

**Paper Review: Ground-level Mapping And Navigating
for Agriculture Based on IoT And Computer Vision**

Name: Tasnim Fuyara Chhoan

ID: 23366035; **CSE 707** (Fall'23)

Submitted to- Annajiat Alim Rasel (**AAR**)

Summary

**Motivation/purpose/
aims/hypothesis**

Contribution

Methodology

Conclusion

Summary

**Motivation/purpose/
aims/hypothesis**

Contribution

Methodology

Conclusion

Summary

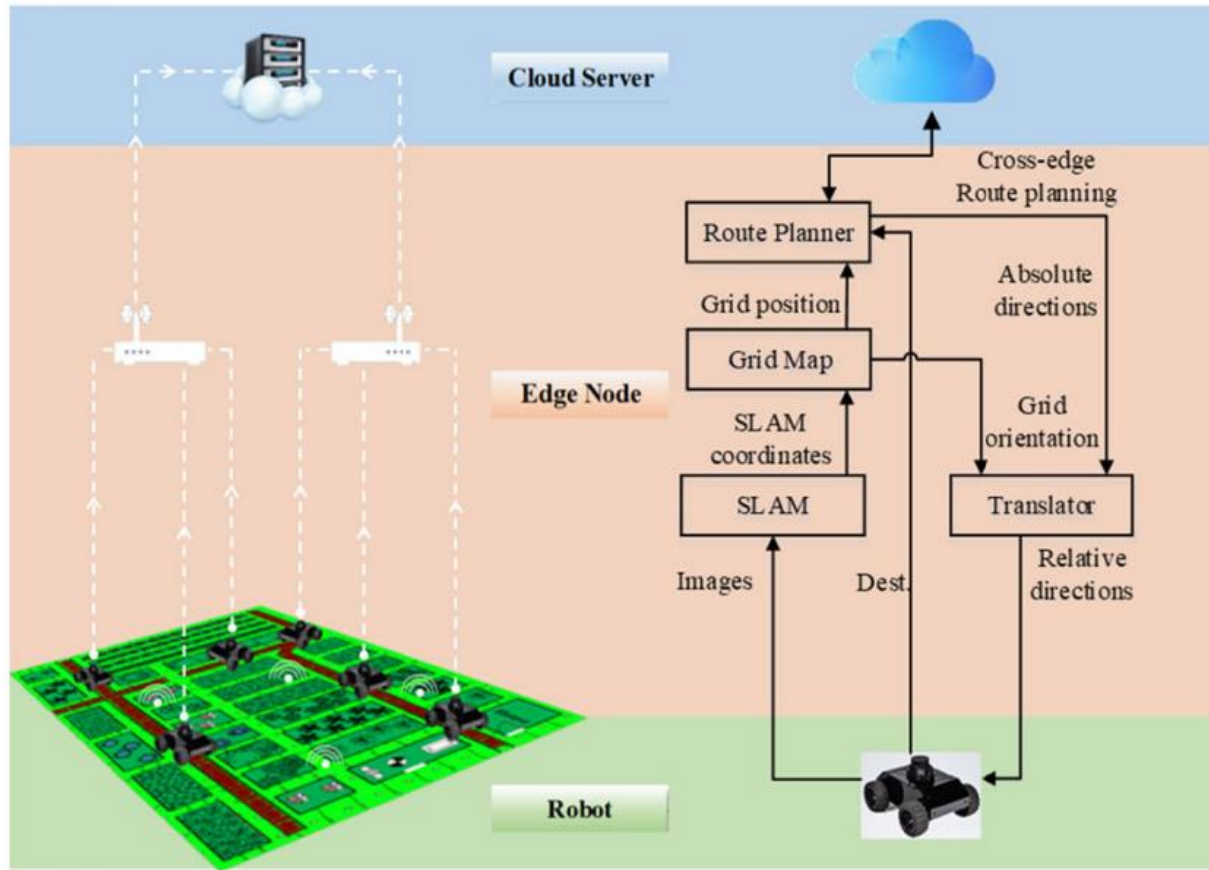


FIGURE 1. The IoT architecture of Cloud-Edge-Robot.

Contribution

Methodology

Conclusion

Summary

Motivation/purpose/ aims/hypothesis

- Enhance agriculture mapping
- Combine IoT and computer vision
- IoT benefits precision agriculture

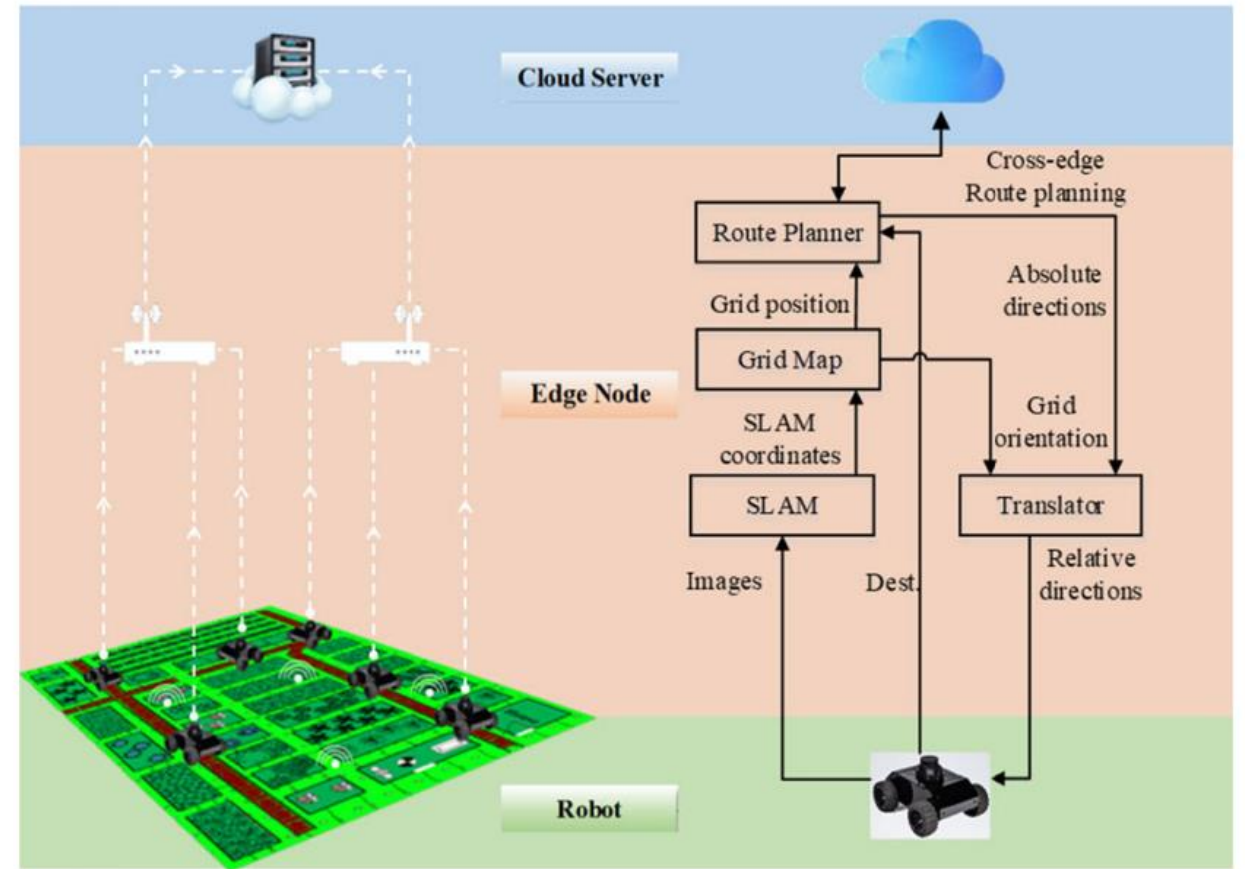


FIGURE 1. The IoT architecture of Cloud-Edge-Robot.

Summary

**Motivation/purpose/
aims/hypothesis**

Contribution

Methodology

Conclusion

Summary

**Motivation/purpose/
aims/hypothesis**

Contribution

Methodology

Conclusion

Summary

Motivation/purpose/
aims/hypothesis

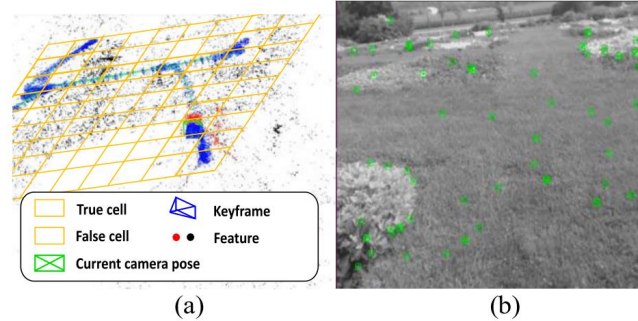


FIGURE 3. a) A demonstration of mn-scaled meshing with SLAM map at b) real-time farm view.

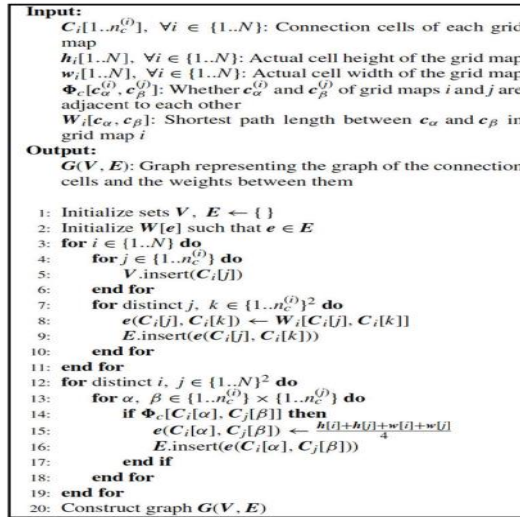


FIGURE 5. Route planning algorithm based on the Mesh-map.

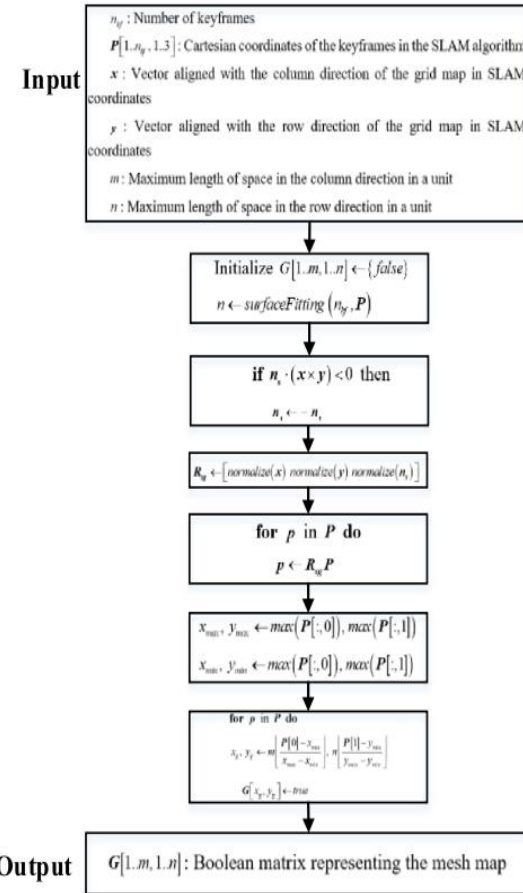


FIGURE 4. mn-Scaled Meshing algorithm.

Methodology

Conclusion

Summary

Contribution

- IoT-based mapping (Fig. 3)
- Computer vision and edge computing (Fig. 4)
- Advancing precision agriculture (Fig. 5)

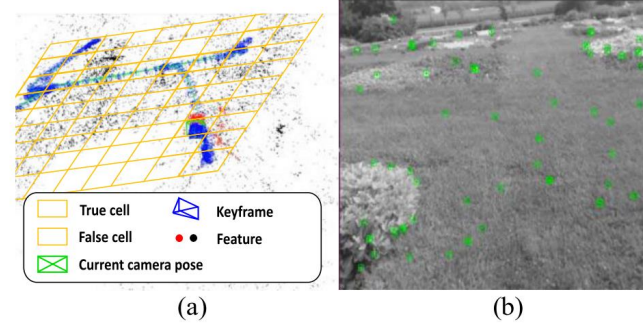


FIGURE 3. a) A demonstration of mn-scaled meshing with SLAM map and b) real-time farm view.

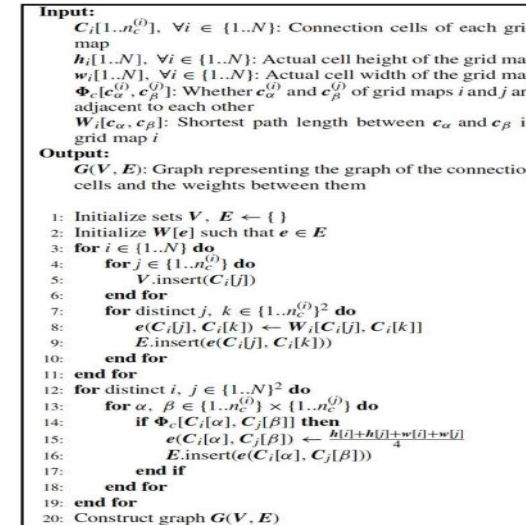


FIGURE 5. Route planning algorithm based on the Mesh-map.

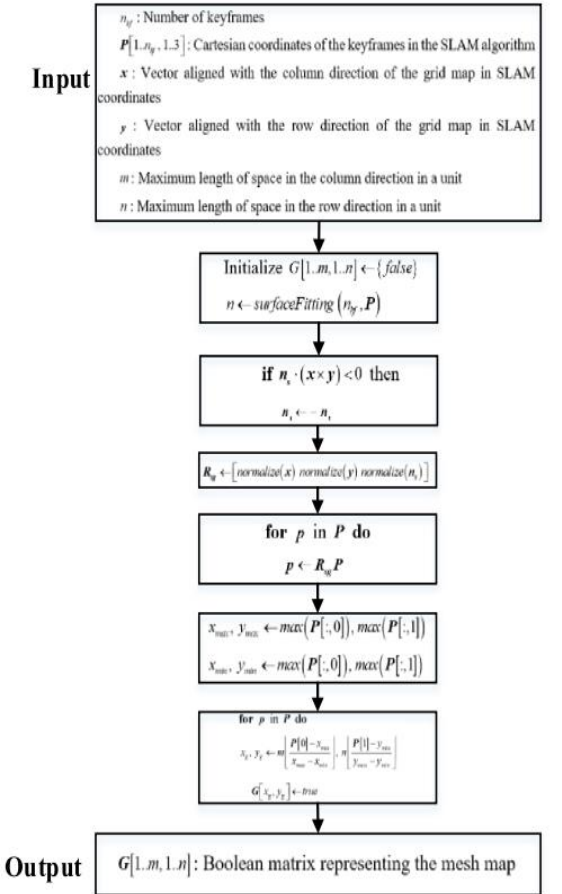


FIGURE 4. mn-Scaled Meshing algorithm.

Summary

Contribution

- IoT-based mapping (Fig. 3)
- Computer vision and edge computing (Fig. 4)
- Advancing precision agriculture (Fig. 5)

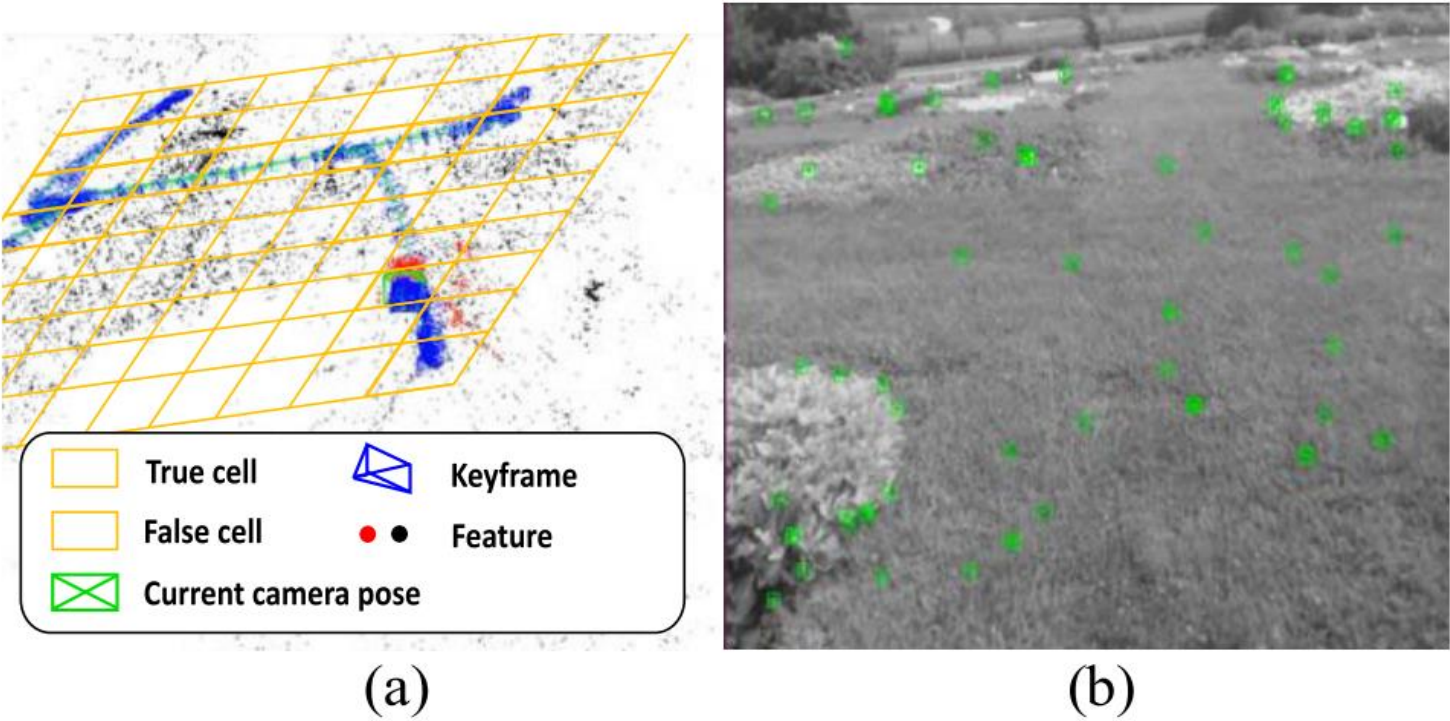


FIGURE 3. a) A demonstration of mn-scaled meshing with SLAM map and b) real-time farm view.

Input:
 $C_i \in \{L, H\}^N$, $W_i \in \{1, N\}$: Connection cells of each grid map
 $h_i \in \{1, N\}$, $v_i \in \{1, N\}$: Actual cell height of the grid map
 $w_i \in \{1, N\}$, $v_i \in \{1, N\}$: Actual cell width of the grid map
 $\Phi_i(C_i^{(h)}, C_j^{(h)})$: Whether $C_i^{(h)}$ and $C_j^{(h)}$ of grid maps i and j are adjacent to each other
 $W_{ij}(C_i, C_j)$: Shortest path length between C_i and C_j in grid map i
Output:
 GV, E : Graph representing the graph of the connection cells and the weights between them

```
1: Initialize sets  $V, E \leftarrow \{\}$ 
2: Initialize  $W_{ij}$  such that  $e \in E$ 
3: for  $i \in \{1, N\}$  do
4:   for  $j \in \{1, N\}$  do
5:      $V \leftarrow V \cup \{C_i, C_j\}$ 
6:   end for
7:   for distinct  $j, k \in \{1, N\}^2$  do
8:      $e(C_i, C_j, C_k) \leftarrow W_{ij}(C_i, C_j) + W_{jk}(C_j, C_k)$ 
9:      $E \leftarrow E \cup \{e(C_i, C_j, C_k)\}$ 
10:  end for
11: end for
12: for distinct  $i, j \in \{1, N\}^2$  do
13:   for  $u, v \in \{1, N\} \times \{1, N\}$  do
14:     if  $\Phi_i(C_i^{(u)}, C_j^{(v)})$  then
15:        $e(C_i^{(u)}, C_j^{(v)}) \leftarrow \frac{W_{ij}(C_i^{(u)}, C_j^{(v)}) + W_{jk}(C_j^{(u)}, C_k^{(v)})}{4}$ 
16:        $E \leftarrow E \cup \{e(C_i^{(u)}, C_j^{(v)})\}$ 
17:     end if
18:   end for
19: end for
20: Construct graph  $GV, E$ 
```

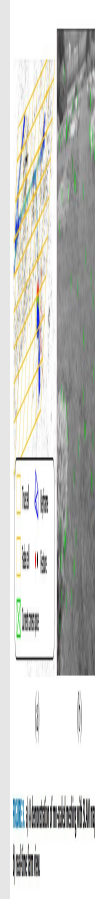
Out

FIGURE 5. Route planning algorithm based on the Mesh-map.

Summary

Contribution

- IoT-based mapping (Fig. 3)
- Computer vision and edge computing (Fig. 4)
- Advancing precision agriculture (Fig. 5)



Input

n_k : Number of keyframes
 $P[1..n_k, 1..3]$: Cartesian coordinates of the keyframes in the SLAM algorithm
 x : Vector aligned with the column direction of the grid map in SLAM coordinates
 y : Vector aligned with the row direction of the grid map in SLAM coordinates
 m : Maximum length of space in the column direction in a unit
 n : Maximum length of space in the row direction in a unit

Output

$G[1..m, 1..n]$: Boolean matrix representing the mesh map

FIGURE 4. mn-Scaled Meshing algorithm.

Input:

$C_i[1..n_c^{(i)}]$, $\forall i \in \{1..N\}$: Connection cells of each grid map
 $h_i[1..N]$, $\forall i \in \{1..N\}$: Actual cell height of the grid map
 $w_i[1..N]$, $\forall i \in \{1..N\}$: Actual cell width of the grid map
 $\Phi_c[c_\alpha^{(i)}, c_\beta^{(j)}]$: Whether $c_\alpha^{(i)}$ and $c_\beta^{(j)}$ of grid maps i and j are adjacent to each other
 $W_i[c_\alpha, c_\beta]$: Shortest path length between c_α and c_β in grid map i

Output:

$G(V, E)$: Graph representing the graph of the connection cells and the weights between them

```

1: Initialize sets  $V, E \leftarrow \{\}$ 
2: Initialize  $W[e]$  such that  $e \in E$ 
3: for  $i \in \{1..N\}$  do
4:   for  $j \in \{1..n_c^{(i)}\}$  do
5:      $V.insert(C_i[j])$ 
6:   end for
7:   for distinct  $j, k \in \{1..n_c^{(i)}\}^2$  do
8:      $e(C_i[j], C_i[k]) \leftarrow W_i[C_i[j], C_i[k]]$ 
9:      $E.insert(e(C_i[j], C_i[k]))$ 
10:  end for
11: end for
12: for distinct  $i, j \in \{1..N\}^2$  do
13:   for  $\alpha, \beta \in \{1..n_c^{(i)}\} \times \{1..n_c^{(j)}\}$  do
14:     if  $\Phi_c[C_i[\alpha], C_j[\beta]]$  then
15:        $e(C_i[\alpha], C_j[\beta]) \leftarrow \frac{h[i]+h[j]+w[i]+w[j]}{4}$ 
16:        $E.insert(e(C_i[\alpha], C_j[\beta]))$ 
17:     end if
18:   end for
19: end for
20: Construct graph  $G(V, E)$ 

```

FIGURE 5. Route planning algorithm based on the Mesh-map.

Summary

**Motivation/purpose/
aims/hypothesis**

Contribution

Methodology

Conclusion

Summary

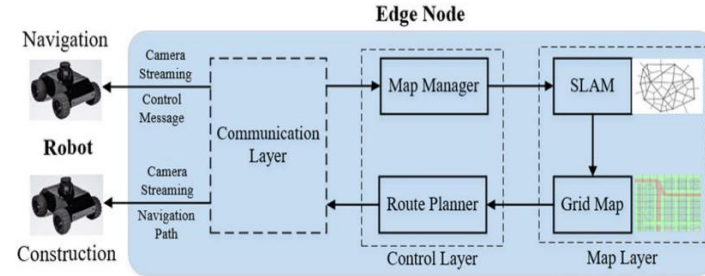
**Motivation/purpose/
aims/hypothesis**

Contribution

Methodology

Conclusion

Summary



Input:
 $C_i[1..n_c^{(i)}], \forall i \in \{1..N\}$: Connection cells of each grid map
 $h_i[1..N], \forall i \in \{1..N\}$: Actual cell height of the grid map
 $w_i[1..N], \forall i \in \{1..N\}$: Actual cell width of the grid map
 $\Phi_c[c_\alpha^{(i)}, c_\beta^{(j)}]$: Whether $c_\alpha^{(i)}$ and $c_\beta^{(j)}$ of grid maps i and j are adjacent to each other
 $W_i[c_\alpha, c_\beta]$: Shortest path length between c_α and c_β in grid map i

Output:
 $G(V, E)$: Graph representing the graph of the connection cells and the weights between them

```

1: Initialize sets  $V, E \leftarrow \{\}$ 
2: Initialize  $W[e]$  such that  $e \in E$ 
3: for  $i \in \{1..N\}$  do
4:   for  $j \in \{1..n_c^{(i)}\}$  do
5:      $V.insert(C_i[j])$ 
6:   end for
7:   for distinct  $j, k \in \{1..n_c^{(i)}\}^2$  do
8:      $e(C_i[j], C_i[k]) \leftarrow W_i[C_i[j], C_i[k]]$ 
9:      $E.insert(e(C_i[j], C_i[k]))$ 
10:  end for
11: end for
12: for distinct  $i, j \in \{1..N\}^2$  do
13:   for  $\alpha, \beta \in \{1..n_c^{(i)}\} \times \{1..n_c^{(j)}\}$  do
14:     if  $\Phi_c[C_i[\alpha], C_j[\beta]]$  then
15:        $e(C_i[\alpha], C_j[\beta]) \leftarrow \frac{h[i] + h[j] + w[i] + w[j]}{4}$ 
16:        $E.insert(e(C_i[\alpha], C_j[\beta]))$ 
17:     end if
18:   end for
19: end for
20: Construct graph  $G(V, E)$ 
    
```

FIGURE 5. Route planning algorithm based on the Mesh-map.

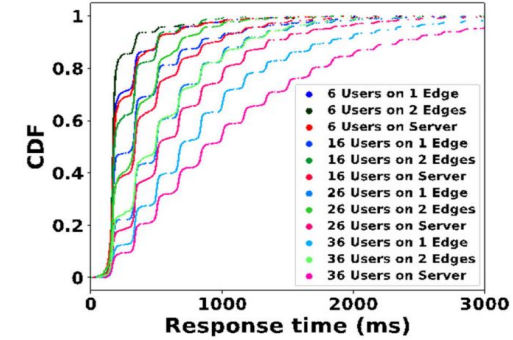


FIGURE 12. The CDF of the time intervals between responses. Note: User means a working robot.

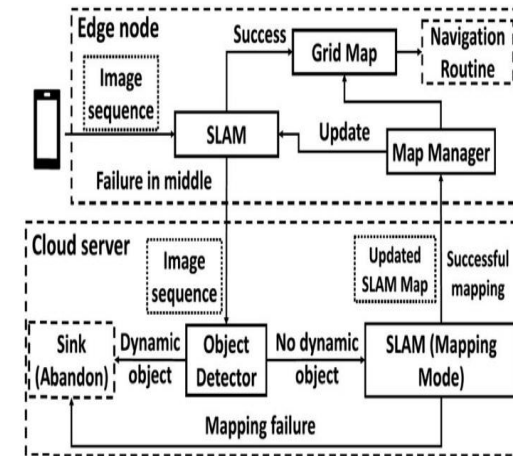


FIGURE 9. The flowchart of map maintenance.

Motivation/purpose/
aims/hypothesis

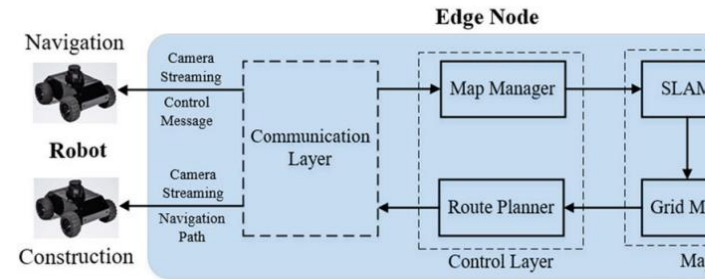
Contribution

Conclusion

Summary

Methodology

- Monocular cameras, SLAM, mesh maps (Figs. 2, 5)
- Accuracy, CPU usage, localization
 - experiments (Figs. 9, 12)
- Real-time mapping for precision agriculture (Fig. 8)



Input:

- $C_i[1..n_c^{(i)}], \forall i \in \{1..N\}$: Connection cells of each grid map
- $h_i[1..N], \forall i \in \{1..N\}$: Actual cell height of the grid map
- $w_i[1..N], \forall i \in \{1..N\}$: Actual cell width of the grid map
- $\Phi_c[c_\alpha^{(i)}, c_\beta^{(j)}]$: Whether $c_\alpha^{(i)}$ and $c_\beta^{(j)}$ of grid maps i and j are adjacent to each other
- $W_i[c_\alpha, c_\beta]$: Shortest path length between c_α and c_β in grid map i

Output:

$G(V, E)$: Graph representing the graph of the connection cells and the weights between them

- 1: Initialize sets $V, E \leftarrow \{\}$
- 2: Initialize $W[e]$ such that $e \in E$
- 3: **for** $i \in \{1..N\}$ **do**
- 4: **for** $j \in \{1..n_c^{(i)}\}$ **do**
- 5: $V.insert(C_i[j])$
- 6: **end for**
- 7: **for** distinct $j, k \in \{1..n_c^{(i)}\}^2$ **do**
- 8: $e(C_i[j], C_i[k]) \leftarrow W_i[C_i[j], C_i[k]]$
- 9: $E.insert(e(C_i[j], C_i[k]))$
- 10: **end for**
- 11: **end for**
- 12: **for** distinct $i, j \in \{1..N\}^2$ **do**
- 13: **for** $\alpha, \beta \in \{1..n_c^{(i)}\} \times \{1..n_c^{(j)}\}$ **do**
- 14: **if** $\Phi_c[C_i[\alpha], C_j[\beta]]$ **then**
- 15: $e(C_i[\alpha], C_j[\beta]) \leftarrow \frac{h[i]+h[j]+w[i]+w[j]}{4}$
- 16: $E.insert(e(C_i[\alpha], C_j[\beta]))$
- 17: **end if**
- 18: **end for**
- 19: **end for**
- 20: Construct graph $G(V, E)$

FIGURE 5. Route planning algorithm based on the Mesh-map.

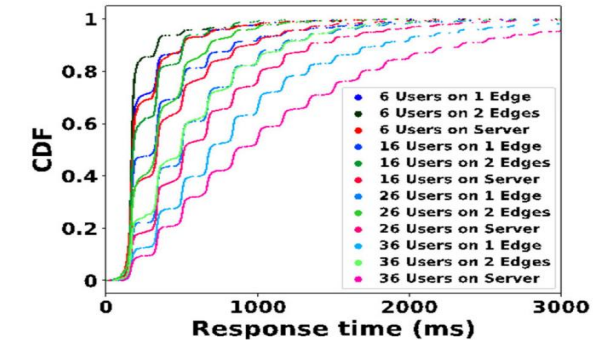


FIGURE 12. The CDF of the time intervals between responses. Note: User means a working robot.

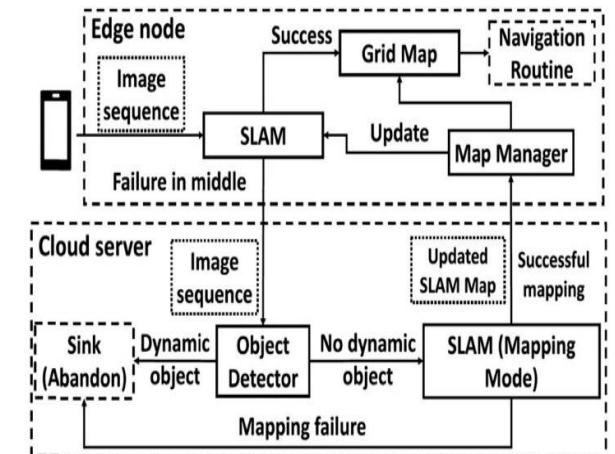


FIGURE 9. The flowchart of map maintenance.

Summary

Methodology

- Monocular cameras, SLAM, mesh maps (Figs. 2, 5)
- Accuracy, CPU usage, localization
- experiments (Figs. 9, 12)
- Real-time mapping for precision agriculture (Fig. 8)

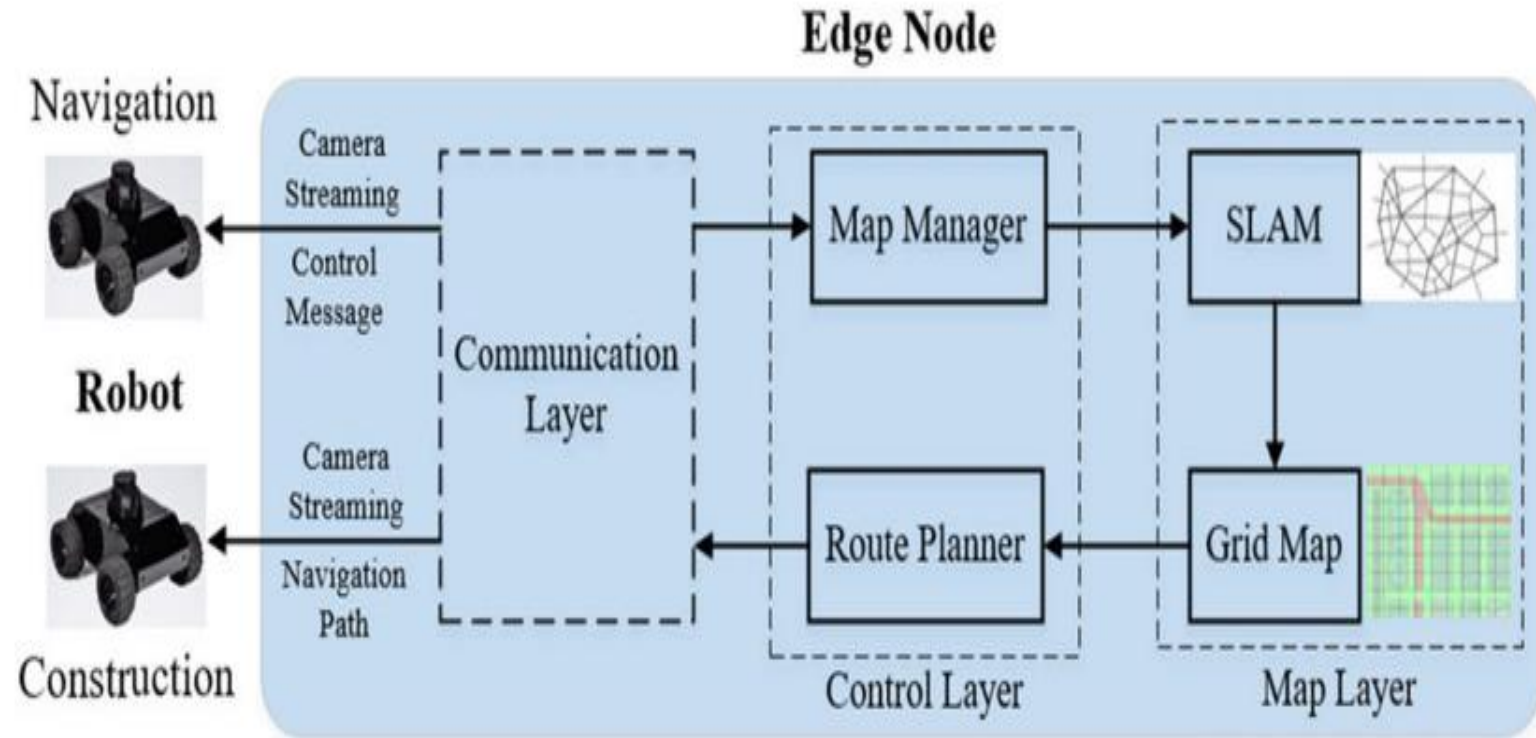


FIGURE 2. The Structure of the Edge Node Layer.

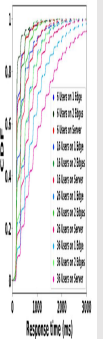


FIGURE 10. The US of the line intervals between responses. Note: User was a working child.



USE 1 The forecast of my maintenance.

Appendix 1

Fig. 1 $W_{ij}(x, y)$ and $W_{ij}(y, x)$ Gametic values of each p_{ij} gene

$A_1A_2, B_1B_2, C_1C_2, D_1D_2$: Actual alleles of the polymorphic loci

$a_1a_2, b_1b_2, c_1c_2, d_1d_2$: Actual alleles of the polymorphic loci

$W_{ij}(x, y)$: Gametic value of p_{ij} gene when x and y are adjacent in a molecule

$W_{ij}(y, x)$: Gametic value of p_{ij} gene when y and x are adjacent in a molecule

Fig. 2 Graph representing the graph of the association values and allele weights between loci

1. $W_{ij}(x, y) = 1, x = y$

2. $W_{ij}(x, y) = 0, x \neq y$

3. $W_{ij}(x, y) = 1, x = y$

4. $W_{ij}(x, y) = 0, x \neq y$

5. $W_{ij}(x, y) = 1, x = y$

6. $W_{ij}(x, y) = 0, x \neq y$

7. $W_{ij}(x, y) = 1, x = y$

8. $W_{ij}(x, y) = 0, x \neq y$

9. $W_{ij}(x, y) = 1, x = y$

10. $W_{ij}(x, y) = 0, x \neq y$

11. $W_{ij}(x, y) = 1, x = y$

12. $W_{ij}(x, y) = 0, x \neq y$

13. $W_{ij}(x, y) = 1, x = y$

14. $W_{ij}(x, y) = 0, x \neq y$

15. $W_{ij}(x, y) = 1, x = y$

16. $W_{ij}(x, y) = 0, x \neq y$

17. $W_{ij}(x, y) = 1, x = y$

18. $W_{ij}(x, y) = 0, x \neq y$

19. $W_{ij}(x, y) = 1, x = y$

20. $W_{ij}(x, y) = 0, x \neq y$

21. $W_{ij}(x, y) = 1, x = y$

22. $W_{ij}(x, y) = 0, x \neq y$

23. $W_{ij}(x, y) = 1, x = y$

24. $W_{ij}(x, y) = 0, x \neq y$

25. $W_{ij}(x, y) = 1, x = y$

26. $W_{ij}(x, y) = 0, x \neq y$

27. $W_{ij}(x, y) = 1, x = y$

28. $W_{ij}(x, y) = 0, x \neq y$

29. $W_{ij}(x, y) = 1, x = y$

30. $W_{ij}(x, y) = 0, x \neq y$

31. $W_{ij}(x, y) = 1, x = y$

32. $W_{ij}(x, y) = 0, x \neq y$

33. $W_{ij}(x, y) = 1, x = y$

34. $W_{ij}(x, y) = 0, x \neq y$

35. $W_{ij}(x, y) = 1, x = y$

36. $W_{ij}(x, y) = 0, x \neq y$

37. $W_{ij}(x, y) = 1, x = y$

38. $W_{ij}(x, y) = 0, x \neq y$

39. $W_{ij}(x, y) = 1, x = y$

40. $W_{ij}(x, y) = 0, x \neq y$

41. $W_{ij}(x, y) = 1, x = y$

42. $W_{ij}(x, y) = 0, x \neq y$

43. $W_{ij}(x, y) = 1, x = y$

44. $W_{ij}(x, y) = 0, x \neq y$

45. $W_{ij}(x, y) = 1, x = y$

46. $W_{ij}(x, y) = 0, x \neq y$

47. $W_{ij}(x, y) = 1, x = y$

48. $W_{ij}(x, y) = 0, x \neq y$

49. $W_{ij}(x, y) = 1, x = y$

50. $W_{ij}(x, y) = 0, x \neq y$

51. $W_{ij}(x, y) = 1, x = y$

52. $W_{ij}(x, y) = 0, x \neq y$

53. $W_{ij}(x, y) = 1, x = y$

54. $W_{ij}(x, y) = 0, x \neq y$

55. $W_{ij}(x, y) = 1, x = y$

56. $W_{ij}(x, y) = 0, x \neq y$

57. $W_{ij}(x, y) = 1, x = y$

58. $W_{ij}(x, y) = 0, x \neq y$

59. $W_{ij}(x, y) = 1, x = y$

60. $W_{ij}(x, y) = 0, x \neq y$

61. $W_{ij}(x, y) = 1, x = y$

62. $W_{ij}(x, y) = 0, x \neq y$

63. $W_{ij}(x, y) = 1, x = y$

64. $W_{ij}(x, y) = 0, x \neq y$

65. $W_{ij}(x, y) = 1, x = y$

66. $W_{ij}(x, y) = 0, x \neq y$

67. $W_{ij}(x, y) = 1, x = y$

68. $W_{ij}(x, y) = 0, x \neq y$

69. $W_{ij}(x, y) = 1, x = y$

70. $W_{ij}(x, y) = 0, x \neq y$

71. $W_{ij}(x, y) = 1, x = y$

72. $W_{ij}(x, y) = 0, x \neq y$

73. $W_{ij}(x, y) = 1, x = y$

74. $W_{ij}(x, y) = 0, x \neq y$

75. $W_{ij}(x, y) = 1, x = y$

76. $W_{ij}(x, y) = 0, x \neq y$

77. $W_{ij}(x, y) = 1, x = y$

78. $W_{ij}(x, y) = 0, x \neq y$

79. $W_{ij}(x, y) = 1, x = y$

80. $W_{ij}(x, y) = 0, x \neq y$

81. $W_{ij}(x, y) = 1, x = y$

82. $W_{ij}(x, y) = 0, x \neq y$

83. $W_{ij}(x, y) = 1, x = y$

84. $W_{ij}(x, y) = 0, x \neq y$

85. $W_{ij}(x, y) = 1, x = y$

86. $W_{ij}(x, y) = 0, x \neq y$

87. $W_{ij}(x, y) = 1, x = y$

88. $W_{ij}(x, y) = 0, x \neq y$

89. $W_{ij}(x, y) = 1, x = y$

90. $W_{ij}(x, y) = 0, x \neq y$

91. $W_{ij}(x, y) = 1, x = y$

92. $W_{ij}(x, y) = 0, x \neq y$

93. $W_{ij}(x, y) = 1, x = y$

94. $W_{ij}(x, y) = 0, x \neq y$

95. $W_{ij}(x, y) = 1, x = y$

96. $W_{ij}(x, y) = 0, x \neq y$

97. $W_{ij}(x, y) = 1, x = y$

98. $W_{ij}(x, y) = 0, x \neq y$

99. $W_{ij}(x, y) = 1, x = y$

100. $W_{ij}(x, y) = 0, x \neq y$

101. $W_{ij}(x, y) = 1, x = y$

102. $W_{ij}(x, y) = 0, x \neq y$

103. $W_{ij}(x, y) = 1, x = y$

104. $W_{ij}(x, y) = 0, x \neq y$

105. $W_{ij}(x, y) = 1, x = y$

106. $W_{ij}(x, y) = 0, x \neq y$

107. $W_{ij}(x, y) = 1, x = y$

108. $W_{ij}(x, y) = 0, x \neq y$

109. $W_{ij}(x, y) = 1, x = y$

110. $W_{ij}(x, y) = 0, x \neq y$

111. $W_{ij}(x, y) = 1, x = y$

112. $W_{ij}(x, y) = 0, x \neq y$

113. $W_{ij}(x, y) = 1, x = y$

114. $W_{ij}(x, y) = 0, x \neq y$

115. $W_{ij}(x, y) = 1, x = y$

116. $W_{ij}(x, y) = 0, x \neq y$

117. $W_{ij}(x, y) = 1, x = y$

118. $W_{ij}(x, y) = 0, x \neq y$

119. $W_{ij}(x, y) = 1, x = y$

120. $W_{ij}(x, y) = 0, x \neq y$

121. $W_{ij}(x, y) = 1, x = y$

122. $W_{ij}(x, y) = 0, x \neq y$

123. $W_{ij}(x, y) = 1, x = y$

124. $W_{ij}(x, y) = 0, x \neq y$

125. $W_{ij}(x, y) = 1, x = y$

126. $W_{ij}(x, y) = 0, x \neq y$

127. $W_{ij}(x, y) = 1, x = y$

128. $W_{ij}(x, y) = 0, x \neq y$

129. $W_{ij}(x, y) = 1, x = y$

130. $W_{ij}(x, y) = 0, x \neq y$

131. $W_{ij}(x, y) = 1, x$

Summary

Methodology

- Monocular cameras, SLAM, mesh maps (Figs. 2, 5)
- Accuracy, CPU usage, localization
- experiments (Figs. 9, 12)
- Real-time mapping for precision agriculture



Input:
 $C_i[1..n_c^{(i)}], \forall i \in \{1..N\}$: Connection cells of each grid map
 $h_i[1..N], \forall i \in \{1..N\}$: Actual cell height of the grid map
 $w_i[1..N], \forall i \in \{1..N\}$: Actual cell width of the grid map
 $\Phi_c[c_\alpha^{(i)}, c_\beta^{(j)}]$: Whether $c_\alpha^{(i)}$ and $c_\beta^{(j)}$ of grid maps i and j are adjacent to each other
 $W_i[c_\alpha, c_\beta]$: Shortest path length between c_α and c_β in grid map i

Output:
 $G(V, E)$: Graph representing the graph of the connection cells and the weights between them

```

1: Initialize sets  $V, E \leftarrow \{\}$ 
2: Initialize  $W[e]$  such that  $e \in E$ 
3: for  $i \in \{1..N\}$  do
4:   for  $j \in \{1..n_c^{(i)}\}$  do
5:      $V.insert(C_i[j])$ 
6:   end for
7:   for distinct  $j, k \in \{1..n_c^{(i)}\}^2$  do
8:      $e(C_i[j], C_i[k]) \leftarrow W_i[C_i[j], C_i[k]]$ 
9:      $E.insert(e(C_i[j], C_i[k]))$ 
10:  end for
11: end for
12: for distinct  $i, j \in \{1..N\}^2$  do
13:   for  $\alpha, \beta \in \{1..n_c^{(i)}\} \times \{1..n_c^{(j)}\}$  do
14:    if  $\Phi_c[C_i[\alpha], C_j[\beta]]$  then
15:       $e(C_i[\alpha], C_j[\beta]) \leftarrow \frac{h[i]+h[j]+w[i]+w[j]}{4}$ 
16:       $E.insert(e(C_i[\alpha], C_j[\beta]))$ 
17:    end if
18:  end for
19: end for
20: Construct graph  $G(V, E)$ 
  
```

FIGURE 5. Route planning algorithm based on the Mesh-map.

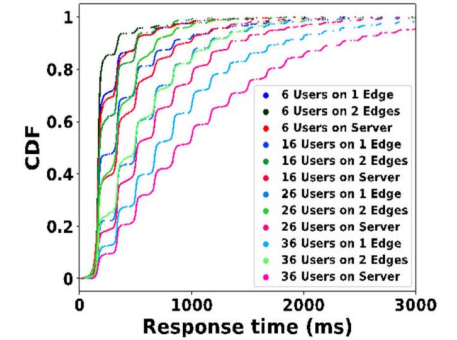


FIGURE 12. The CDF of the time intervals between responses. Note: User means a working robot.

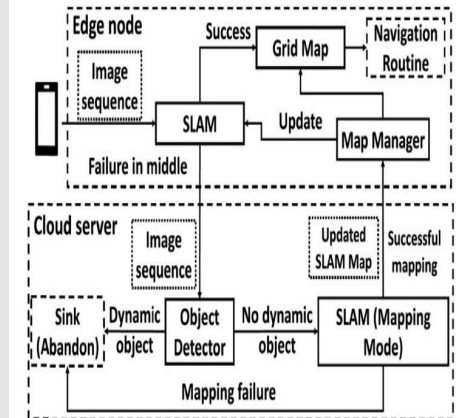


FIGURE 9. The flowchart of map maintenance.

Summary

Methodology

- Monocular cameras, SLAM, mesh maps (Figs. 2, 5)
- Accuracy, CPU usage, localization
- experiments (Figs. 9, 12)
- Real-time mapping for precision agriculture

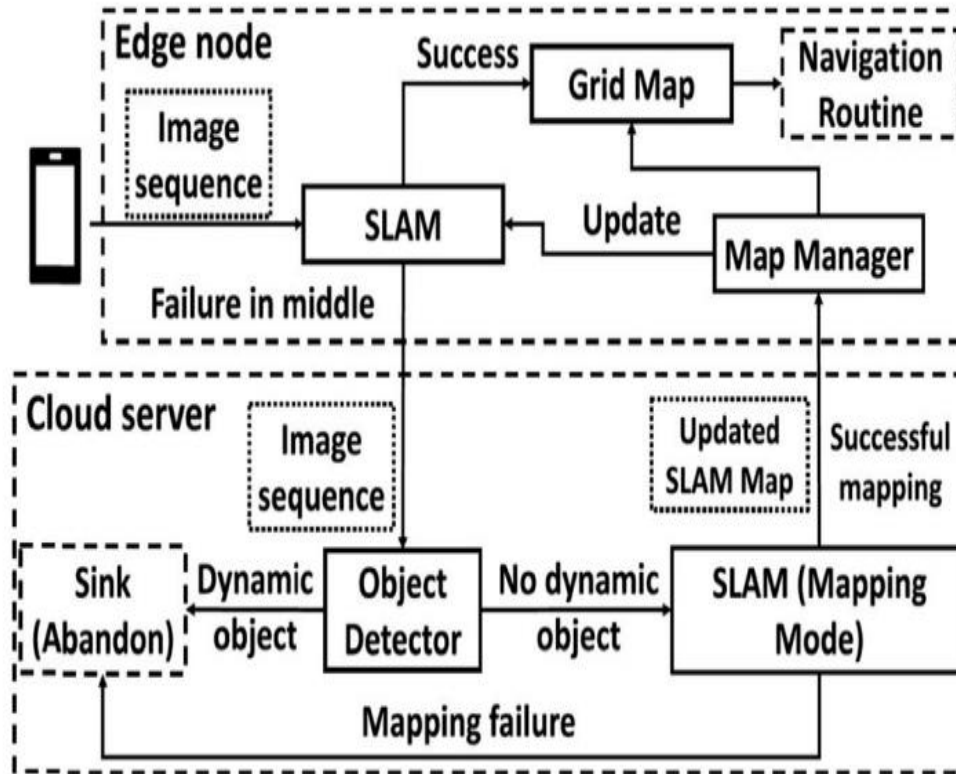


FIGURE 9. The flowchart of map maintenance.

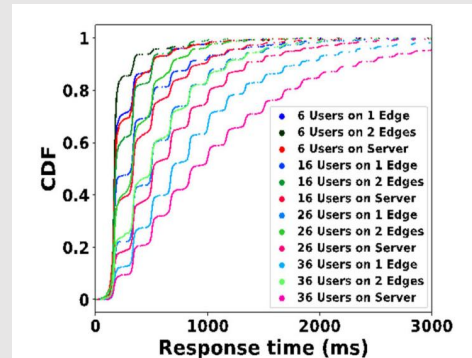


FIGURE 12. The CDF of the time intervals between responses. Note: User means a working robot.

Summary

Methodology

- Monocular cameras, SLAM, mesh maps (Figs. 2, 5)
- Accuracy, CPU usage, localization
- experiments (Figs. 9, 12)
- Real-time mapping for precision agriculture

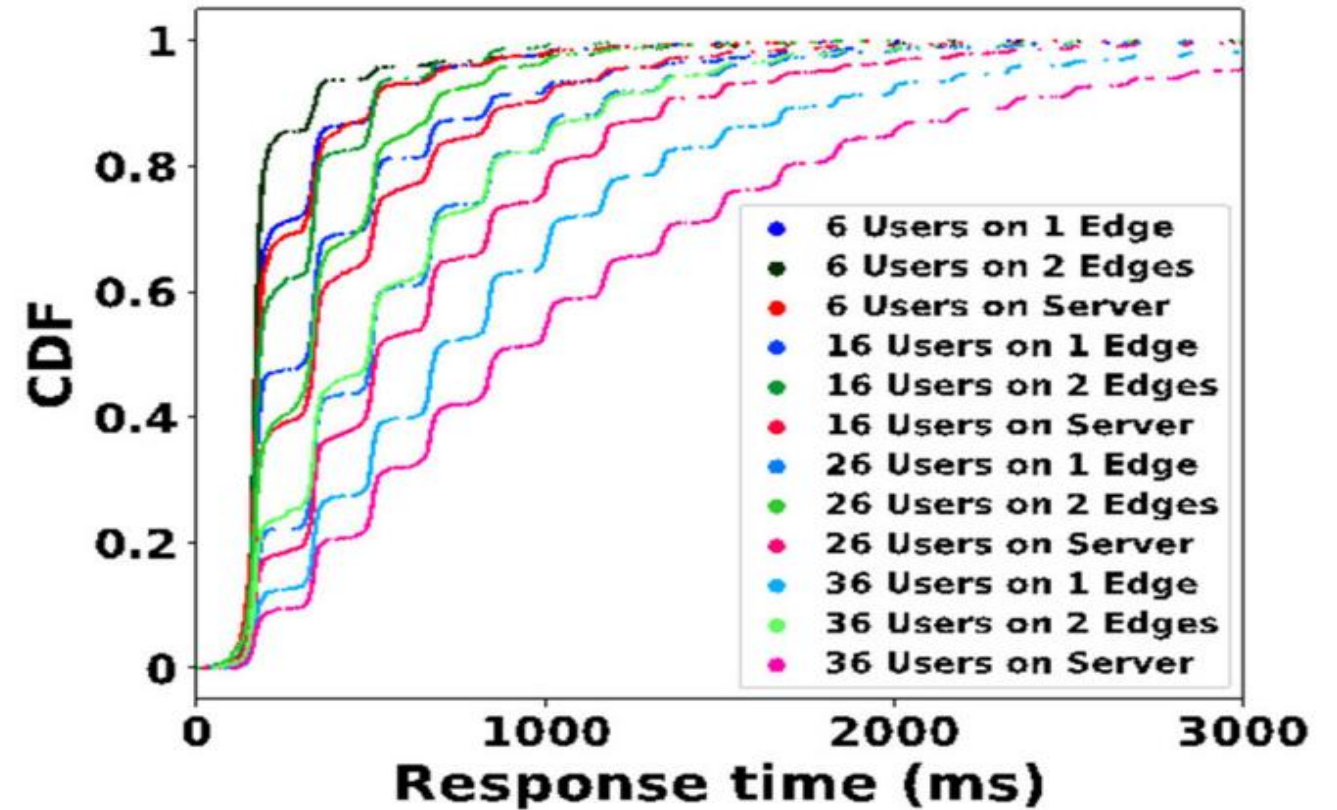
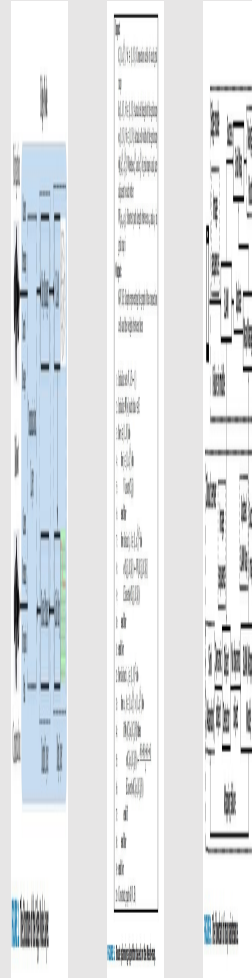


FIGURE 12. The CDF of the time intervals between responses. Note: User means a working robot.

Summary

**Motivation/purpose/
aims/hypothesis**

Contribution

Methodology

Conclusion

Summary

**Motivation/purpose/
aims/hypothesis**

Contribution

Methodology

Conclusion

Summary

Motivation/purpose/
aims/hypothesis

Contribution

Methodology

TABLE 4. Results from the accuracy experiment.

Cell length (approximate) (cm)	30	60
Localization success frequency (%)	84.7	89.3
RMSE (cm)	19.5	0
Maximal error (cm)	36.9	0
Orientation accuracy (%)	100	100

Note: Orientation includes 8 directions separated by 45 degrees.

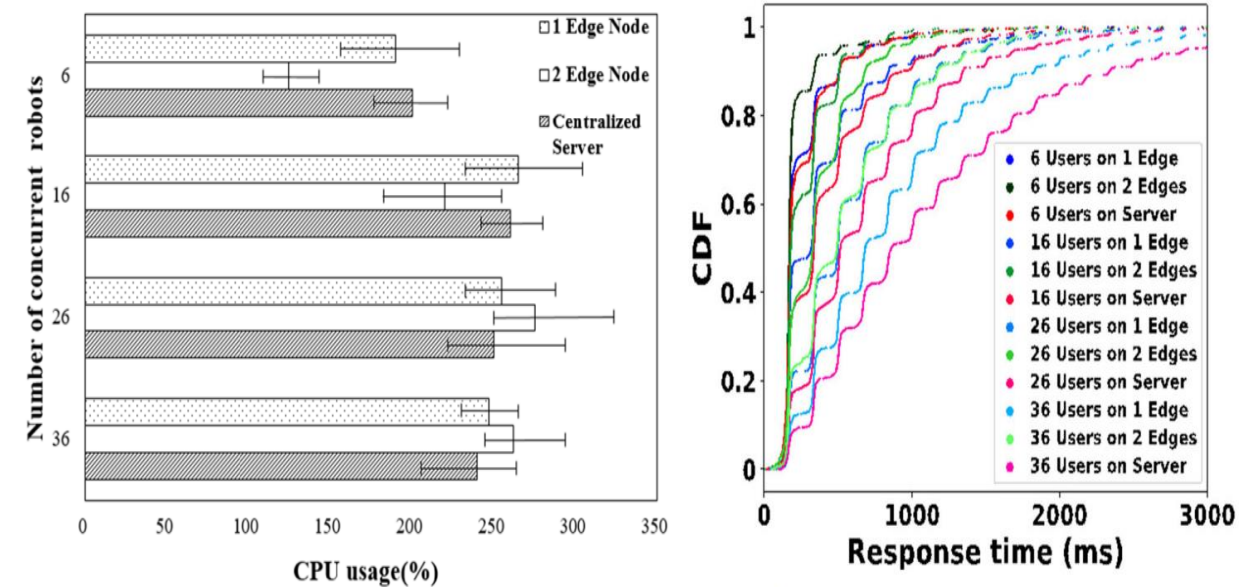


FIGURE 13. CPU usages in each experiment configuration.

FIGURE 12. The CDF of the time intervals between responses. Note: User means a working robot.

Summary

TABLE 4. Results from the accuracy experiment.

Cell length (approximate) (cm)	30	60
Localization success frequency (%)	84.7	89.3
RMSE (cm)	19.5	0
Maximal error (cm)	36.9	0
Orientation accuracy (%)	100	100

Note: Orientation includes 8 directions separated by 45 degrees.

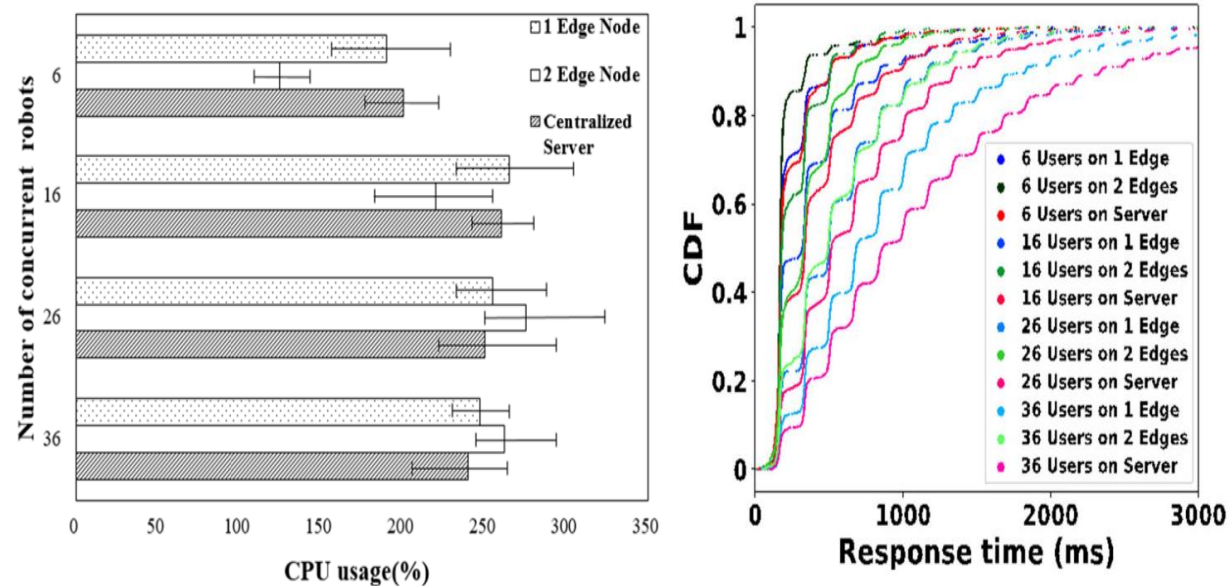


FIGURE 13. CPU usages in each experiment configuration.

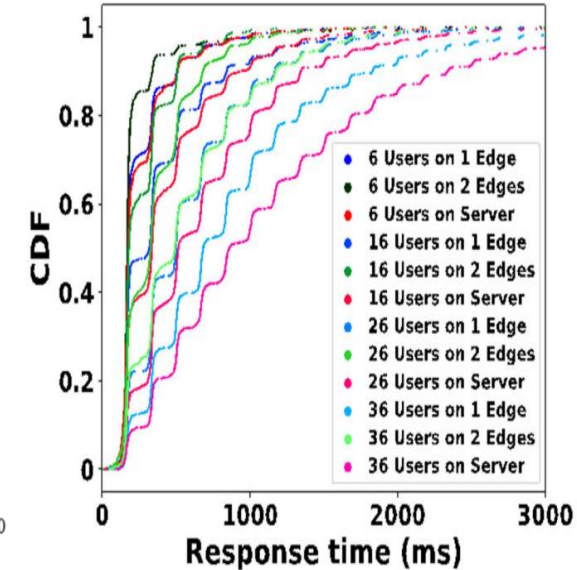


FIGURE 12. The CDF of the time intervals between responses. Note: User means a working robot.

Conclusion

- High accuracy (Table 4).
- Efficiency of edge computing (Fig. 13).
- Real-time data and map updates (Fig. 12).

Summary

TABLE 4. Results from the accuracy experiment.

Cell length (approximate) (cm)	30	60
Localization success frequency (%)	84.7	89.3
RMSE (cm)	19.5	0
Maximal error (cm)	36.9	0
Orientation accuracy (%)	100	100

Note: Orientation includes 8 directions separated by 45 degrees.

Conclusion

- High accuracy (Table 4).
- Efficiency of edge computing (Fig. 13).
- Real-time data and map updates (Fig. 12).

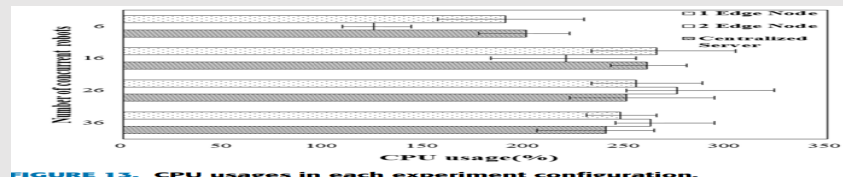


FIGURE 13. CPU usages in each experiment configuration.

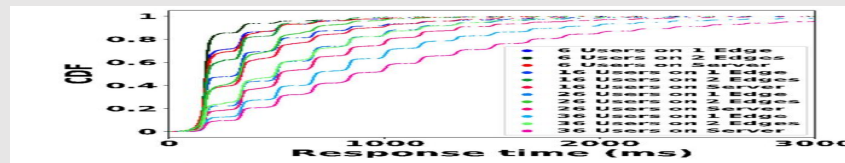


FIGURE 12. The CDF of the time intervals between responses. Note: User means a working robot.

Summary

TABLE 4. Results from the accuracy experiment.

Cell length (approximate) (cm)	30	60
Localization success frequency (%)	84.7	89.3
RMSE (cm)	12.0	0
Maximal error (cm)	10.0	0
Orientation accuracy (%)	100	100

Note: Orientation includes 8 directions separated by 45 degrees.

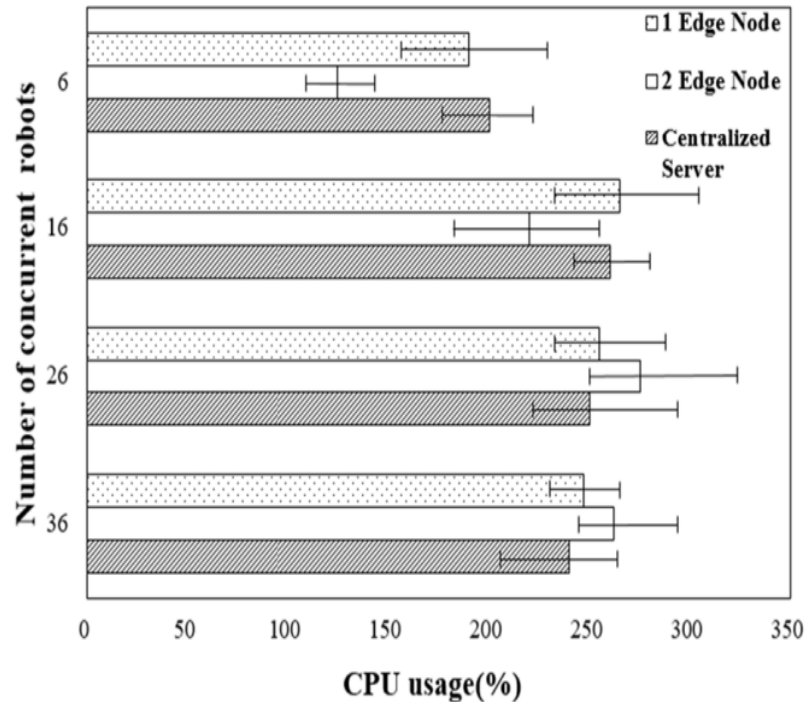


FIGURE 13. CPU usages in each experiment configuration.

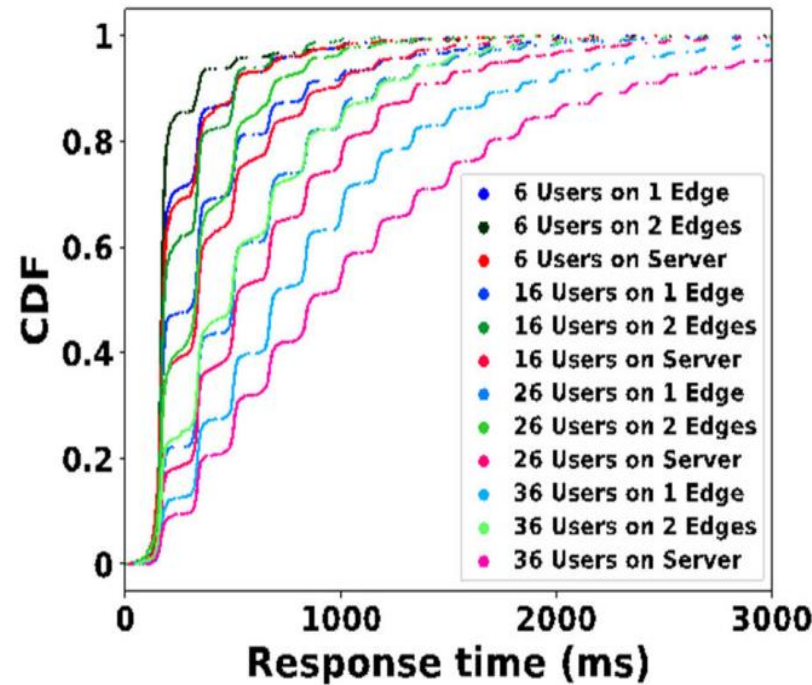


FIGURE 12. The CDF of the time intervals between responses. Note: User means a working robot.

Conclusion

- High accuracy (Table 4).
- Efficiency of edge computing (Fig. 13).
- Real-time data and map updates (Fig. 12).

Summary

**Motivation/purpose/
aims/hypothesis**

Contribution

Methodology

Conclusion

Limitations

Limitation 1

Limitation 2

Limitations

Limitation 1

Planar terrain
assumption

Limitation 2

SLAM technology
limitations

Limitations

Limitation 1

Planar terrain assumption

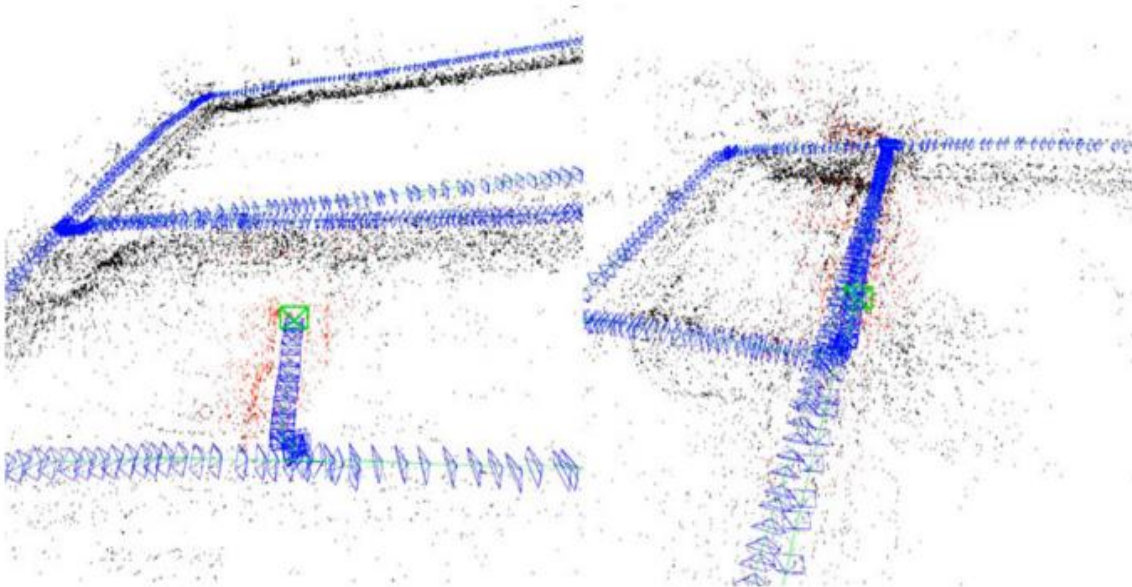


FIGURE 11. Building the SLAM map including (left) the map before path merging and (right) after path merging.

Limitation 2

SLAM technology limitations

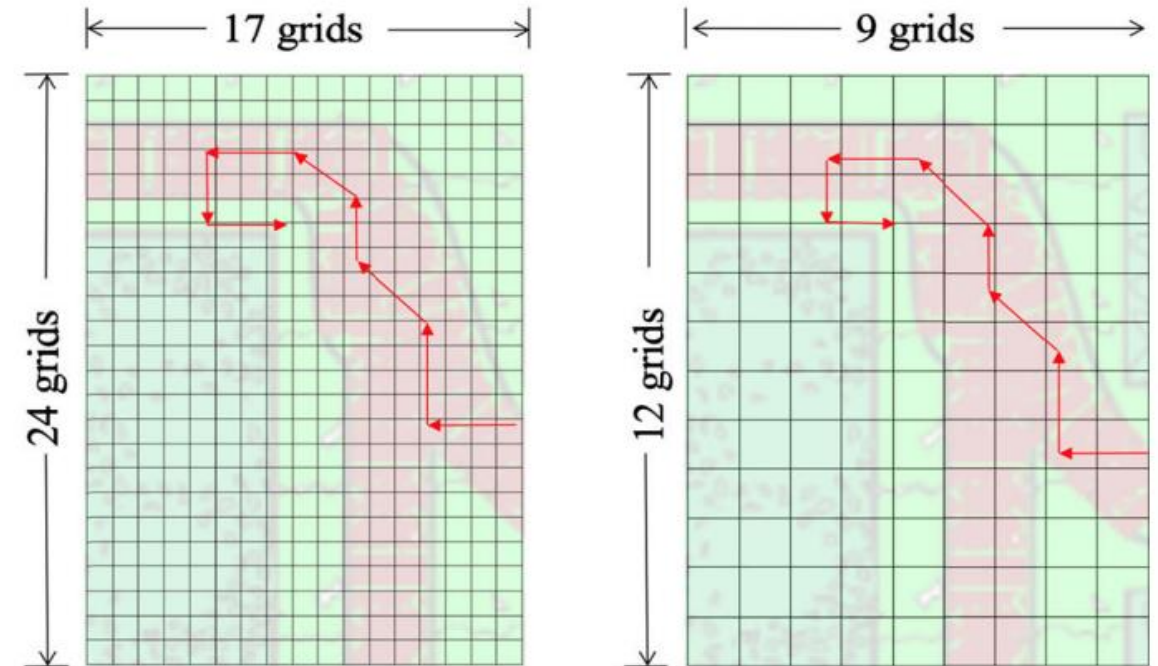


FIGURE 10. The configuration of the accuracy experiment with a unit of 30cm-cells (left) and 60cm-cells (right).

Synthesis

**Future
Applications**

**Environmental
Monitoring**

**Smart
Cities**

**Disaster
Response**

**Exploration &
Mapping**

**Versatile
Industries**

**Overcoming
Limitations**

**Enhancing
Adaptability**

Synthesis

**Future
Applications**

**Environmental
Monitoring**

**Smart
Cities**

**Disaster
Response**

**Exploration &
Mapping**

**Versatile
Industries**

**Overcoming
Limitations**

**Enhancing
Adaptability**

Synthesis

**Future
Applications**

**Environmental
Monitoring**

**Smart
Cities**

**Disaster
Response**

**Exploration &
Mapping**

**Versatile
Industries**

**Overcoming
Limitations**

**Enhancing
Adaptability**

Synthesis

**Future
Applications**

**Environmental
Monitoring**

**Smart
Cities**

**Disaster
Response**

**Exploration &
Mapping**

**Versatile
Industries**

**Overcoming
Limitations**

**Enhancing
Adaptability**

Synthesis

**Future
Applications**

**Environmental
Monitoring**

**Smart
Cities**

**Disaster
Response**

**Exploration &
Mapping**

**Versatile
Industries**

**Overcoming
Limitations**

**Enhancing
Adaptability**

Synthesis

**Future
Applications**

**Environmental
Monitoring**

**Smart
Cities**

**Disaster
Response**

**Exploration &
Mapping**

**Versatile
Industries**

**Overcoming
Limitations**

**Enhancing
Adaptability**

Synthesis

**Future
Applications**

**Environmental
Monitoring**

**Smart
Cities**

**Disaster
Response**

**Exploration &
Mapping**

**Versatile
Industries**

**Overcoming
Limitations**

**Enhancing
Adaptability**

Synthesis

**Future
Applications**

**Environmental
Monitoring**

**Smart
Cities**

**Disaster
Response**

**Exploration &
Mapping**

**Versatile
Industries**

**Overcoming
Limitations**

**Enhancing
Adaptability**

Synthesis

**Future
Applications**

**Environmental
Monitoring**

**Smart
Cities**

**Disaster
Response**

**Exploration &
Mapping**

**Versatile
Industries**

**Overcoming
Limitations**

**Enhancing
Adaptability**

Reference: Wei Zhao, Xuan Wang, Bozhao Qi, & Troy Runge. (2020). Ground-Level Mapping and Navigating for Agriculture Based on IoT and Computer Vision. *API* (Digital Object Identifier 10.1109/ACCESS.2020.3043662).