eigenvalue of a state-dependent graph Laplacian. IEEE Trans Autom Control, 2006, 51: 116–120
21 Alonso-Mora J, Breitenmoser A, Rufli M, et al. Multi-robot system for artistic pattern formation. In: Proceedings of IEEE International Conference on Robotics and Automation, Shanghai, 2011. 4512–4517
22 Yu S, Barca J C. Autonomous formation selection for ground moving multi-robot systems. In: Proceedings of IEEE International Conference on Advanced Intelligent Mechatronics, Busan, 2015. 54–59
23 Benozzi L. A detection strategy for line formation of a team of humanoid robots. Dissertation for Master Degree. Florence: University of Florence, 2017
24 Aldebaran: Nao software 1.14.5 documentation/hardware/nao technical overview/joints. 2013. http://doc.aldebaran.com/1-14/family/robots/jointsrobot.html
25 Aldebaran: Nao software 1.14.5 documentation/reference/naoqi api/naoqi vision/alvisualcompass. 2013. http://doc.aldebaran.com/1-14/naoqi/vision/alvisualcompass.html?highlight=compass
26 Fujinaga N, Yamauchi Y, Ono H, et al. Pattern formation by oblivious asynchronous mobile robots. SIAM J Comput, 2015, 44: 740–785
27 Aldebaran: Nao software 1.14.5 documentation/hardware/nao technical overview/contact and tactile sensors. 2013. http://doc.aldebaran.com/1-14/family/robots/contact-sensorsrobot.html
28 Aldebaran: Nao software 1.14.5 documentation/hardware/nao technical overview/sonars. 2013. http://doc.aldebaran.com/1-14/family/robots/sonarrobot.html