

Article

Line and V-Shape Formation Based Distributed Processing for Robotic Swarms

Jian Yang ^{1,†,‡}, Xin Wang ^{1,†,‡,*} and Peter Bauer ^{2,‡}

¹ Department of Mechanical and Automation Engineering, Harbin Institute of Technology Shenzhen, Shenzhen 518055, China; yangjian_hit@yahoo.com

² Department of Electrical Engineering, University of Notre Dame, Notre Dame, IN 46656, USA; pbauer@nd.edu

* Correspondence: wangxinsz@hit.edu.cn; Tel.: +86-755-2603-3286

† Current address: D414, HIT Campus, Shenzhen University Town, Shenzhen 518055, China.

‡ These authors contributed equally to this work.

Received: 7 June 2018; Accepted: 1 August 2018; Published: 3 August 2018



Abstract: Efficient distributed processing is vital for collaborative searching tasks of robotic swarm systems. Typically, those systems are decentralized, and the members have only limited communication and processing capacities. What is illustrated in this paper is a distributed processing paradigm for robotic swarms moving in a line or v-shape formation. The introduced concept is capable of exploits the line and v-shape formations for 2-D filtering and processing algorithms based on a modified multi-dimensional Roesser model. The communication is only between nearest adjacent members with a simple state variable. As an example, we applied a salient region detection algorithm to the proposed framework. The simulation results indicate the designed paradigm can detect salient regions by using a moving line or v-shape formation in a scanning way. The requirement of communication and processing capability in this framework is minimal, making it a good candidate for collaborative exploration of formatted robotic swarms.

Keywords: swarm robotics; sensor networks; pattern formation; distributed processing; collaborative exploration

1. Introduction

Searching large areas for a particular signal or feature is a formidable task that, at the moment, has no satisfactory solution. Typical approaches include the use of very few high complexity robotic devices that randomly explore the area of interest in a sequential fashion [1]. The disadvantages of such approaches are apparent: long search time, high probability of missing the target, high cost, and the possibility of loss of the agent, which again adds to the expected cost value. Also, to keep the detection probability high enough, one has to pursue a large number of false positives, further adding to the time cost for such a scenario. Using multi-robot systems to perform such tasks in parallel will be a better choice.

Swarm robotics is a particular approach for multi-robot systems that takes inspiration from the self-organized behaviors of social animals such as birds, bees and fish groups, etc. [2]. Those systems are generally with a scalable decentralized scheme that the members do not have any global knowledge, and only have local sensing and limited communication capability [3]. This area has been perceived to have relevance in a variety of application areas ranging from exploration of virgin territories (mid-air [4] or underwater [5]), area coverage for military defense to contamination detection or tracking [6], etc. Ordinarily, the members in swarms are homogeneous. However, some complex applications require to use multiple types of sensors and robots simultaneously, all of which could not be integrated into

a single type of agent. Those applications, therefore, require the system have some heterogeneity [7]. Depending on the requirements of different tasks, it may need physically heterogeneous members, or the heterogeneity in behavior could be achieved in physically homogeneous teams by specializing the behaviors [8].

In this paper, we are adopting robotic swarms for collaborative searching tasks. To cope with this, the swarms must first move together and then make a decision together under the condition of limited sensing and communication [9]. One conventional approach is searching randomly in an area to get the optimal target location. This idea is also the basis of particle swarm optimization (PSO) algorithm [10], which have many extensions of imitating species including ants, fish groups as well as glowworms [11]. However, robots in those solutions in an open area have the high probability of losing contact with each other and, therefore, lose the ability to finish the task [12]. By forming a specific formation and moving together to search an unknown area in a scanner-like way is a better option for those tasks.

Formation forming is one of the fundamental research topics of swarm collaborative behaviors. It aims at deploying robots regularly and repetitively that keeps specific distances from each other to get the desired pattern during moving. There are numbers of research works dealing with formation control problems in literature: structure-based methods (leader-follower [13], virtual structure [14]), behavior-based approaches (finite state machine [15], potential fields [16], consensus-based methods [17]) as well as multicellular mechanism-inspired paradigms [18]. In particular, flying geese inspired line or v-shape formation control is well-studied in this area. It is easy to deploy and has the significant potential for searching an area in a cooperative and parallel way. Sousselier et al. proposed a line formation algorithm for the underwater environment searching task [1]. Nathan et al. introduced a forming method which employed a small set of distributed positioning rules, i.e., coalescing, gap-seeking and stationing rules, to guide the movements of artificial flocking agents aimed to get the v-like formation [19]. Li et al. demonstrated a v-shape formation method from control engineering standpoint. By combining the visual communication constraints and a cost function, a gradient-based navigation control algorithm was given for the forming task [20]. However, applying those formations to collaborative searching tasks require processing the sampled data instantly during moving. Especially the swarm member is under the condition of local communication and limited processing capacity.

In fact, the event searching scheme with those formations requires fast processing capability. Since the system is designed to be decentralized, uploading the sampled and possibly pre-processed data to an entity outside the swarm and then communicating the decision back to the appropriate swarm agents is simply not an option. Fast, in-swarm distributed processing is essential for timely decision making to increase the ability to avoid false negatives at the cost of allowing false positives. This paper illustrates an in-swarm distributed processing paradigm for flying geese inspired line and v-shape formations that have the capability of covering and exploring vast areas effectively. We have shown a formation-based method for collaborative decision making in a scanner-like way of a low-complexity robots swarm with distributed processing paradigm [21]. In this work, formation formed swarm was treated as a moving sensor network, in which could utilize some multi-dimensional-based algorithms in a distributed manner [22]. We extended the work to more general cases that address not only the line formation case but also the v-shape case on both distributed signal processing and collaborative decision making aspects. An example of distributed salient region detection algorithm is applied to this paradigm to show the effectiveness.

The main contributions of this paper are the real-time realization of a modified 2-D Roesser model for in-swarm distributed processing of line and v-shape formations. This work extends the results in [22] that only work for rectangular static 2-D sensor networks. The approach allows detecting signals of interest during motion without communicating data into or out of the swarm. A salient region detection example is given to show the effectiveness of the proposed paradigm.

2. Problem Statements

We are proposing to develop a distributed processing scheme that assigns one swarm agent per sampling point and processes data using simple communication with only nearest neighbors. As shown in Figure 1, suppose the line and v-shape formation are already created and moving perpendicular to the direction of formed patterns (v in the Figure). By constraint the information i is only exchanged between closest neighbors, denote u_n and y_n are input and output of moving node n , the distributed processing problem can be written as follows:

$$y_n(t) = f[u_n(t), i_{n-1}(t), i_{n+1}(t), i_n(t-1)] \quad (1)$$

where $i_t(n-1)$ and $i_t(n+1)$ are signals received by node n from the previous and next adjacent member in the swarm respectively. $i_{t-1}(n)$ is the stored self-state of last time slot. That means the system has propagation causality in the directions of signal transmission. Here we assume that the signals that need to be exchanged are simple, and can be transmitted by radio or visual beacons without packet loss. Now the distributed processing problem translates to how to apply suitable filters into this scheme depending on task requirements.

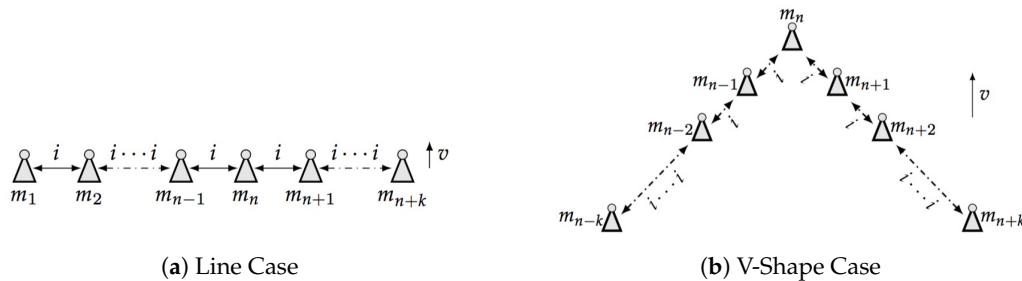


Figure 1. Desired Distributed Processing Procedure.

3. Methods

Due to its many advantages, we propose a modified 2-D Roesser model for distributed processing problem under the line and v-shape formations. This two-dimensional processing model, initially designed for centralized (image) processing, has many features that are of particular interest for the problem at hand: scalability, high network capacity, and low node complexity.

3.1. Standard 2-D Roesser Model

There are several causality structures (and therefore directions of propagation) that can be chosen within this framework. The choice of the structure depends on different aspects of swarm operations and is discussed in more details later on. For reasons of simplicity, we initially introduce the case of “quarter plane causal” of a standard 2-D model, which can easily be transformed to other regions of support, and then present the modified model for line and v-shape processing case.

In its most basic, linear form, the 2-D Roesser state space model is given by [23]:

$$\begin{bmatrix} x_h(n_1+1, n_2) \\ x_v(n_1, n_2+1) \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} x_h(n_1, n_2) \\ x_v(n_1, n_2) \end{bmatrix} + \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} u(n_1, n_2) \quad (2a)$$

$$y(n_1, n_2) = \begin{bmatrix} C_1 & C_2 \end{bmatrix} \begin{bmatrix} x_h(n_1, n_2) \\ x_v(n_1, n_2) \end{bmatrix} + Du(n_1, n_2) \quad (2b)$$

where $(x_h(n_1, n_2), x_v(n_1, n_2))^T$ is the state vector, x_h and x_v are the horizontal and vertical propagating components, $u(n_1, n_2)$ and $y(n_1, n_2)$ are the input and output signals of the filtering process, respectively. The direction of response propagation is shown in Figure 2. The center node (n_1, n_2)

in the figure receives the state information from only the closest adjacent nodes horizontally and vertically (see $x_h(n_1 - 1, n_2)$ and $x_v(n_1, n_2 - 1)$ in the figure). By combining with its own sample input $u(n_1, n_2)$, the node is able to compute its output $y(n_1, n_2)$ as well as the transmission status $x_h(n_1, n_2)$ and $x_v(n_1, n_2)$ respectively. Those status again propagate to next neighbor nodes $(n_1 + 1, n_2)$ and $(n_1, n_2 + 1)$.

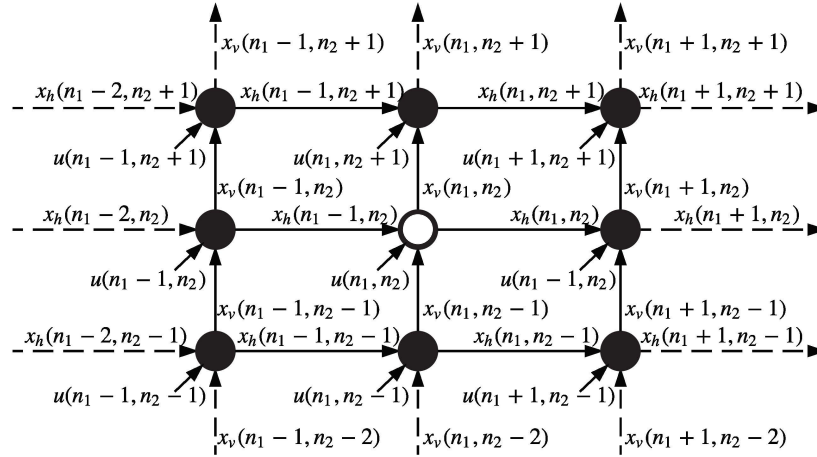


Figure 2. Standard 2-D Roesser model with first quadrant causality.

While the Roesser model in its original form clearly creates first quadrant causality, other options exist. For example on a square lattice, four different quarter-plane causalities correspond to the four different diagonal signal propagation directions, that in turn can be combined to generate the total outputs. The following three causalities exist for a 2-D quarter-plane by changing the left-hand side of Equation (2a) to $\begin{bmatrix} x_h(n_1 - 1, n_2) \\ x_v(n_1, n_2 + 1) \end{bmatrix}$, $\begin{bmatrix} x_h(n_1 - 1, n_2) \\ x_v(n_1, n_2 + 1) \end{bmatrix}$ and $\begin{bmatrix} x_h(n_1 + 1, n_2) \\ x_v(n_1, n_2 - 1) \end{bmatrix}$, which corresponding to 2nd, 3rd and 4th quadrant causalities respectively.

By comparing the standard Roesser model and the distributed processing problem we modeled in Equation (1), we found that the two models have similarities. If we replace $i_{n-1}(t)$, $i_{n+1}(t)$ and $i_n(t-1)$ in (1) with the corresponding horizontal status x_h and vertical status x_v in (2), the distributed processing problem will be solved by transmitting the status variable with certain propagation causalities defined above. The work in [22] provides a distributed processing model for the case of a fixed rectangular two-dimensional lattice of a sensor network. Since the full 2-D sampling lattice is not available at any given time for moving swarm in formations, one cannot perform the algorithm in (2) as outlined in Figure 2. Therefore modified versions need to be formulated for moving line or v-shape cases where the agents need to remember the state of the previous time instant to simulate the vertical status propagation.

3.2. Modified 2-D Roesser Model Based Distributed Processing

Assume for a moment the line-formatted swarm motion is in the direction of the n_2 axis in Figure 3. Then all sampling lattice locations (n_1, n_2) , $n_2 = 0, 1, 2, \dots$ with n_1 fixed are represented by the same swarm agent, uniquely identified as agent n_1 . Sampling at location (n_1, n_2) then just happens t time units ($t = n_2/v$) after sampling at $(n_1, 0)$, where v is the agent speed in the n_2 direction, and agent speed is defined merely as sampling periods per time. In this case, only the horizontal part of Equation (2a) requires communication between neighboring agents. All the computations are executed entirely inside each line agent. Therefore this scheme (unlike the 2-D process before) has a sampling and processing rate limit that is dictated by the swarm speed itself. A slow-moving swarm takes more time to arrive at the next sampling instant n_2 and thus has plenty of time for propagating

the state value from left to right, vice versa, or in both $\pm n_1$ directions. The question of causality choice is almost trivial for this case, only in the first and second quadrant, since the swarm does not have any information about the un-scanned area.

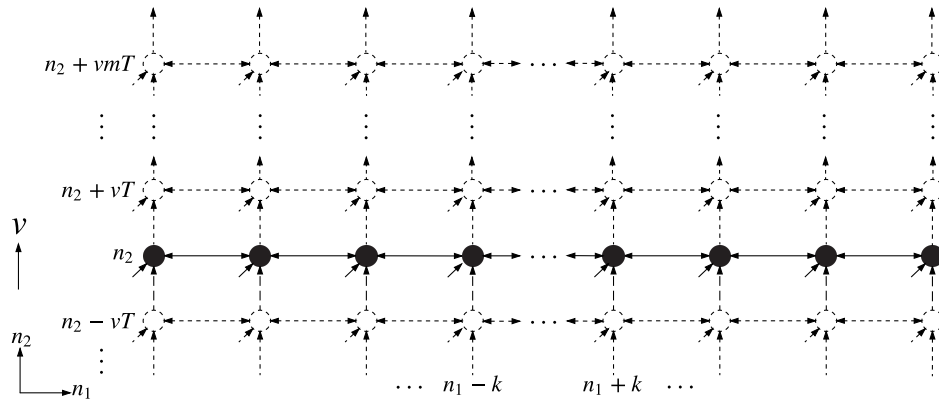


Figure 3. Roeser Model Based Distributed Processing in 1-D Case.

By assigning n_1 as a member among the swarm, the left part members are $(n_1 - 1, \dots, n_1 - k)$ while the right part members are $(n_1 + 1, \dots, n_1 + l)$, the processing procedure of v-shape case and line case are similar in most respects. The shape of the coverage area may differ according to the final formation of the swarm. Details will be shown in the following sections of results and discussions.

3.3. Distributed Salient Region Detection

Salient region detection can be utilized to perform fast scene analysis and target tracking. Here we assume the formations will be used to detect salient regions of an unknown area. In this example, suppose every single agent can acquire image sequences and calculate some simple features, such as average color weight or intensity, which is treated as the input of each agent. According to the center-surrounding mechanism [24], saliency values for those features should be the differences between the current region and its neighbors. The difference equations of this mechanism for a $N_1 \times N_2$ filter with first quadrant causality can be written as:

$$y(n_1, n_2) = \left| \sum_{k_1=0}^{N_1-1} \sum_{k_2=0}^{N_2-1} u(n_1, n_2) - \frac{u(n_1 - k_1, n_2 - k_2)}{N_1 \times N_2} \right| = |\tilde{y}(z_1, z_2)| \quad (3)$$

where u and y are the inputs and outputs respectively. Let $N_1 = N_2 = 2$, the following filter in its z transform can be established for the transfer function above:

$$H(z_1, z_2) = \frac{\tilde{y}(z_1, z_2)}{u(z_1, z_2)} = -\frac{1}{4}z_1^{-1} - \frac{1}{4}z_2^{-1} - \frac{1}{4}z_1^{-1}z_2^{-1} + \frac{3}{4} \quad (4)$$

Based on the multidimensional system theory [25], one corresponding Roeser state space model for this transfer function in the first and second quadrant causality is written as (5) and (6) respectively.

$$\begin{bmatrix} x_h(n_1 + 1, n_2) \\ x_v(n_1, n_2 + 1) \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ -\frac{1}{4} & 0 \end{bmatrix} \begin{bmatrix} x_h(n_1, n_2) \\ x_v(n_1, n_2) \end{bmatrix} + \begin{bmatrix} 1 \\ -\frac{1}{4} \end{bmatrix} u(n_1, n_2) \quad (5a)$$

$$y_1(n_1, n_2) = \left[-\frac{1}{4} \quad 1 \right] \begin{bmatrix} x_h(n_1, n_2) \\ x_v(n_1, n_2) \end{bmatrix} + \frac{3}{4} u(n_1, n_2) \quad (5b)$$

$$\begin{bmatrix} x_h(n_1 - 1, n_2) \\ x_v(n_1, n_2 + 1) \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ -\frac{1}{4} & 0 \end{bmatrix} \begin{bmatrix} x_h(n_1, n_2) \\ x_v(n_1, n_2) \end{bmatrix} + \begin{bmatrix} 1 \\ -\frac{1}{4} \end{bmatrix} u(n_1, n_2) \quad (6a)$$

$$y_2(n_1, n_2) = \left| \begin{bmatrix} -\frac{1}{4} & 1 \end{bmatrix} \begin{bmatrix} x_h(n_1, n_2) \\ x_v(n_1, n_2) \end{bmatrix} + \frac{3}{4} u(n_1, n_2) \right| \quad (6b)$$

where $x_h(n_1, n_2)$ and $x_v(n_1, n_2)$ are horizontal and vertical state variables respectively. The salient value can be assigned to a scanned region by combining the two directions of propagation causalities:

$$y(n_1, n_2) = y_1(n_1, n_2) + y_2(n_1, n_2) \quad (7)$$

4. Results

For line case, we split a large scene to 21×35 blocks to simulate distributed sampling and processing procedure. As shown in Figure 4, this segmentation simulates 21 robots formatted a line in n_1 axis, moving in n_2 direction and processing in 35 consecutive time slots. The simulation dataset consists of 222 large scenes of forest captured by unmanned aerial vehicle (UAVs), which meets our searching requirement of unknown environments, is referring to the work of Xu et al. [26].

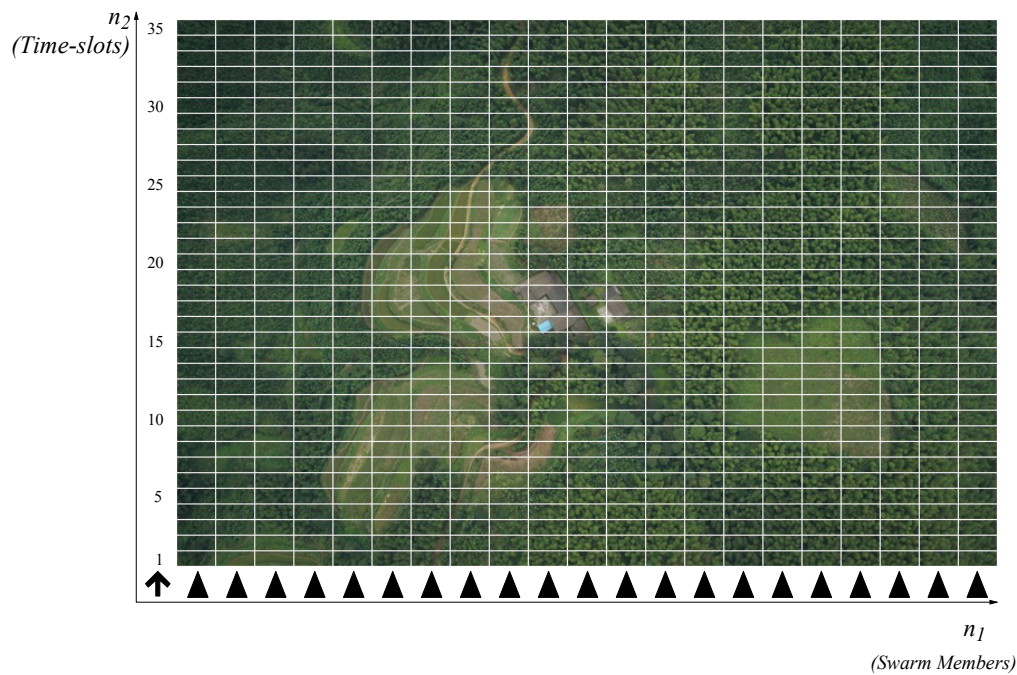


Figure 4. Simulation Configuration of Line Scan Case.

By using the average intensity feature as the input of each member and applying (5)–(7), the salient region detection results are obtained, as shown in Figure 5.

Figure 5a shows an original scene. Start from the bottom row of the split scene with zero initial condition, the line formed swarm scan the whole scene line by line to the top. The state values of each block sampled by swarm members are calculated by applying (5), and then transmitted to the neighbors on left or right depends on the propagating causalities defined before. Based on the received state values, stored previous state values and the input (average intensity of the block) of each sample node, the output of each block is obtained by combining the outputs of both causalities. Figure 5b is the region saliency map obtained by putting the outputs to the corresponding location of a gray image and resize it to the same dimension as the original scene. We applied a discrimination threshold

$\sigma = 20$ to eliminate negligible values. We can see this method can detect the blue roof of a house from the original scene since it is more salient than the surroundings.

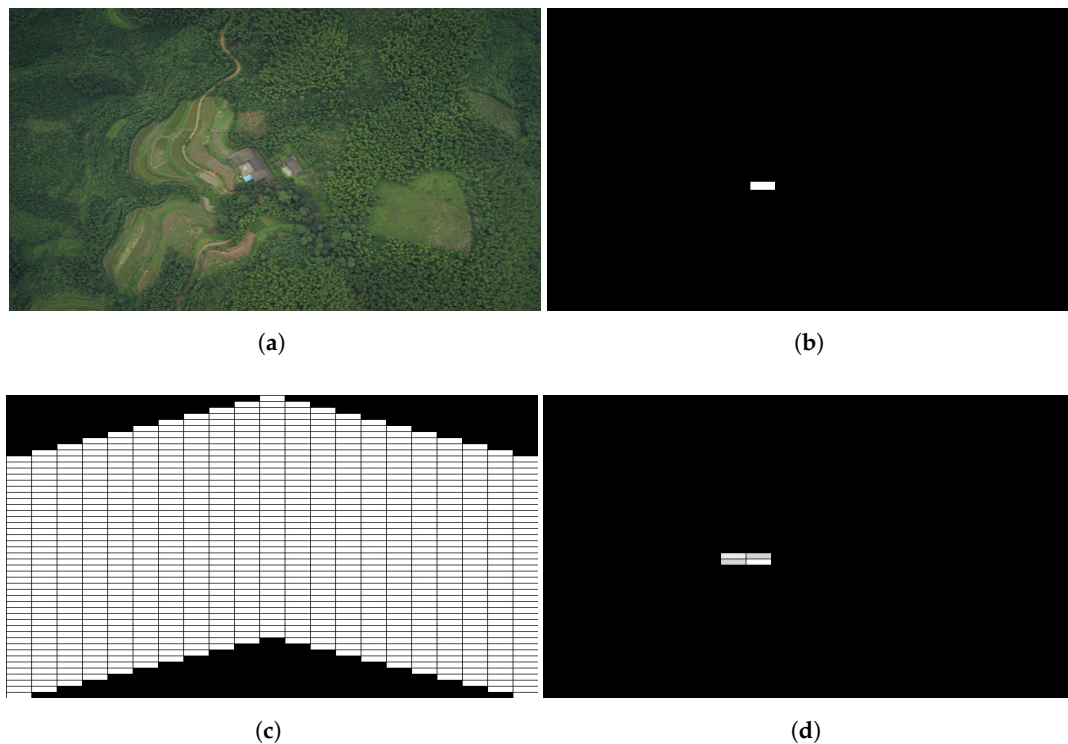


Figure 5. Simulated Distributed Salient Region Detection of Swarm Formations. (a) Original Scene; (b) Detection Result of Line Scan Case; (c) Coverage Area of V-Shape Scan Case; (d) Detection Result of V-Shape Scan Case.

The simulation procedure of distributed processing for the v-shape case is similar to line formation case. The difference is that the horizontal states propagation route is bent which causes the coverage area change. The line case is able to cover the whole scene shown in Figure 4, while the coverage of the v-shape case is shown in Figure 5c, the white blocks are the covered area. Despite the swarm in this formation case have some blind area initially, it could be ignored in a long distance scanning. The salient region detection result of the same original scene is shown in Figure 5d, we can see it still holds the similar salient region location compares to the line case. More visual results are presented in Figure 6. Figure 6a shows the original scene of our simulation. Figure 6b,d demonstrate the processing results of line case and v-shape case respectively. Figure 6c,e are shown the thresholded results. The gray blocks indicate the salient regions in the scene, and the white one indicates the most salient region.

By splitting all the images in the dataset, and running the proposed methods for line and v-shape case, we get the results of detected salient regions. With comparing with the results with manually labeled salient regions of original scenes, the statistical results of proposed methods are shown in Figure 7 for line case and Figure 8 for v-shape case respectively. Figures 7a and 8a illustrate the number of salient regions (shown with the bars) and the detecting true positives (shown with the dots) in each scene. Figures 7b and 8b give the number of false positives for line and v-shape case respectively. We can see the number of false positives in the v-shape case is higher than that in line case. Since the state propagation direction is bent, the reference blocks are not exactly left or right blocks. It makes the salient value is not the same as the line case, use the same threshold to segment the saliency map will get different results. Additionally, Figures 7c and 8c show the number of false negatives of those two situations. This number of v-shape case is also more significant than line case because there are some blind areas of v-shape scanning. With long distance scanning, the blind areas can be ignored.

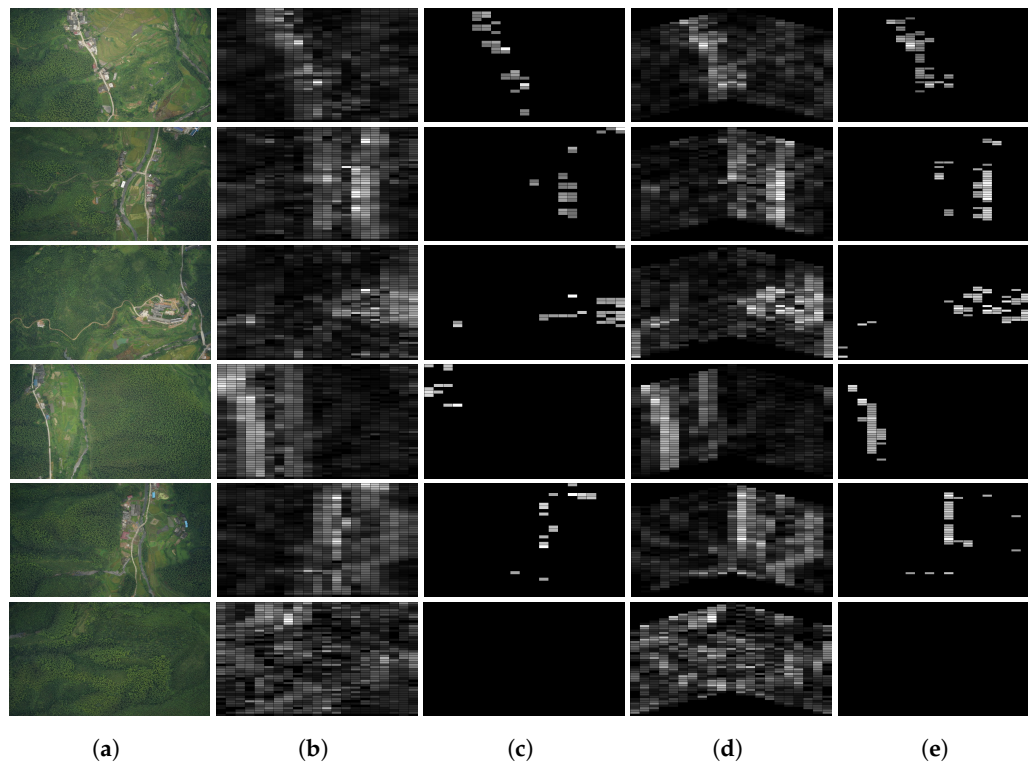


Figure 6. Formation Based Distributed Salient Region Detection. (a) Original Scene; (b) Line Case; (c) $\sigma = 20$ (Line); (d) V-shape Case; (e) $\sigma = 20$ (V-shape).

The average results of the above values are shown in Table 1. The recall rate and precision rate of line formation case is 95.77% and 88.96% respectively. Those rates are both higher than the v-shape case, which is 84.95% and 80.86% correspondingly. Since the salient region detection is typically used for rough inspection of an area, after this procedure, one may take other operations to the specific region for detail inspection. Under this consideration, those rates are acceptable for salient region detection tasks. This example evaluated proposed distributed framework is suitable for formation formed swarm tasks.

Table 1. Statistical Results.

	Line Case	V-Shape Case
No. of Scenes	222	
No. of Salient Regions	732	
True Positives (TPs)	701	621
False Positives (FPs)	87	147
False Negatives (FNs)	31	111
Recall Rate: TPs/(TPs + FNs)	95.77%	84.95%
Precision Rate: TPs/(TPs + FPs)	88.96%	80.86%

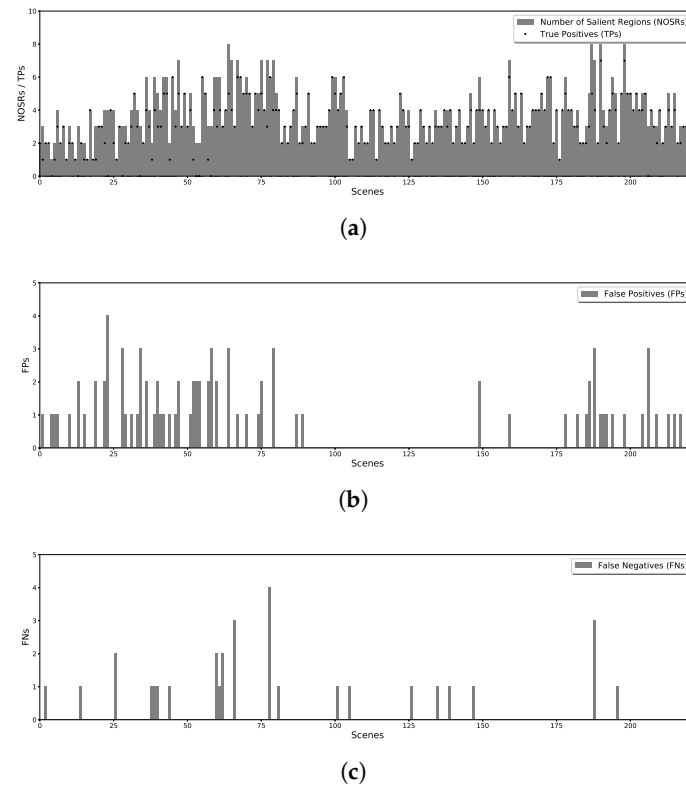


Figure 7. Statistical Results of Line Formation Case. (a) True Positives of Each Scene; (b) False Positives of Each Scene; (c) False Negatives of Each Scene.

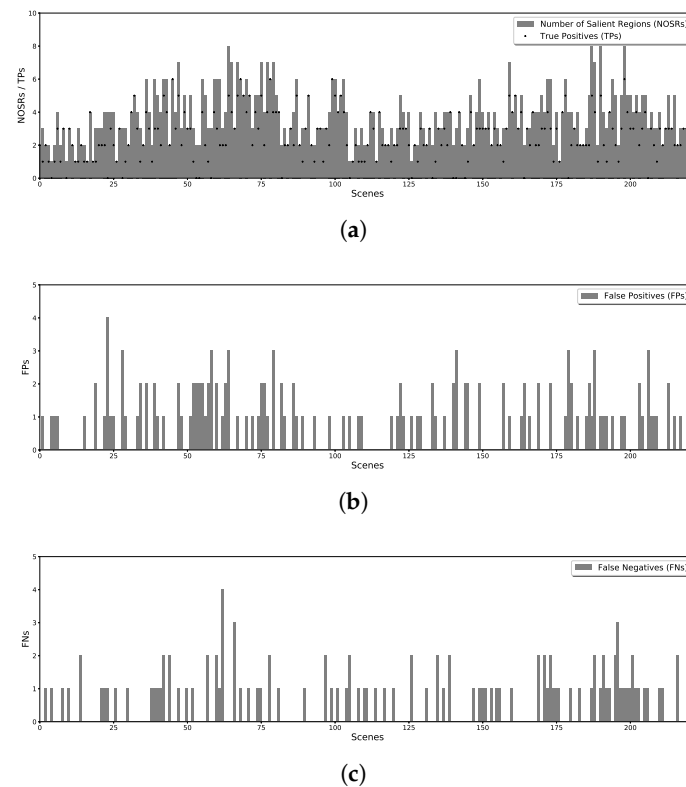


Figure 8. Statistical Results of V-shape Formation Case. (a) True Positives of Each Scene; (b) False Positives of Each Scene; (c) False Negatives of Each Scene.

5. Discussion

It can be seen from the listed results above, the presented distributed processing method for collaborative salient region detection is proved to be effective with swarms in line or v-shape formations. Here are some more questions that have to be discussed.

5.1. Unbalanced V-Shape Formation

The final shape of the v-shape formation may vary, i.e., it may result in unbalanced formations, which will influence the coverage area of the swarm. As shown in Figure 9, we tested several unbalanced v-shape cases for our distributed salient region detection algorithm. The images in the first row are the coverage areas of different unbalanced formations. The images in the second row are the corresponding detection results. Although the coverage area has changed, the detected salient region is the same in those situations. That means the overall properties of the algorithm still hold.

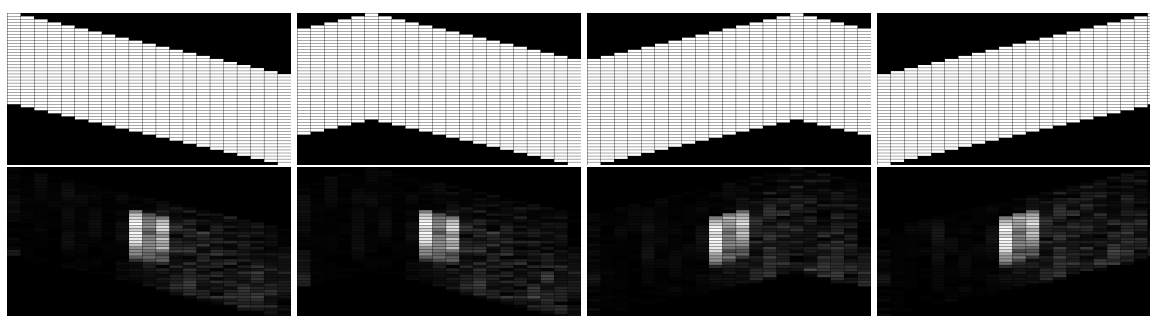


Figure 9. Distributed Salient Region Detection with Unbalanced V-shape.

5.2. Imperfect Formations

Since the sensor zero drift or controller precision of the swarm member introduces position errors, it will result in imperfect formations. Figure 10 shows an example of processing with an inaccurate v-shape, we can see since the relative positions of members are not changing during flying, the salient region is still detected. Therefore, it can still hold the effectiveness under the condition of imperfect formations with position errors. The detected region is the same as in Figure 9.

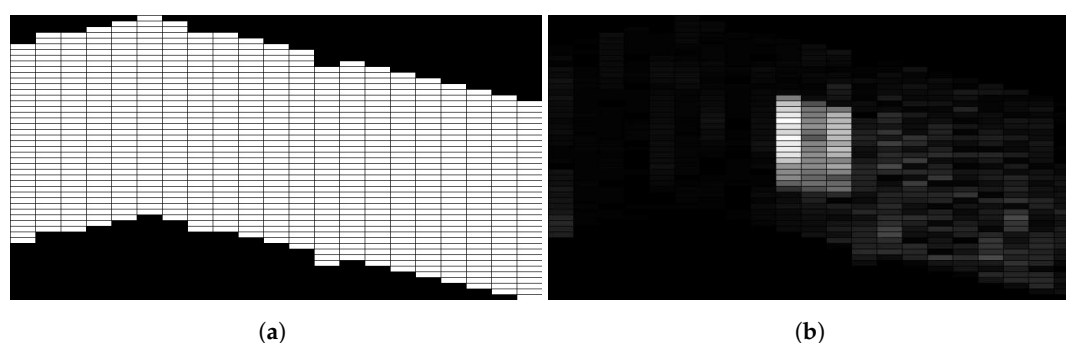


Figure 10. Results with Imperfect V-Shape Scan. (a) Imperfect V-shape Scan Coverage; (b) Processing Results.

5.3. Other Concerns

The proposed processing method is based on spatial related multidimensional system theory. The relative positions of members in the swarm are essential for processing. In the initial or transient phase of pattern formation, if the relative positions are totally disordered, the output of the method will be invalid unless the positions of every member in the swarm are recorded, and some recovery

algorithms are applied. We can use other strategies to avoid this case, for example, forming the formation using a slower speed before reaching the interest area, or revisit the area after formation, etc. Furthermore, the proposed example of salient region detection we have shown is only able to detect static or slowly moving salient targets. Nevertheless, by applying other filters (e.g., velocity filter [27]) to the distributed processing framework, the introduced Roesser model based distributed processing paradigm has the capability to detect other events or dynamically changing signals.

6. Conclusions

This paper introduces a new distributed processing paradigm for formation formed robotic swarms in searching tasks. The in-swarm distributed processing is performed using a modified 2-D Roesser model with half-plan causality. The concept is illustrated using a line (or v-shape) scanning swarm that implements the 2-D algorithm sequentially, i.e., by executing the 2-D filtering algorithm line by line. The effectiveness of this framework is shown by an example of salient region detection task, making it a good candidate for collaborative sensing of robotic swarms.

Author Contributions: Conceptualization, J.Y. and P.B.; Methodology, J.Y.; Software, J.Y.; Validation, X.W., P.B. and J.Y.; Investigation, J.Y., X.W. and P.B.; Resources, X.W.; Data Curation, J.Y.; Writing—Original Draft Preparation, J.Y.; Writing—Review & Editing, P.B.; Visualization, J.Y.; Supervision, X.W.; Project Administration, X.W.; Funding Acquisition, X.W.

Funding: This research was funded by Shenzhen Science and Technology Innovation Commission grant number JCYJ20170413110656460 and JCYJ20150403161923545.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Sousselier, T.; Dreio, J.; Sevaux, M. Line formation algorithm in a swarm of reactive robots constrained by underwater environment. *Expert Syst. Appl.* **2015**, *42*, 5117–5127. [[CrossRef](#)]
2. Ducatelle, F.; Di Caro, G.A.; Pinciroli, C.; Gambardella, L.M. Self-organized cooperation between robotic swarms. *Swarm Intell.* **2011**, *5*, 73. [[CrossRef](#)]
3. Bayındır, L. A review of swarm robotics tasks. *Neurocomputing* **2016**, *172*, 292–321. [[CrossRef](#)]
4. Hylsberg Jacobsen, R.; Zhang, Q.; Skjodeberg Toftegaard, T. Bioinspired principles for large-scale networked sensor systems: An overview. *Sensors* **2011**, *11*, 4137–4151. [[CrossRef](#)] [[PubMed](#)]
5. Du, H.; Xia, N.; Zheng, R. Particle swarm inspired underwater sensor self-deployment. *Sensors* **2014**, *14*, 15262–15281. [[CrossRef](#)] [[PubMed](#)]
6. Scheutz, M.; Bauer, P. Ultra-Low Complexity Control Mechanisms for Sensor Networks and Robotic Swarms. *Int. J. New Comput. Archit. Appl.* **2013**, *3*, 86–119.
7. Thenius, R.; Moser, D.; Varughese, J.C.; Kernbach, S.; Kuksin, I.; Kernbach, O.; Kuksina, E.; Mišković, N.; Bogdan, S.; Petrović, T.; et al. subCULTron-Cultural Development as a Tool in Underwater Robotics. In *Artificial Life and Intelligent Agents Symposium*; Springer: Cham, Switzerland, 2016; pp. 27–41.
8. Parker, L.E.; Rus, D.; Sukhatme, G.S. Multiple Mobile Robot Systems. In *Springer Handbook of Robotics*; Springer: Cham, Switzerland, 2016; pp. 1335–1384.
9. Trianni, V.; Campo, A. Fundamental collective behaviors in swarm robotics. In *Springer Handbook of Computational Intelligence*; Springer: Berlin/Heidelberg, Germany, 2015; pp. 1377–1394.
10. Kennedy, J. Particle swarm optimization. In *Encyclopedia of Machine Learning*; Springer: Boston, MA, USA, 2011; pp. 760–766.
11. Wang, J.; Wang, D. Particle swarm optimization with a leader and followers. *Progress Nat. Sci.* **2008**, *18*, 1437–1443. [[CrossRef](#)]
12. Burgard, W.; Moors, M.; Fox, D.; Simmons, R.; Thrun, S. Collaborative multi-robot exploration. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA 2000), San Francisco, CA, USA, 24–28 April 2000; Volume 1, pp. 476–481.
13. Alur, R.; Das, A.; Esposito, J.; Fierro, R.; Grudic, G.; Hur, Y.; Kumar, V.; Lee, I.; Ostrowski, J.; Pappas, G.; et al. A framework and architecture for multirobot coordination. In *Experimental Robotics VII*; Springer: Berlin/Heidelberg, Germany, 2001; pp. 303–312.

14. Askari, A.; Mortazavi, M.; Talebi, H. UAV formation control via the virtual structure approach. *J. Aerosp. Eng.* **2013**, *28*, 04014047. [[CrossRef](#)]
15. Bayindir, L.; Sahin, E. Modeling self-organized aggregation in swarm robotic systems. In Proceedings of the 2009 IEEE Swarm Intelligence Symposium, Nashville, TN, USA, 30 March–2 April 2009; pp. 88–95.
16. Balch, T.; Hybinette, M. Social potentials for scalable multi-robot formations. In Proceedings of the IEEE International Conference on Robotics & Automation, San Francisco, CA, USA, 24–28 April 2000; Volume 1, pp. 73–80.
17. Ren, W.; Beard, R.W. Consensus seeking in multiagent systems under dynamically changing interaction topologies. *IEEE Trans. Autom. Control* **2005**, *50*, 655–661. [[CrossRef](#)]
18. Oh, H.; Shirazi, A.R.; Sun, C.; Jin, Y. Bio-inspired self-organising multi-robot pattern formation: A review. *Robot. Auton. Syst.* **2017**, *91*, 83–100. [[CrossRef](#)]
19. Nathan, A.; Barbosa, V.C. V-like formations in flocks of artificial birds. *Artif. Life* **2008**, *14*, 179–188. [[CrossRef](#)] [[PubMed](#)]
20. Li, X.; Tan, Y.; Fu, J.; Mareels, I. On V-shaped flight formation of bird flocks with visual communication constraints. In Proceedings of the 13th IEEE International Conference on Control & Automation (ICCA), Ohrid, Macedonia, 3–6 July 2017; pp. 513–518.
21. Yang, J.; Wang, X.; Bauer, P. Formation forming based low-complexity swarms with distributed processing for decision making and resource allocation. In Proceedings of the 14th International Conference on Control, Automation, Robotics and Vision (ICARCV), Phuket, Thailand, 13–15 November 2016; pp. 1–6.
22. Sumanasena, B.; Bauer, P.H. Realization using the Roesser model for implementations in distributed grid sensor networks. *Multidimens. Syst. Signal Process.* **2011**, *22*, 131–146. [[CrossRef](#)]
23. Roesser, R.P. A discrete state-space model for linear image processing. *IEEE Trans. Autom. Control* **1975**, *20*, 1–10. [[CrossRef](#)]
24. Itti, L.; Koch, C. Computational modelling of visual attention. *Nat. Rev. Neurosci.* **2001**, *2*, 194–203. [[CrossRef](#)] [[PubMed](#)]
25. Rogers, E.; Galkowski, K.; Paszke, W.; Moore, K.; Bauer, P.; Hladowski, L.; Dabkowski, P. Multidimensional control systems: Case studies in design and evaluation. *Multidimens. Syst. Signal Process.* **2015**, *26*, 895–939. [[CrossRef](#)]
26. Xu, Y.; Ou, J.; He, H.; Zhang, X.; Mills, J. Mosaicking of unmanned aerial vehicle imagery in the absence of camera poses. *Remote Sens.* **2016**, *8*, 204. [[CrossRef](#)]
27. Schwerdtfeger, T.; Velten, J.; Kummert, A. A multidimensional wave digital filter bank for video-based motion analysis. *Multidimens. Syst. Signal Process.* **2014**, *25*, 295–311. [[CrossRef](#)]



© 2018 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).