



## Faculty Publications

---

2008-12-01

# Decentralized Perimeter Surveillance Using a Team of UAVs

Randal Beard  
beard@byu.edu

David Casbeer

Derek Kingston

Follow this and additional works at: <https://scholarsarchive.byu.edu/facpub>



Part of the [Electrical and Computer Engineering Commons](#)

## Original Publication Citation

Derek Kingston, Randal W. Beard, and Ryan S. Holt. "Decentralized Perimeter Surveillance Using a Team of UAVs," IEEE Transactions on Robotics, Volume 24, Issue 6, 18 November 2008.

## BYU ScholarsArchive Citation

Beard, Randal; Casbeer, David; and Kingston, Derek, "Decentralized Perimeter Surveillance Using a Team of UAVs" (2008). *Faculty Publications*. 1224.  
<https://scholarsarchive.byu.edu/facpub/1224>

This Peer-Reviewed Article is brought to you for free and open access by BYU ScholarsArchive. It has been accepted for inclusion in Faculty Publications by an authorized administrator of BYU ScholarsArchive. For more information, please contact [ellen\\_amatangelo@byu.edu](mailto:ellen_amatangelo@byu.edu).

# Decentralized Perimeter Surveillance Using a Team of UAVs

Derek B. Kingston

Randal W. Beard\*

David W. Casbeer

Timothy W. McLain

*Brigham Young University, Provo, UT, 84602*

**This paper develops a distributed algorithm to maintain a current estimate of the state of the perimeter using a team of UAVs. Using notions of consensus, an algorithm will be developed and shown to distribute a UAV team uniformly around the perimeter.**

## Nomenclature

$N$	Number of UAVs on team
$P$	Length of perimeter
$\xi_i$	Size of perimeter segment to be monitored by the $i^{\text{th}}$ UAV
$\mathcal{G}$	Directed graph representing communication topology
$\bar{\mathcal{G}}$	Set of all possible simple interaction graphs

## I. Introduction

This paper develops a decentralized multiple-UAV approach to perimeter surveillance. One important application of perimeter surveillance is tracking the movement of the perimeter of a forest fire.<sup>1,2</sup> In general, there are two basic perimeter topologies: circular (no specific endpoints) and linear (UAVs double-back at endpoints). Most of our work has been focused on circular perimeters, but any perimeter topologically close to a line or circle can easily be accommodated in the same framework.

Perimeter surveillance is the process of gathering data at all points of the perimeter and transmitting that data to a base station for analysis. The state of the perimeter will be most accurate – with respect to latency – if UAVs are spread uniformly along the perimeter in both directions of travel.<sup>3</sup> Of course, if some areas of the perimeter are more important than others, the spacing of the UAVs should not be uniform, but should be concentrated at the areas of interest. For our work, we assume that all points along the perimeter have the same value, and so, we aim to find a distributed algorithm that allows the UAVs to constantly stream gathered data back to the base station while at the same time maintaining uniform spacing. We also assume that data can only be transmitted when a UAV is near the base station or other UAVs. In this way, data gathered far from the base station must be taken by a UAV (or a chain of UAVs) back to the base station for transmission.

A key challenge in implementing decentralized cooperation strategies is to form consensus among members of the team when communication links are intermittent or noisy and sensed information is inconsistent among team members. Recent work on consensus algorithms provide a means for convergence to consistent cooperation information among team members. In [4], sufficient conditions are given for consensus of the heading angles of a group of agents under undirected switching interaction topologies. In [5], average consensus problems are solved for a network of integrators using directed graphs. In [6], a set-valued Lyapunov approach is used to consider consensus problems with unidirectional time-dependent communication links. Using directed graphs, Refs. [7] and [8] show necessary and/or sufficient conditions for consensus of information under time-invariant and switching interaction topologies. We will use the notion of consensus to ensure that the team of UAVs tasked to monitor a perimeter will agree on the length of perimeter segment that each member will survey.

In this paper we present a multiple UAV cooperative control solution to the perimeter surveillance problem. In Section III we introduce consensus algorithms as a way to guarantee the uniform spread of the team. Section IV

---

\*Corresponding author: [beard@ee.byu.edu](mailto:beard@ee.byu.edu)

gives an algorithm with an embedded consensus method that overcomes the initial uncertainties in the system. The algorithm will be shown to overcome arbitrary initial conditions, and so, allows for such features as: the ability to systematically add and remove UAVs from the team (important for re-fueling) and the ability to supply perimeter state information along the entire perimeter length.

## II. Problem Statement

Consider the task of designing a flight pattern for a team of UAVs to monitor phenomena along a perimeter or border. We desire information about all points along the perimeter to be transmitted to a central location for analysis. Due to limited communication range of the UAVs, transmission of gathered data can only occur when a UAV is close to the base station or other UAVs. We assume that, generally, UAVs are not in communication with any other member of the team or the base station, necessitating a distributed algorithm.

A flight pattern where  $N/2$  UAVs are traveling clockwise and  $N/2$  are traveling counter-clockwise at constant velocity allows for data regarding the entire perimeter to be gathered and transmitted to a location on the perimeter. When the UAVs are spaced evenly along the perimeter in both directions and if UAVs update their copy of the perimeter state each time another team member is met, the state of the perimeter as seen at the base station is updated with the most current information.<sup>3</sup> Note that UAVs could just as easily reverse direction after rendezvous with other team members and the data flow to the base station would remain the same. In this way, each team member would be responsible for monitoring a segment of the perimeter. When all of the segments are of equal length, the ideal data flow conditions are achieved.

We will develop a distributed algorithm that allows a team of UAVs to evenly spread itself around the perimeter. We will assume that UAVs fly at constant speed and can reverse direction instantaneously – this simplifies the analysis and for large perimeters, is a good approximation. UAVs are assumed to either travel along the perimeter with constant velocity or loiter at a specific point on the perimeter.

### Example

Figure 1 shows an example illustrating how a team of UAVs can adjust to spread out evenly along the perimeter of the fire. Two pairs of UAVs are launched with the first pair leading the second by two time units (Figures 1(a) and 1(b)). After four time units have passed, the first pair of UAVs meet (Figure 1(c)). Due to the distributed nature of the problem, neither is aware of the trailing pair of UAVs. For this reason, both UAVs determine to rendezvous again back at the base station. After the first pair rendezvous and double-back, the second pair meets the first pair at  $t = 5$  (Figure 1(d)). By this time, UAVs 1 and 2 have traveled  $\frac{1}{8}P$  and UAVs 3 and 4 have traveled  $\frac{3}{8}P$  from their last rendezvous. Since the first pair has traveled the least, they will decide to loiter after traveling  $\frac{1}{4}P$  from their next rendezvous. Both pairs leave the rendezvous location in the opposite direction in which they entered. At  $t = 6$ , the first pair meets again (Figure 1(e)). Since both have traveled equal distances from their previous rendezvous (with the other pair), then neither will loiter after the next rendezvous. However, due to the interruption by the other pair of UAVs, a distance of at most  $\frac{1}{4}P$  will be traveled before beginning to loiter. At  $t = 8$ , the first pair begins loitering and the second pair meets at the base station (Figure 1(f)). UAV 3 and 4 have both traveled the same distance, so neither will loiter after the next rendezvous. Figure 1(g) shows the time when the two pairs of UAVs meet and have traveled equal distance. Thereafter, the UAVs are in the optimal configuration to mini-

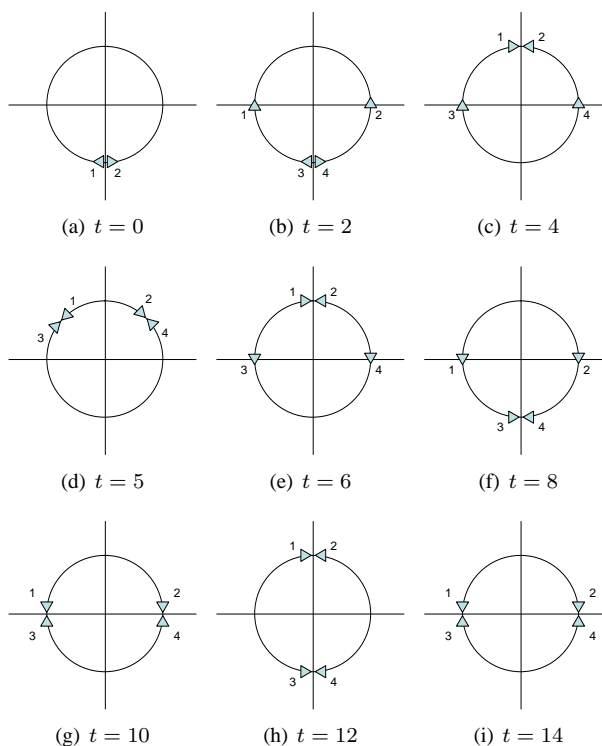


Figure 1. Example scenario in which UAV spread is adjusted to evenly space the UAVs around the perimeter

mize overall latency and provide the fastest possible update rate. This pattern is maintained due to the fact that each UAV will travel exactly  $\frac{1}{4}P$  between rendezvous and none will begin to loiter (Figures 1(h) and 1(i)).

### III. Consensus

At the heart of distributed perimeter surveillance, UAVs must negotiate with others on the team as to how much of the perimeter each is responsible for monitoring. A consensus algorithm can be used to ensure that each UAV has an equal portion and that the entire perimeter is being monitored. Consensus methods give necessary conditions to ensure that as  $t \rightarrow \infty$ , each member of the team approaches the same value of the *information variable*.<sup>9</sup> For perimeter surveillance, the *information variable* is the length of the segment that each UAV should monitor. If the UAVs are truly spread uniformly along the perimeter, then each will have the same length of segment to monitor and the sum of those lengths will be the entire perimeter length. General consensus methods only ensure that the team will eventually agree – they do not specify the value that will be agreed upon. For this reason, we give additional conditions, that when met, will guarantee not only consensus among team members, but also that the value of convergence will be  $P/N$ .

The main result from [7, 8] for discrete-time consensus is the following theorem.

**Theorem III.1** *Let  $\mathcal{G}[k] \in \bar{\mathcal{G}}$  be a switching interaction graph at time  $t = kT$ . Also let  $\alpha_{ij}[k] \in \bar{\alpha}$ , where  $\bar{\alpha}$  is a finite set of arbitrary positive numbers. The discrete update scheme*

$$\xi_i[k+1] = \frac{1}{\sum_{j=1}^n \alpha_{ij}[k] G_{ij}[k]} \sum_{j=1}^n \alpha_{ij}[k] G_{ij}[k] \xi_j[k], \quad (1)$$

*achieves consensus asymptotically for all  $N$  agents if there exists an infinite sequence of uniformly bounded, non-overlapping time intervals  $[k_j T, (k_j + l_j) T)$ ,  $j = 1, 2, \dots$ , starting at  $k_1 = 0$ , with the property that each interval  $[(k_j + l_j) T, k_{j+1} T)$  is uniformly bounded and the union of the graphs across each interval  $[(k_j + l_j) T, k_{j+1} T)$  has a spanning tree.*

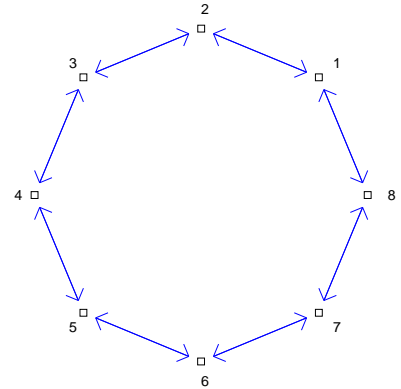
Theorem III.1 states that if a weighted averaging scheme is used (Equation (1)) and the union of all communication graphs has a spanning tree, then the *information variable*,  $\xi$ , for each team member will eventually come into agreement, i.e. consensus will be reached. The flexibility of Theorem III.1 is in the condition that only the *union* of all communication graphs must have a spanning tree. This means that even if team members only communicate with one other member of the team, as long as the team communication graph (the union of all the single communication links) has a spanning tree, consensus will be achieved.

Consider the communication topology where each team member communicates only with its two neighbors on the perimeter. Communication will likely occur with only one neighbor at a time, but the union of the communication graphs will be a complete bi-directional ring as shown in Figure 2. A bi-directional ring obviously has a spanning tree, and so by Theorem III.1 we conclude that team members will eventually come into consensus.

To ensure that the team members come into consensus and that the value to which they converge is  $P/N$ , consider the following lemma.

**Lemma III.2** *A team of agents will form consensus with the property that for each agent  $i$ ,  $\lim_{t \rightarrow \infty} \xi_i(t) = 1/N$  if the following are true: (1) all agents are equally weighted (i.e.  $\alpha_{ij} = 1$  for all  $i, j$  in Equation (1)), (2)  $\sum \xi_{i_0} = 1$ , (3) the union of all communication graphs has a spanning tree, and (4) all communication is bi-directional (i.e. if one agent updates its estimate of  $\xi_i$  then the corresponding agents who communicated with it also update their respective values of  $\xi_i$ ).*

*Proof:* When (3) is satisfied, Theorem III.1 shows that the team will achieve consensus, i.e.  $\lim_{t \rightarrow \infty} \xi_i(t) = c \forall i$ . The combination of conditions (1) and (4) ensure that  $\sum \xi_i$  remains constant. This can be seen by noting that Equation (1) with  $\alpha_{ij} = 1$  will assign each team member in communication the precise average of all the values present during the communication event. Since by condition (2) the initial value of  $\sum \xi_i$  is 1, then the value of  $\lim_{t \rightarrow \infty} \xi_i(t)$  must be  $1/N$  for each agent  $i$ . ■



**Figure 2. Bi-directional ring communication topology for 8 agents.**

## IV. A Distributed Perimeter Surveillance Algorithm

Each UAV in the perimeter surveillance team begins operation without an estimate of perimeter length or number of UAVs on the team. The goal of the team is twofold: spread out uniformly and ensure the most rapid transmission of gathered data back to the base station. The perimeter will be monitored in the most efficient way if team members are spread uniformly in both directions of travel.

Developing a distributed algorithm that causes the members of the UAV team to cover equal lengths of the perimeter is complicated by the initial locations of the UAVs. In addition, the algorithm must adjust the rendezvous locations between two neighboring UAVs without knowledge of the true perimeter length. A distributed algorithm has been developed to accomplish this objective. It operates under the assumption that pairs of UAVs are simultaneously launched in opposite directions from the base station.

### A. Algorithm

A few definitions are needed before presenting the algorithm. Note that this algorithm is identical for each member of the team, so all variables defined are local to the vehicle running the algorithm. The variable  $\ell_c$  is the distance traveled since the last rendezvous and  $\ell_n$  is the corresponding value received during a communication event with a neighbor. This variable is updated by constantly integrating the distance traveled and is assumed to be updated for the algorithm.  $\ell_p$  is the previous distance traveled between rendezvous (*i.e.* from the previous rendezvous to the last rendezvous). Let  $dir$  indicate the current direction of travel around the perimeter either clockwise or counter-clockwise. Each UAV holds two variables for adjusting rendezvous locations,  $L_{cw}$  and  $L_{ccw}$ .  $L_{cw}$  is used to specify the distance from the last rendezvous the UAV should travel before it begins loitering if it is headed in a clockwise direction, similarly  $L_{ccw}$  is the distance to begin loitering from the last rendezvous if the UAV is traveling in the counter-clockwise direction. To ensure rendezvous between UAVs, we let  $L_{dir} = \infty$  so the UAV will not loiter, rather it will continue traveling until it rendezvous with its neighbor. Lastly  $\ell$  is calculated for each rendezvous by averaging neighbors'  $\ell_c$ . It is the distance that must be traveled after the next rendezvous. The algorithm is shown below.

---

#### Algorithm 1: Distributed spread

---

$\ell_c$  is the distance traveled (constantly integrated outside the algorithm)

**Initialize** (when UAV first arrives at perimeter)

$\ell_c = 0$  distance traveled since last rendezvous

$\ell_p = 0$  distance traveled between last two rendezvous

To ensure a rendezvous we set distance to loiter to  $L_{cw} = \infty$ ,  $L_{ccw} = \infty$

$dir$  = the direction of launch for this UAV either cw or ccw

**while 1 do**

- 1 **if**  $\ell_c \geq L_{dir}$  **then**  
     Wait here for next rendezvous)  
      $dir = 0$  (no direction)
  - 2 **if** rendezvous (*i.e.* my neighbor is within communication range) **then**
  - 3     **if** (*this is my first rendezvous*) **then**  $\ell_p = \ell_c$
  - 4     Send  $\ell_c$  to neighbor
  - 5      $\ell_n =$ (the  $\ell_c$  received from neighbor)
  - 6     **if** ( $dir = 0$ ) **then**  $dir =$  opposite of neighbor's  $dir$
  - 7     Average the distance traveled since the UAV's last rendezvous  
      $\ell = \frac{1}{2}(\ell_c + \ell_n)$
  - 8     Determine who should loiter for this pair's next rendezvous  
     **if**  $\ell_c < \ell_n$  (*I traveled less*) **then**  $L_{dir} = \ell$  (I should loiter)  
     **else if**  $\ell_c > \ell_n$  (*I traveled more*) **then**  $L_{dir} = \infty$  (neighbor will loiter)  
     **else** ( $\ell_c = \ell_n$  *equal distance traveled*)  $L_{dir} = \infty$  (neither will loiter)
  - 9     Reverse  $dir$  (For example: if ccw make it cw)
  - 10    Calculate distance to loiter for next rendezvous  $L_{dir} = L_{dir} + (\ell_c - \ell_p)$
  - 11     $\ell_p = \ell_c$ ,  $\ell_c = 0$
-

The initialization of the algorithm occurs when the UAVs first reach the perimeter. At this time, the UAV defaults to continue until the next rendezvous and not loiter. This is done by setting the distance to loiter in both directions to infinity ( $L_{cw} = \infty$ ,  $L_{ccw} = \infty$ ). In general operation, when two UAVs rendezvous, they communicate the distance they have traveled since the last rendezvous to their neighbor (step 4). If the distances traveled do not coincide, then the UAV that has traveled the shorter length (smaller  $\ell_c$ ) is required to adjust its distance to loiter so that when this pair meets again for their next rendezvous the UAV that traveled the shorter distance will be the one to wait at the designated location. This adjustment is given to the UAV that traveled the shorter distance because in most cases it will arrive at the designated location first. Notice that the actual distance to loiter for a given direction is calculated from the average of the distances traveled (step 7) and is set only after a rendezvous has occurred in the opposite direction (step 10).

To explain why the distance to loiter must be set after the next rendezvous the following example is provided. Let the current and past rendezvous be called  $r_0$  and  $r_{-1}$  respectively. Also, label the next two rendezvous as  $r_1$  and  $r_2$ . At  $r_0$  and  $r_2$  the same UAVs will meet. The distance to loiter,  $\ell$  from  $r_1$  to  $r_2$  assumes that the rendezvous at  $r_{-1}$  and  $r_1$  occurred at the same location. However, this may not always be the case due to initial conditions. Therefore the term  $\ell_c - \ell_p$  in step 10 adjusts the distance to loiter to account for this discrepancy.

## B. Analysis

In this section we show that Algorithm 1 converges to a configuration where all of the UAVs are equally spaced around the perimeter. To do so, we will show that the conditions of Lemma III.2 are satisfied after enough time has passed and that the UAVs are able to adjust their actual monitored perimeter length to match the desired perimeter length given by the consensus algorithm.

**Theorem IV.1** *Consider a fixed-length circular perimeter with length  $P$ . When UAVs are launched in pairs, with one UAV in each pair heading clockwise and the other, counter-clockwise, Algorithm 1 will ensure that as  $t \rightarrow \infty$  the team of UAVs will be equally spaced around the perimeter.*

*Proof:* By the geometry of the perimeter, each UAV communicates only with its two neighbors, so a bi-directional ring communication topology is present, satisfying condition (3) of Lemma III.2. UAVs are assumed to be able to communicate without fault when near enough to each other. A rendezvous occurs when two UAVs are mathematically at the same place, therefore communication is assumed to always be bi-directional which satisfies condition (4).

Observe that the two-agent form of Equation (1) is present in step 7 of Algorithm 1. In fact, there are two consensus algorithms embedded into Algorithm 1; one for the clockwise direction and the other for the counter-clockwise direction. The *information variable* for each instance of the consensus algorithms is  $L_{dir}$ . To satisfy condition (1) of Lemma III.2, observe that  $L_{dir}[k-1] = \ell_c + \nu$  where  $\nu = \ell_c - \ell_p$  acts as a disturbance to the system and is a measure of how far out of sync the two directions are. Theorem III.1 has been shown to be ISS,<sup>10</sup> so as time progresses, both directions approach the same value and the disturbance goes to zero.

To invoke Lemma III.2, we also need to show that the sum of the monitored lengths equals the perimeter length at some time. Observe that Algorithm 1 only has memory of the previous and current rendezvous. An addition of a pair of UAVs to the system will cause the overall sum of the monitored lengths to exceed  $P$ , but since no two UAVs cover the same section of the perimeter, the effect of the addition will be nullified after each pair of UAVs has rendezvoused twice after the disturbance. Deletion of a pair of UAVs has similar effect, with UAVs adjusting to monitor the section left unmonitored by always ensuring that one of the pair from the previous rendezvous continues indefinitely until the next rendezvous (see step 8).

As shown above, Algorithm 1 ensures that, after some time, all the conditions of Lemma III.2 are satisfied, at which point, the embedded consensus algorithm guarantees by Lemma III.2 that the UAVs will converge to a uniformly spread configuration. Since the effects of initial conditions die out as time progresses, pairs of UAVs may enter or exit the team for refueling without affecting the long term stability of the algorithm.

## V. Conclusions

A distributed algorithm has been presented which modifies the configuration of a team of UAVs to match the ideal pattern for data transmission by uniformly spreading the UAVs along the perimeter. Convergence to a uniform distribution was guaranteed by application of consensus theory.

## Acknowledgments

We would like to thank Scott Poll, Chad Frost, and Francis Enomoto at NASA Ames Research Center for initiating this research problem and pointing out related work in this area. This research was supported by NASA under STTR contract No. NNA04AA19C to Scientific Systems Company, Inc (SSCI) and Brigham Young University (BYU), and by AFOSR grants F49620-01-1-0091 and F49620-02-C-0094.

## References

- <sup>1</sup>“White Paper on UAV Over-the-Horizon Disaster Management Demonstration Projects,” Feb 2000, <http://geo.arc.nasa.gov/sgc/UAVFiRE/whitepaper.html>.
- <sup>2</sup>Casbeer, D. W., Beard, R. W., McLain, T. W., Li, S.-M., and Mehra, R. K., “Forest Fire Monitoring Using Multiple Small Unmanned Air Vehicles,” *Proc. of the American Control Conference*, 2005, (to appear).
- <sup>3</sup>Casbeer, D. W., Kingston, D. B., Beard, R. W., McLain, T. W., Li, S.-M., and Mehra, R., “Cooperative Forest Fire Surveillance Using a Team of Small Unmanned Air Vehicles,” *International Journal of System Sciences*, (in review), Technical Report available at <https://dspace.byu.edu/handle/1877/55>.
- <sup>4</sup>Jadbabaie, A., Lin, J., and Morse, A. S., “Coordination of Groups of Mobile Autonomous Agents Using Nearest Neighbor Rules,” *IEEE Transactions on Automatic Control*, Vol. 48, No. 6, June 2003, pp. 988–1001.
- <sup>5</sup>Saber, R. O. and Murray, R. M., “Agreement Problems in Networks with Directed Graphs and Switching Topology,” *Proceedings of the IEEE Conference on Decision and Control*, Maui, Hawaii, December 2003, pp. 4126–4132.
- <sup>6</sup>Moreau, L., “Leaderless Coordination via Bidirectional and Unidirectional Time-dependent Communication,” *Proceedings of the IEEE Conference on Decision and Control*, Maui, Hawaii, December 2003, pp. 3070–3075.
- <sup>7</sup>Ren, W. and Beard, R. W., “Consensus Seeking in Multi-agent Systems Under Dynamically Changing Interaction Topologies,” *IEEE Transactions on Automatic Control*, (to appear).
- <sup>8</sup>Ren, W., Beard, R. W., and McLain, T. W., “Coordination Variables and Consensus Building in Multiple Vehicle Systems,” *Cooperative Control*, edited by V. Kumar, N. Leonard, and A. S. Morse, Vol. 309 of *Lecture Notes in Control and Information Systems*, Springer-Verlag, 2004, pp. 171–188.
- <sup>9</sup>McLain, T. W. and Beard, R. W., “Coordination Variables, Coordination Functions, and Cooperative Timing Missions,” *AIAA Journal of Guidance, Control, and Dynamics*, 2004, (to appear).
- <sup>10</sup>Kingston, D. B., Ren, W., and Beard, R. W., “Consensus Algorithms are Input-to-State Stable,” *Proc. of the American Control Conference*, 2005, (to appear).