

A Consensus-Based Approach for Visual Servo Control of Multiple Mobile Robots

Yuhang Chen, Jiahui Qin, Shuai Wang

Abstract—This paper addresses a consensus-based approach for visual servo control of multiple mobile robots. The basic idea is to utilize single-integrator consensus techniques to design a image-based visual servo control algorithm for each robot under a certain network. The aim of this work is to extract common objects between different images captured by different robots and adjust the positions and orientations of robots to concentrate on common objects. We show that if the union of communication graphs has a directed spanning tree, the states of robots would achieve consensus based on the relative states. At the end of this paper we present experiments to verify the feasibility and effectiveness of proposed algorithms.

Index Terms—visual consensus, multi-robot system, image-based visual servoing control, image matching

I. INTRODUCTION

Mobile robots that integrate visual servo control have been widely used in various robotic tasks such as navigation and transportation. According to whether the error signal is defined in 3D coordinates or image coordinates, the systems can simply classified as position-based visual servoing (PBVS) and image-based visual servoing (IBVS) [1]. Compared to the PBVS methods, IBVS methods have gained more attention since they offer advantages in robustness to camera and target calibration errors and simple extension to applications involving multiple cameras. [2].

With the development of mobile robots, there are many tasks that can be accomplished by the collaboration of a group of robots such as cooperative transportation, exploration, and mapping [3]. However, current visual servoing algorithms mainly focus on utilizing visual information captured from a single robot. There is a growing need to extract visual features from multiple robots for facilitating visual servo control of a group of mobile robots.

In this paper, we present a consensus-based approach for IBVS control problems of multiple mobile robots. Our contributions of this work can be concluded as follows. One is we first propose and formulate the visual consensus problem, which refers to all the robots under a certain network concentrating on the common targets by adjusting their positions and orientations. Another one is that we introduce the single-integrator consensus protocols to design the algorithm, which is easily implementable and could make full use of information of all robots. Besides, existing consensus algorithms mainly focus on a team of robots reaching an agreement on some shared states like position, orientation, velocity, etc. But in our work, we emphasize on utilizing such distributed techniques to design an IBVS control approach for each robot in order to reach the predefined state consensus.

The remainder of this paper is structured as follows. The second part summarizes the related works. We formulate the visual consensus problem and introduce the preliminary notions in the third part. In Part IV, we proposed algorithms for image matching and state consensus. Experimental results are presented for illustration in Part V. Conclusions and future work are shown in the final part.

II. RELATED WORKS

A. Image Matching

One key problem in IBVS is how to determine the position and orientation of each feature relative to robot end-effector, which would be used in the image error function. This issue refers to two significant concepts in computer vision, feature detection and image matching. Feature detection aims to identify interested image primitives (e.g. points, lines/curves, and regions), for the purpose of highlighting salient visual cues in digital images [4]. Image matching is the process of finding one or more transformations, so that two or more images from the same scene taken at different times, from different viewpoints, or by different sensors are spatially consistent [5].

The development of image matching using a set of local points of interest can be traced back to the work of Moravec [6]. And Harris and Stephens [7] improved the method to make it more repeatable under small image variations and near edges. Schmid and Mohr [8] introduced invariant local features to deal with a large database of images. In this paper, the matching component aims to extract common objects between different images. To satisfy the reliable and real-time requirements of IBVS, we consider feature detection techniques that are robust to image transformations like scale, rotation, noise, illumination and affine transformations to perform the matching task.

B. Consensus of Multi-robot Systems

Consensus problem has a long history in computer science and form the foundation of the field of distributed computing. The first study of consensus problem was motivated by work of DeGroot [9] in 1970s. And then Olfati-Saber and Murray [10] start the pioneer work for posing and solving consensus problems for dynamic systems. There are a variety of researches about practical applications of consensus, such as spacecraft formation flying, sensor networks, congestion control in communication networks and cooperative surveillance [11] - [13].

The aim of our work is to make robots under certain network could spontaneously focus on some common targets by negotiating with neighbors, for example, tracking an intruder

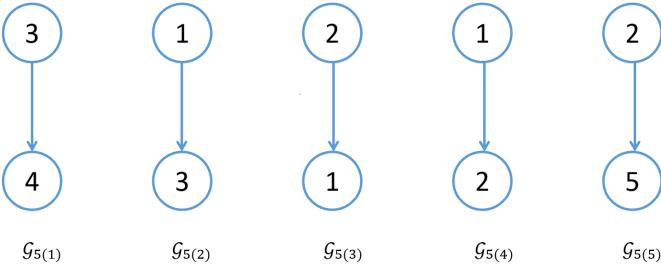


Fig. 1: Interaction topologies for five robots.

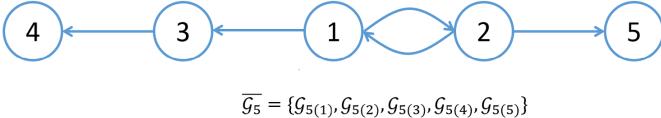


Fig. 2: The union of $\mathcal{G}_{5(1)}, \mathcal{G}_{5(2)}, \mathcal{G}_{5(3)}, \mathcal{G}_{5(4)}, \mathcal{G}_{5(5)}$

without prior information. Or in other word, all robots achieve kind of predefined consensus, which means visual consensus in this paper.

III. PROBLEM FORMULATION

The following subsections briefly recall the corresponding concepts needed to fully understand the proposed algorithms, which is given in next part.

A. Basic Concepts of Graphs

For a system with n robots, its network topology can be modeled as a directed graph denoted by $\mathcal{G}_n \triangleq (\mathcal{V}_n, \varepsilon_n)$, where $\mathcal{V}_n = \{1, 2, \dots, n\}$ is the node set and $\varepsilon_n \subseteq \mathcal{V}_n \times \mathcal{V}_n$ is the edge set. Specifically, the directed edge denoted by an ordered pair (i, j) means that robot j can access the state information from robot i . Note that self-edges (i, i) are not allowed. Let $\mathcal{A}_n = [a_{ij}] \in \mathbb{R}^{n \times n}$ be the adjacency matrix associated with \mathcal{G}_n . a_{ij} is a positive weight if $(j, i) \in \varepsilon_n$, and $a_{ij} = 0$ otherwise. We define $a_{ii} = 0$ since self-edges are not allowed.

A directed graph has a directed spanning tree if there exists at least one node that has a directed path to every other node. In this paper, we consider the dynamic change of interaction topology. Let $\overline{\mathcal{G}_n} = \{\mathcal{G}_{n(1)}, \mathcal{G}_{n(2)}, \dots, \mathcal{G}_{n(m)}\}$ denotes the set of all possible directed interaction graphs defined for \mathcal{A} . Fig. 1 and Fig. 2 show an example of the interaction topologies of five robots at different time and the union of them. We assume that the interaction topology switched randomly among $\mathcal{G}_{5(1)} - \mathcal{G}_{5(5)}$. As we can see, none of them has a directed spanning tree. However, the union of these graphs has a directed spanning tree, which is shown in Fig. 2.

B. Image Matching Notations

We use R_1, R_2, \dots, R_n to denote n robots under a network. Each robot equips with a vision sensor and I_i denotes the image captured by the vision sensor of R_i . Obviously, even R_i and R_j capture the same scene at the same time, I_i and I_j would be quite different since they are taken from different

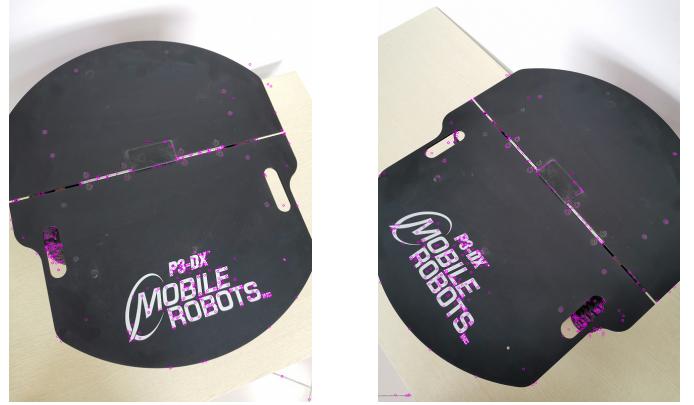


Fig. 3: The two images show the keypoints of different viewpoints. The number of keypoints could be adjusted by applying a threshold on minimum contrast or ratio of principal curvatures.

viewpoints. Fig. 3 gives an example of a Pioneer-3DX robot taken from different viewpoints. In addition, noise, illumination and other environmental factors may also influence the results. Due to the real-time requirement of the robot, it is unacceptable to directly compare the pixels of I_i and I_j during matching process. Therefore, we extract some salient features from images to complete such matching tasks, denoted as keypoints (kp). kp_i contains the necessary information of features of I_i , such as location, scale, orientation. For each keypoint, we construct a descriptor (des) for it, which is invariant to changes in illumination and 3D viewpoint.

We could use the descriptors to measure the similarity between features. If I_i and I_j have some features matched, there must be some views in common. $match_{ij}$ denotes the keypoints in I_i that could match with some others in I_j (shown as Fig. 4). Obviously, $match_{ij} \neq match_{ji}$. As we defined earlier, $a_{ii} = 0$, $match_{ij}$ is defined as an empty set if $j = i$. To utilize such matching results for controlling the motion of robots, the final step we need to do is to design a function to map these matched keypoints $match_{ij}$ into a coordinate relative to image reference system, denoted as $center_{ij}$. The function would be presented in the algorithm part. Given all these preliminaries above, we now could formulate the visual consensus problem.

C. Visual Consensus Formulation

Consider information states with single-integrator dynamics given by

$$\dot{\xi}_i = u_i, i = 1, \dots, n, \quad (1)$$

where $\dot{\xi}_i \subseteq \mathbb{R}^m$ denotes the information state and $u_i \subseteq \mathbb{R}^m$ refers to the control input of the i th robot. Generally, a continuous-time consensus algorithm is given by

$$u_i = - \sum_{j=1}^n a_{ij}(t)(\xi_i - \xi_j), i = 1, \dots, n, \quad (2)$$



Fig. 4: The matching results of the two images shown in Fig. 3. Despite of rotation, the two images match perfectly.

where $a_{ij}(t)$ is the (i, j) entry of $\mathcal{A}_n(t)$ at time t . We say a team of robots achieves consensus if for all $\xi_i(0)$ and all $i, j = 1, \dots, n$, $\|\xi_i(t) - \xi_j(t)\| \rightarrow 0$, as $t \rightarrow \infty$. Similarly, for each robot, we define

$$center_i = \frac{\sum_{j=1}^n a_{ij} * center_{ij}}{\sum_{j=1}^n a_{ij}}, i = 1, \dots, n \quad (3)$$

For IBVS control, we define the error state in image coordinates as follows

$$err_i = center_i(t) - center_j(t), i, j = 1, \dots, n \quad (4)$$

Visual consensus is achieved by the team of robots if for all $I_i(0)$, we have $\|err_i\| \rightarrow 0$, $i = 1, \dots, n$, as $t \rightarrow \infty$.

IV. ALGORITHM

The problem of visual consensus consists of image matching and state consensus. The first component aims to extract common objects between different images. The second one shows how to use matching results from Algorithm 1 and visual servoing techniques to obtain the control input of each robot.

A. Image Matching Based on ORB

There are various kinds of methods could be employed in image matching projects. After performance comparison for distorted images, we finally choose Oriented FAST and Rotated BRIEF (ORB) [14] in our work. Algorithm 1 starts with initializing and then each robot obtains images by its own vision sensor. We get corresponding keypoints and descriptors after applying ORB (line 15). Each robot communicates with its neighbors to match these descriptors (line 20). We calculate the Euclidean distance of two descriptors to measure the differences of features. For each des_i , we utilize the K-NearestNeighbor(KNN) algorithm [15] to find the k descriptors from des_j with smallest Euclidean distance (line 9). Since some of the matching results may be misleading, we need to filter them out. We set k as 2, which means that it would return the best two results for each descriptor. In line 24, the first inequation reserves results that distinguishly match better than others. α is a constant that is often set between 0.5 and 1.0. The second inequation guarantees that all the results will under a certain threshold, or in other words, a certain confidence level.

If the distance is less than this threshold, we could consider it as credible. β is a positive constant determined by prior experiments, depending on resolution, illumination and some other environmental factors. After Algorithm 1, we get series of matching results for each robot and its neighbors.

Algorithm 1 Image Matching Based on ORB

```

1: function GETVISIONSENSOR( $R$ )
2:   return the image captured by  $R$ 's vision sensor
3: end function
4: function ORBCOMPUTE( $img$ )
5:   return the results of ORB features(keypoints, features
   descriptor)
6: end function
7: function KNNMATCH( $des_i, des_j, k$ )
8:   calculate the Euclidean distance for each in  $des_i$  and
    $des_j$ 
9:   for each descriptor in  $des_i$ , return the  $k$  descriptors
   from  $des_j$  with smallest distance
10: end function
11: function MAIN( )
12:   initialize all robots and sensors
13:   for  $i = 1 \rightarrow n$  do
14:      $I_i \leftarrow$  GETVISIONSENSOR( $R_i$ )
15:      $kpi, des_i \leftarrow$  ORBCOMPUTE( $I_i$ )
16:   end for
17:   for  $i = 1 \rightarrow n$  do
18:     for  $j = 1 \rightarrow n, j \neq i$  do
19:       if  $a_{ij} > 0$  then
20:          $match_{ij} \leftarrow$  KNNMATCH( $des_i, des_j, 2$ )
21:       end if
22:     end for
23:     for all  $m, n$  in  $match_{ij}$  do
24:       if  $m.distance > \alpha * n.distance$  or
    $m.distance > \beta$  then
25:         delete  $m$  from  $match_{ij}$ 
26:       end if
27:     end for
28:   end for
29: end function

```

B. State Consensus Based on Matching Results

Before Algorithm 2, we define a matching matrix, $\mathcal{B} = [b_{ij}] \in \mathbb{R}^{n \times n}$ as follows:

$$b_{ij} = \begin{cases} a_{ij}, match_{ij} \text{ is not empty} \\ 0, \text{otherwise} \end{cases} \quad (5)$$

Similar to adjacency matrix used to represent the topology of a graph, the matching matrix denotes the matching relationships of a graph. $b_{ij} = 0$ means that the views captured by R_i and R_j have nothing matched.

Algorithm 2 shows how to use matching results from Algorithm 1 to get the control input of each robot. For a node with indegree of zero, the robot corresponding should

remain in place. Therefore, we initialize all $center_i$ to the center of I_i . Now we get a series of matching results. One of the questions is to determine which object they belong to in the original image. Clustering is the most frequently used method in segmentation [16]. After comparing the complexity of computation and implementation, we choose K-Means to carry out this task [17]. The number of clusters could be set manually. The first function simply return the cluster with most keypoints because it tends to represent the largest common part of two images (line 3). After getting $center_{ij}$ (line 17), which denotes the center for the common view of I_i and I_j , we get $center_i$ based on $center_{ij}$ (line 20).

To perform IBVS control, the error is defined as (4). In Line 24, a continuous-time consensus algorithm is presented based on this error function, which would be crucial for adjusting the position and orientation of a robot. During the process of the error converging to zero, we could infer that the robot tend to adjust itself to focus on the largest common view between itself and its neighbors. Finally, we give a lemma to theoretically analysis the convergence of our algorithms.

Algorithm 2 State Consensus Based on Matching Results

```

1: function KMEANSPREDICTOR(data)
2:   use KMeans cluster algorithm to segment the matched
   keypoints
3:   return the cluster with most keypoints
4: end function
5: function CALCULATECENTER(match, kp)
6:   data  $\leftarrow$  select the keypoints in kp that reserved in
   match
7:   res  $\leftarrow$  KMEANSPREDICTOR(data)
8:   return the center of res
9: end function
10: function MAIN()
11: input: all matching results match, kp
12: output: control input u
13:   initialize all  $center_i$  to the center of  $I_i$ 
14:   for  $i = 1 \rightarrow n$  do
15:     for  $j = 1 \rightarrow n, j \neq i$  do
16:       if  $b_{ij} > 0$  then
17:          $center_{ij} \leftarrow \text{CALCULATECEN-}$ 
            $\text{TER}(\text{match}_{ij}, \text{kp}_j)$ 
18:       end if
19:     end for
20:     if  $\sum_{j=1}^n b_{ij} = 0$  then
21:       Could not achieve consensus! Return
22:     end if
23:      $center_i = \frac{\sum_{j=1}^n b_{ij} * center_{ij}}{\sum_{j=1}^n b_{ij}}$ 
24:      $u_i = -\sum_{j=1}^n b_{ij} * (center_i - center_j)$ 
25:   end for
26: end function
```

C. Convergence Analysis

In this section, we give a lemma and an example to illuminate the convergence of proposed algorithms.

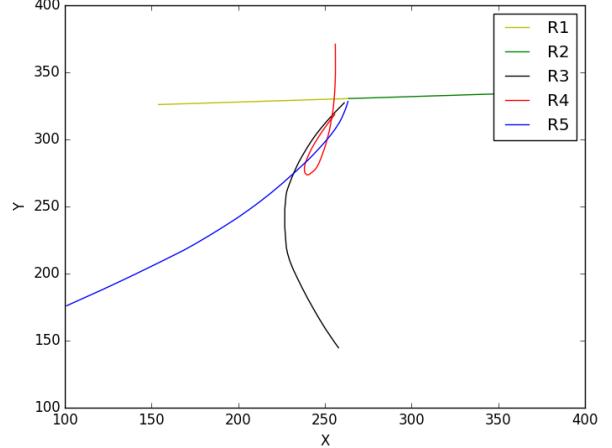


Fig. 5: Visual consensus with topologies randomly switching from Fig. 1

Lemma 1: [18] Let t_1, t_2, \dots be an infinite time sequence at which the interaction graph or weighting factors switch and $\tau_i = t_{i+1} - t_i \in \Upsilon, i = 0, 1, \dots$. Let $\mathcal{G}(t_i) \in \bar{\mathcal{G}}$ be a switching interaction graph at time $t = t_i$ and $\sigma_{ij}(t_i) \in \bar{\sigma}$, where σ is a finite set of arbitrary positive numbers. The continuous time update scheme

$$\dot{\xi}_i(t) = - \sum_{j=1}^n \sigma_{ij}(t) G_{ij}(t) (\xi_i(t) - \xi_j(t)) \quad (6)$$

achieves consensus asymptotically for \mathcal{A} if there exists an infinite sequence of uniformly bounded, non-overlapping time intervals $[t_{i_j}, t_{i_j+l_j}], j = 1, 2, \dots$, starting at $t_{i_1} = t_0$, with the property that each interval $[t_{i_j+l_j}, t_{i_{j+1}}]$ is uniformly bounded and the union of the graphs across each such interval has a spanning tree. Furthermore, if the union of the graphs after some finite time does not have a spanning tree, then consensus cannot be achieved asymptotically for \mathcal{A} .

Let Γ and $\bar{\Gamma}$ denote the directed graph and the set of dynamical interaction topology associated with matching matrix \mathcal{B} . $\Gamma(t_i) \in \bar{\Gamma}$ denotes the interaction graph at time $t = t_i$. From Lemma 1, we could get that our algorithms achieve consensus if the union of matching topologies has a spanning tree across each bounded, non-overlapping interval.

Intuitively, the assertion above is a special case of Lemma 1. Furthermore, if I_i and I_j have none of any common parts, they would not achieve consensus even R_i and R_j could communicate with each other. In such a situation, the $match_{ij}$ would not contribute to $center_i$, which means that the state of R_j does not contribute to the state of R_i . Lemma 1 guarantee the convergence of our algorithms. In other words, as long as the union of the graphs associated with matching matrix has a spanning tree across each interval, the center of the vision sensor of each robot would gradually convergence to the common part of the scene, which means that visual consensus is achieved.



Fig. 6: The experiment done in real environment.

To illuminate this conclusion, we give an example here. We set $\alpha = 0.92$ and $\beta = 360.0$. The topology is shown as Fig. 1 and Fig. 2. We assume that the interaction topology switches randomly among $\mathcal{G}_1 - \mathcal{G}_5$ at each random time. And the initial state is calculated from images captured arbitrarily by five robots. To simplify, we make $\mathcal{B} = \mathcal{A}$, which means that robots who could communicate with each other also have matching parts of view. Besides, we make every neighbor of a robot contributes equally, that is, $a_{ij} = 1$ if $(j, i) \in \varepsilon_n$.

Fig. 5 shows the visual consensus results based on Algorithm 1 and Algorithm 2. We could see that all the states of robots gradually converge to the reference state from their neighbors, for example, $R_3 \rightarrow R_1$, $R_4 \rightarrow R_3$, $R_5 \rightarrow R_2$. Note that the union graph $\bar{\mathcal{G}}$ has two spanning tree with root 1 and root 2. Therefore all the states would finally converge to where R_1 and R_2 converge. Since the the union of graphs has a spanning tree, consensus could be achieved even none of the topologies has not a directed spanning tree at any time.

V. EXPERIMENTAL RESULTS

In this section, we present experimental results of three robots under dynamically changing interaction topologies. The experiment of this work was implemented based on ROS (Robot Operating System) [19]. ROS is an open-source operating system, which is particularly suitable for distributed multi-robot system. We construct a scene, including all the robots to be controlled and targets to be observed. Each robot corresponds to an independent node in ROS. Robots transmit and obtain information by publishing and subscribing topics. The output of our algorithms is transformed into the control input of the robot based on IBVS.

For simplify, we follow the assumptions in Covgence Analysis section. In fact, if one of the robot do not satisfy the assumption of sharing common views with neighbors, we could simply make it adjust its orientation by slowly rotating. If there exists one robot no matter how it rotates itself, it could not find matching parts with others. Obviously it is an

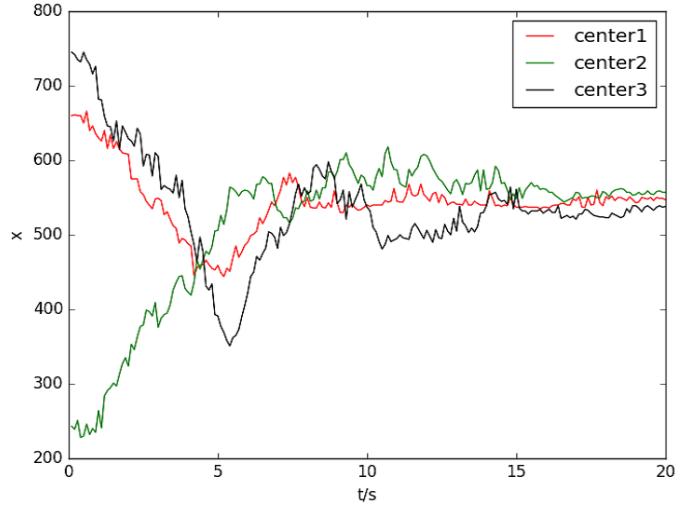


Fig. 7: $center_i$ of three robots during the experimental process.

isolated node. From Lemma 1 we could know that a graph with an isolated node would never achieve consensus since the union of the graphs does not have a spanning tree.

Fig. 6 shows our experiment done in the real environment. We use ZED as a visual sensor to observe the environment and capture the targets. NVIDIA TX2 is embedded on the robot to execute our algorithms and issue control instructions. The Pioneer-3DX robot acts as an executing mechanism to complete instructions by adjusting its position and posture.

Fig. 7 shows the variation of $center_i$, $i = 1, 2, 3$ with time during the experimental process. We could see from the image that $center_2$ converges to $center_1$, $center_3$ converges to $center_2$. And all states converge at $t = 20s$. Comparing Fig. 7 with Fig. 5, the curves in Fig. 7 are not completely smooth since image matching would be influenced by robot motion, illumination and some other factors in the real environment. And due to the existence of friction, the error between different states could not eventually converge to zero. From Fig. 8 we could see that in the beginning, Robot 2 and Robot3 could only capture part of the quadrotor. As time goes by, they all gradually concentrate on the quadrotor, which means that visual consensus are achieved. As for the reason why concentrating on the quadrotor, it is because the quadrotor holds the most features.

VI. CONCLUSION AND FUTURE WORK

In this paper, we discussed a consensus-based approach for visual servo control of multiple mobile robots. To this end, we have formulated the visual consensus problem and put forward two algorithms under dynamically changing topology. After that, we give a theoretical analysis of the convergence of proposed algorithms. Besides, the experimental results are presented for demonstrating this problem.

Although the combination of image matching and IBVS work well, there are still some issues to be handled. The first is how to make full use of the depth data to extend the algorithm for moving targets. Another interesting direction is to use some

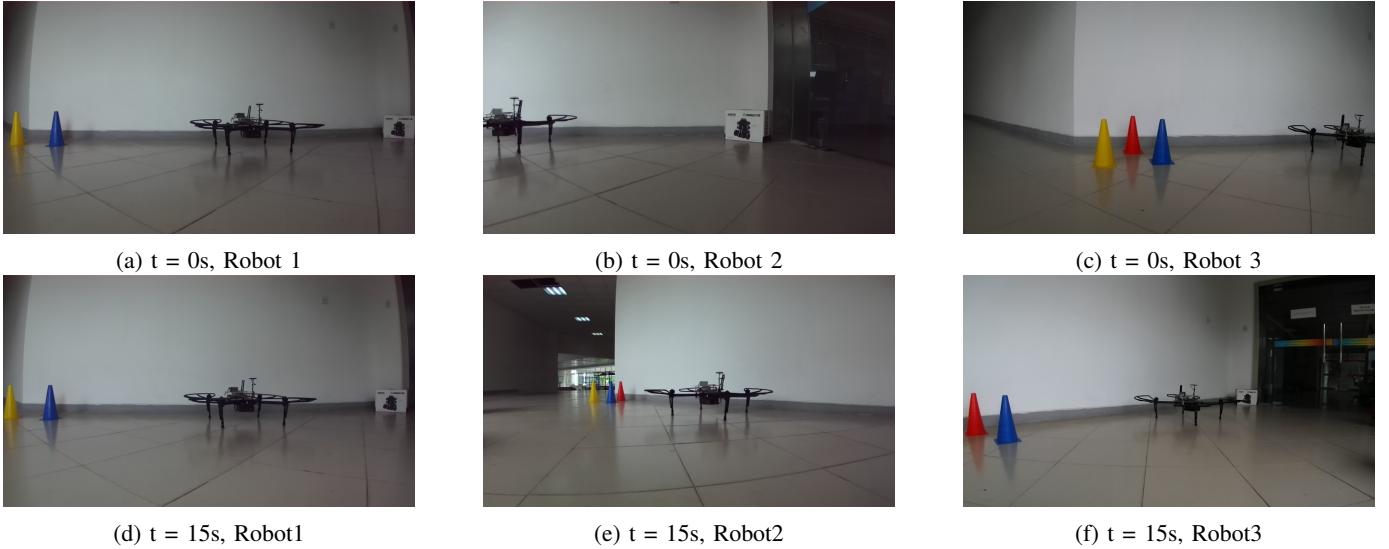


Fig. 8: Fig. 8a- 8c show the initial images captured by three robots. Fig. 8d- 8f show the images captured at t=15s by three robots.

computer vision techniques, such as foreground extraction and saliency detection to reduce the computational complexity of the algorithm.

REFERENCES

- [1] S. Hutchinson, G. D. Hager, P. I. Corke, A tutorial on visual servo control, *IEEE Trans. Robot. Automat.*, vol. 12, pp. 651-670, 1996.
- [2] T. Hamel, R. Mahony, Visual servoing of an under-actuated dynamic rigid-body system: an image-based approach, *IEEE Trans. Robot. Automat.*, vol. 18, pp. 187-198, 2002.
- [3] R. Simmons, D. Apfelbaum, W. Burgard, *et al*, Coordination for multi-robot exploration and mapping, In :*Proceedings of the AAAI National Conference on Artificial Intelligence*, Austin, pp. 299-308, 2000.
- [4] Y. Li, S. Wang, Q. Tian, *et al*, A survey of recent advances in visual feature detection, *Neurocomputing*, vol. 149, pp. 736-751, 2015.
- [5] T. Lindeberg, Image matching using generalized scale-space interest points, *J. Math. Imaging Vis.*, vol. 52, pp. 3-36, 2015.
- [6] H. P. Moravec, Rover visual obstacle avoidance, In: *Proceedings of International Joint Conference on Artificial Intelligence(IJCAI)*, Vancouver, pp. 785-790, 1981.
- [7] C. G. Harris, M. Stephens, A combined corner and edge detector, In: *Proceedings of Alvey Vision Conference(AVC)*, Manchester, pp. 147-151, 1988.
- [8] C. Schmid, R. Mohr, Local grayvalue invariants for image retrieval, *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, pp. 530-534, 1997.
- [9] M. H. Degroot, Reaching a consensus, *J. Am. Stat. Assoc.*, vol. 69, pp. 118-121, 1974.
- [10] R. O. Saber, R. M. Murry, Consensus problems in networks of agents with switching topology and time-delays, *IEEE Trans. Autom. Control*, vol. 49, pp. 1520-1533, 2004.
- [11] T. H. Chang, N. Angelia, S. Anna, Distributed constrained optimization by consensus-based primal-dual perturbation method, *IEEE Trans. Autom. Control*, Vol. 59, pp. 1524-1538, 2014.
- [12] G. Antonelli, Interconnected dynamic systems: An overview on distributed control, *IEEE Control Syst. Mag.*, vol. 33, pp. 76-88, 2013.
- [13] M. Guo, D. V. Dimarogonas, Consensus with quantized relative state measurements, *Automatica*, vol. 49, pp. 2531-2537, 2013.
- [14] E. Rublee, V. Rabaud, K. Konolige, *et al*, ORB: an efficient alternative to SIFT or SURF, In: *Proceedings of International Conference on Computer Vision (ICCV)*, Barcelona, pp. 2564-2571, 2011.
- [15] X. D. Wu, V. Kumar, J. Rehder, *et al*, Top 10 algorithms in data mining, *Knowl. Inf. Syst.*, vol. 14, pp. 1-37, 2008.
- [16] F. Schwenker, E. Trentin, Pattern classification and clustering: A review of partially supervised learning approaches, *Pattern Recognit. Lett.*, Vol. 37, pp. 4-14, 2014.
- [17] A. K. J, Data clustering: 50 years beyond K-means, *Pattern Recognit. Lett.*, Vol. 31, pp. 651-666, 2010.
- [18] W. Ren, R. W. Beard, Consensus Seeking in Multi-agent Systems Under Dynamically Changing Interaction Topologies, *IEEE Trans. Autom. Control*, vol.50, pp.655-661, 2005.
- [19] I. A. Susan, M. Moll, L. E. Kavraki, The open motion planning library, *IEEE Robot. Autom. Mag.*, vol. 19, pp.72-82, 2012.