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

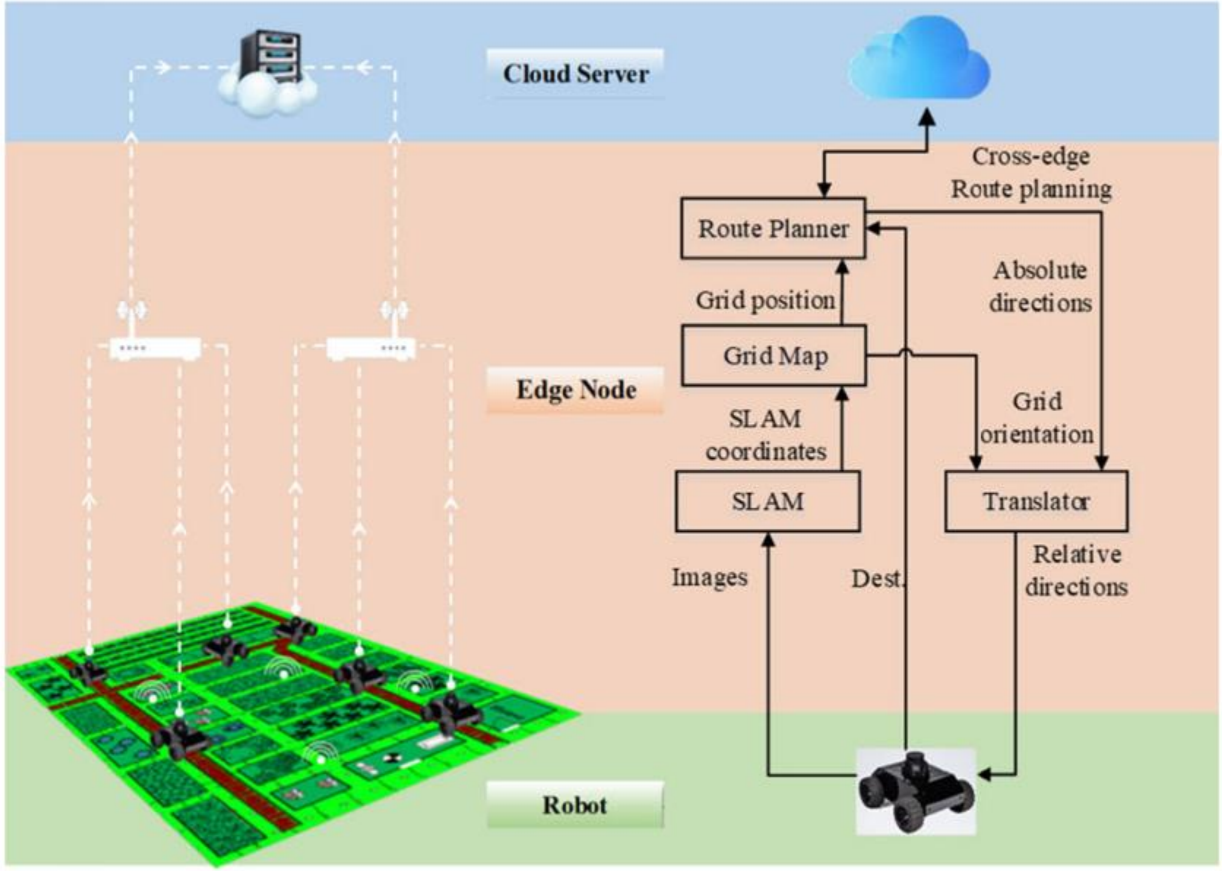


FIGURE 1. The IoT architecture of Cloud-Edge-Robot.

Contribution

Methodology

Conclusion

Summary

Motivation/purpose/ aims/hypothesis

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

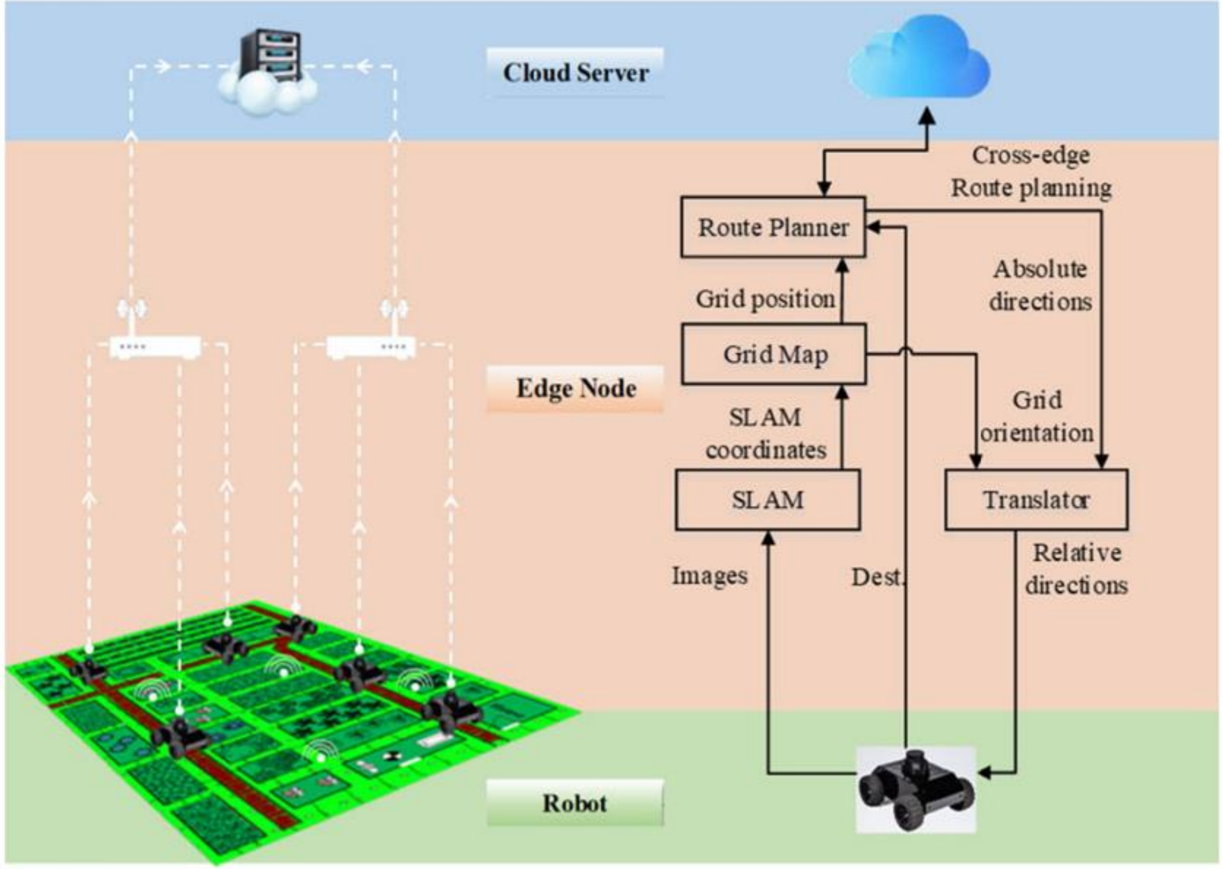


FIGURE 1. **The IoT architecture of Cloud-Edge-Robot.**

Summary

Motivation/purpose/aims/hypothesis

Contribution

Methodology

Conclusion

Summary

Motivation/purpose/
aims/hypothesis

Contribution

Methodology

Conclusion

Summary

Motivation/purpose/aims/hypothesis

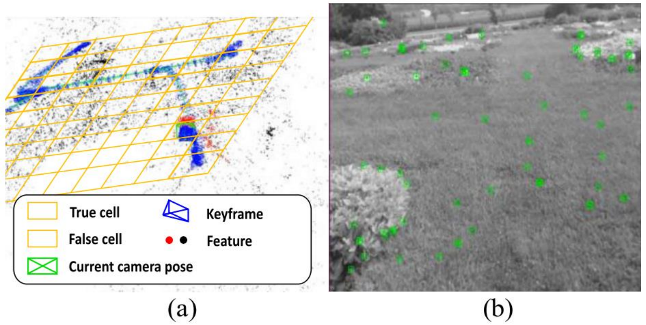
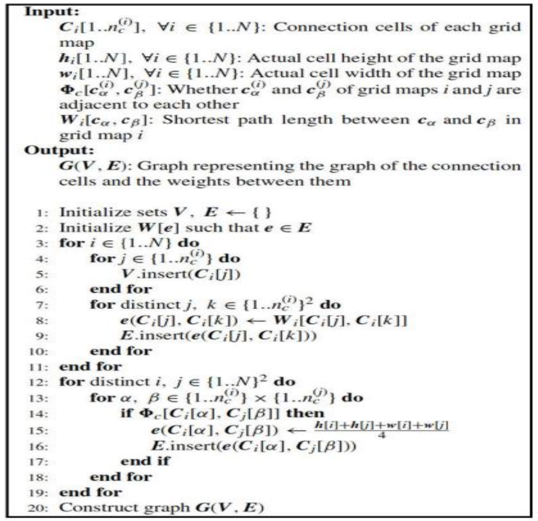
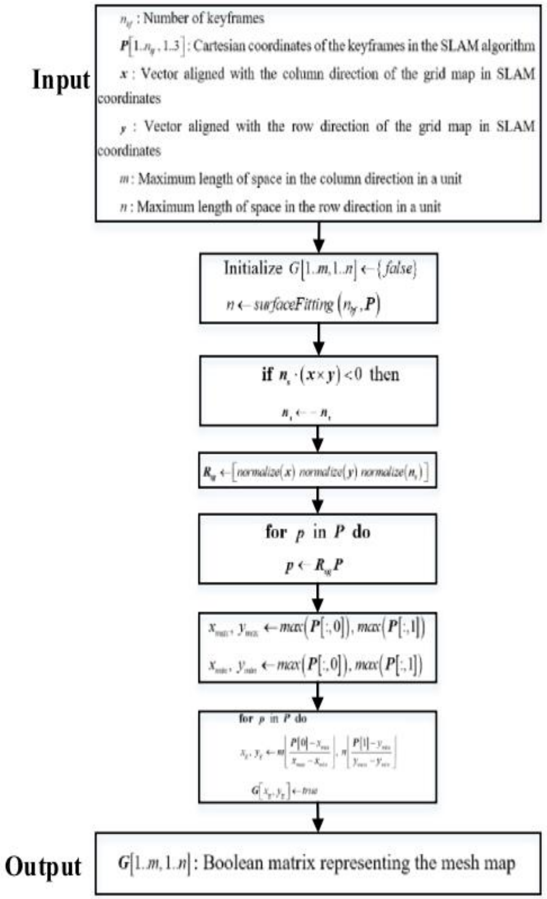


FIGURE 3. a) A demonstration of mn-scaled meshing with SLAM map a b) real-time farm view.



```
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 $c_\alpha$ and $c_\beta$ in grid map $i$
Output:
    $G(V, E)$: Graph representing the graph of the connection cells and the weights between them

Initialize sets $V,\ E \leftarrow \{\ \}$
Initialize $W[e]$ such that $e \in E$
for $i \in \{1..N\}$ do
    for $j \in \{1..n_c^{(i)}\}$ do
        $V$.insert($C_i[j]$)
    end for
    for distinct $j,\ k \in \{1..n_c^{(i)}\}^2$ do
        $e(C_i[j], C_i[k]) \leftarrow W_i[C_i[j], C_i[k]]$
        $E$.insert($e(C_i[j], C_i[k])$)
    end for
end for
for distinct $i,\ j \in \{1..N\}^2$ do
    for $\alpha,\ \beta \in \{1..n_c^{(i)}\} \times \{1..n_c^{(j)}\}$ do
        if $\Phi_c[C_i[\alpha], C_j[\beta]]$ then
            $e(C_i[\alpha], C_j[\beta]) \leftarrow \frac{h[i]+h[j]+w[i]+w[j]}{4}$
            $E$.insert($e(C_i[\alpha], C_j[\beta])$)
        end if
    end for
end for
Construct graph $G(V, E)$
```

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



Input

n_{kf} : Number of keyframes

$P[1..n_{kf}, 1..3]$: 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

Initialize $G[1..m, 1..n] \leftarrow \{false\}$

$n \leftarrow surfaceFitting(n_{kf}, P)$

if $n_s \cdot (x \times y) < 0$ then

$n_s \leftarrow -n_s$

$R_{sg} \leftarrow [normalize(x)\ normalize(y)\ normalize(n_s)]$

for p in P do

$p \leftarrow R_{sg}P$

$x_{max}, y_{max} \leftarrow max(P[:,0]), max(P[:,1])$

$x_{min}, y_{min} \leftarrow max(P[:,0]), max(P[:,1])$

for p in P do

$x_g, y_g \leftarrow m\left\lfloor \frac{P[0]-x_{min}}{x_{max}-x_{min}} \right\rfloor, n\left\lfloor \frac{P[1]-y_{min}}{y_{max}-y_{min}} \right\rfloor$

$G[x_g, y_g] \leftarrow true$

Output

$G[1..m, 1..n]$: Boolean matrix representing 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)

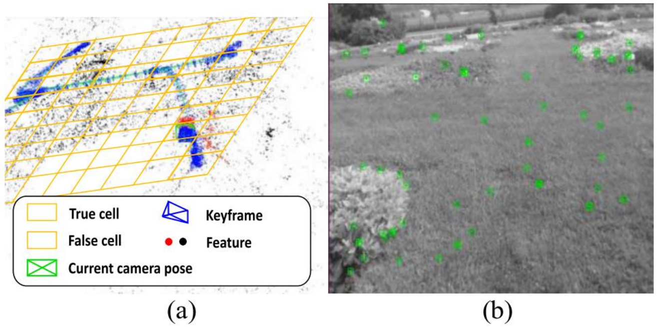
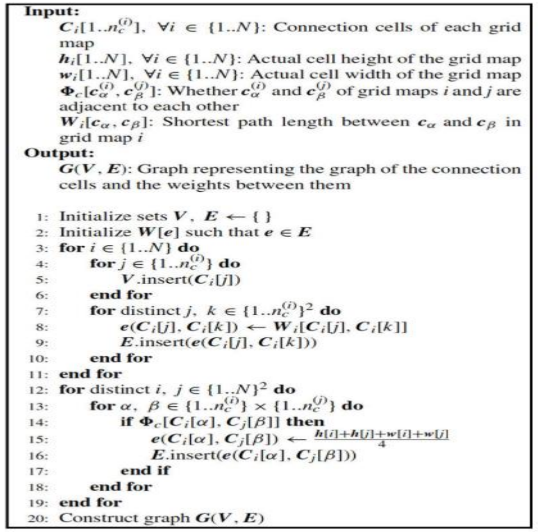
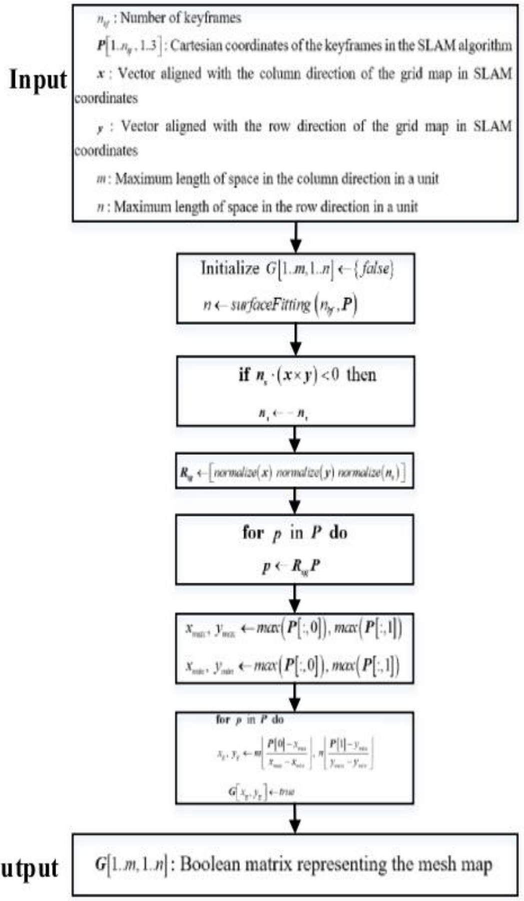


FIGURE 3. a) A demonstration of mn-scaled meshing with SLAM map and b) real-time farm view.



```
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 $c_\alpha$ and $c_\beta$ in grid map $i$
Output:
    $G(V, E)$: Graph representing the graph of the connection cells and the weights between them

Initialize sets $V$, $E \leftarrow \{\}$
Initialize $W[e]$ such that $e \in E$
for $i \in \{1..N\}$ do
    for $j \in \{1..n_c^{(i)}\}$ do
        $V$.insert($C_i[j]$)
    end for
    for distinct $j, k \in \{1..n_c^{(i)}\}^2$ do
        $e(C_i[j], C_i[k]) \leftarrow W_i[C_i[j], C_i[k]]$
        $E$.insert($e(C_i[j], C_i[k])$)
    end for
end for
for distinct $i, j \in \{1..N\}^2$ do
    for $\alpha, \beta \in \{1..n_c^{(i)}\} \times \{1..n_c^{(j)}\}$ do
        if $\Phi_c[C_i[\alpha], C_j[\beta]]$ then
            $e(C_i[\alpha], C_j[\beta]) \leftarrow \frac{h[i]+h[j]+w[i]+w[j]}{4}$
            $E$.insert($e(C_i[\alpha], C_j[\beta])$)
        end if
    end for
end for
Construct graph $G(V, E)$
```

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



Input

n_{kf} : Number of keyframes

$P[1..n_{kf}, 1..3]$: 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

Initialize $G[1..m, 1..n] \leftarrow \{false\}$

$n \leftarrow surfaceFitting(n_{kf}, P)$

if $n_s \cdot (x \times y) < 0$ then

$n_s \leftarrow -n_s$

$R_{sg} \leftarrow [normalize(x)\ normalize(y)\ normalize(n_s)]$

for p in P do

$p \leftarrow R_{sg} P$

$x_{max}, y_{max} \leftarrow max(P[:,0]), max(P[:,1])$

$x_{min}, y_{min} \leftarrow max(P[:,0]), max(P[:,1])$

for p in P do

$x_g, y_g \leftarrow m\left[\frac{P[0]-x_{min}}{x_{max}-x_{min}}\right], n\left[\frac{P[1]-y_{min}}{y_{max}-y_{min}}\right]$

$G[x_g, y_g] \leftarrow true$

Output

$G[1..m, 1..n]$: Boolean matrix representing 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)

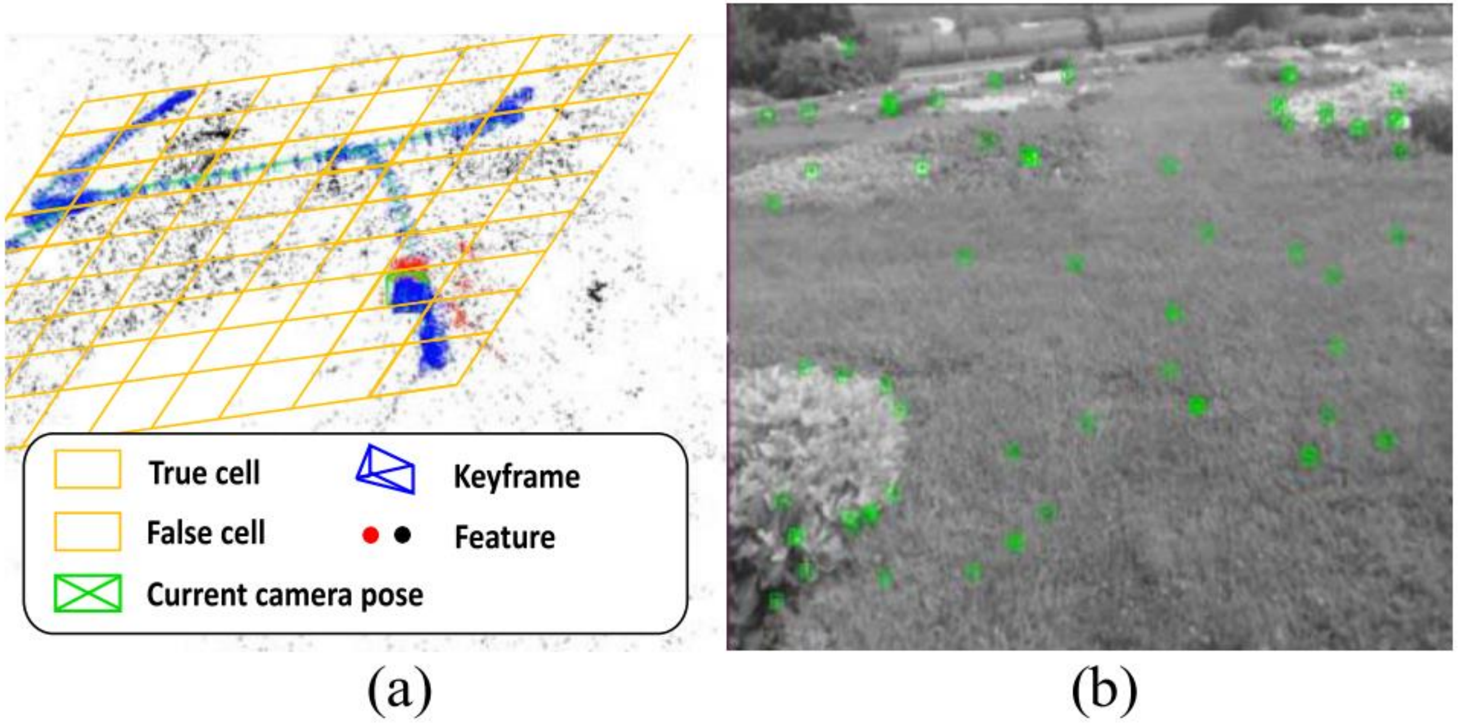
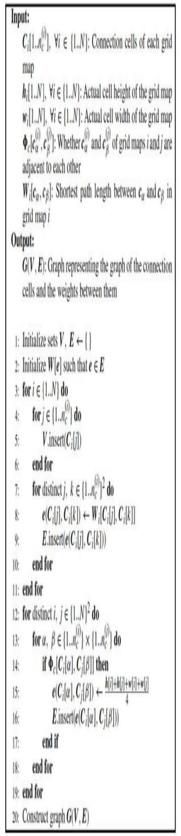


FIGURE 3. a) A demonstration of mn-scaled meshing with SLAM map and b) real-time farm view.



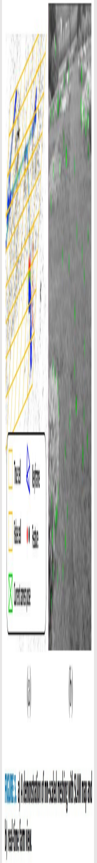
```
Input:
    $C_i[1..n_c^{(i)}]$, $\forall i \in \{1..N\}$: Connection cells of each grid map
    $h_i[1..N]$, $\forall i \in \{1..N\}$: Actual cell height of the grid map
    $w_i[1..N]$, $\forall i \in \{1..N\}$: Actual cell width of the grid map
    $\Phi_c[c_\alpha^{(i)}, c_\beta^{(j)}]$: Whether $c_\alpha^{(i)}$ 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 $c_\alpha$ and $c_\beta$ in grid map $i$
Output:
    $G(V, E)$: Graph representing the graph of the connection cells and the weights between them

Initialize sets $V, E \leftarrow \{\}$
Initialize $W[e]$ such that $e \in E$
for $i \in \{1..N\}$ do
    for $j \in \{1..n_c^{(i)}\}$ do
        $V$.insert($C_i[j]$)
    end for
    for distinct $j, k \in \{1..n_c^{(i)}\}^2$ do
        $e(C_i[j], C_i[k]) \leftarrow W_i[C_i[j], C_i[k]]$
        $E$.insert($e(C_i[j], C_i[k])$)
    end for
end for
for distinct $i, j \in \{1..N\}^2$ do
    for $\alpha, \beta \in \{1..n_c^{(i)}\} \times \{1..n_c^{(j)}\}$ do
        if $\Phi_c[C_i[\alpha], C_j[\beta]]$ then
            $e(C_i[\alpha], C_j[\beta]) \leftarrow \frac{h[i]+h[j]+w[i]+w[j]}{4}$
            $E$.insert($e(C_i[\alpha], C_j[\beta])$)
        end if
    end for
end for
Construct graph $G(V, E)$
```

FIGURE 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)



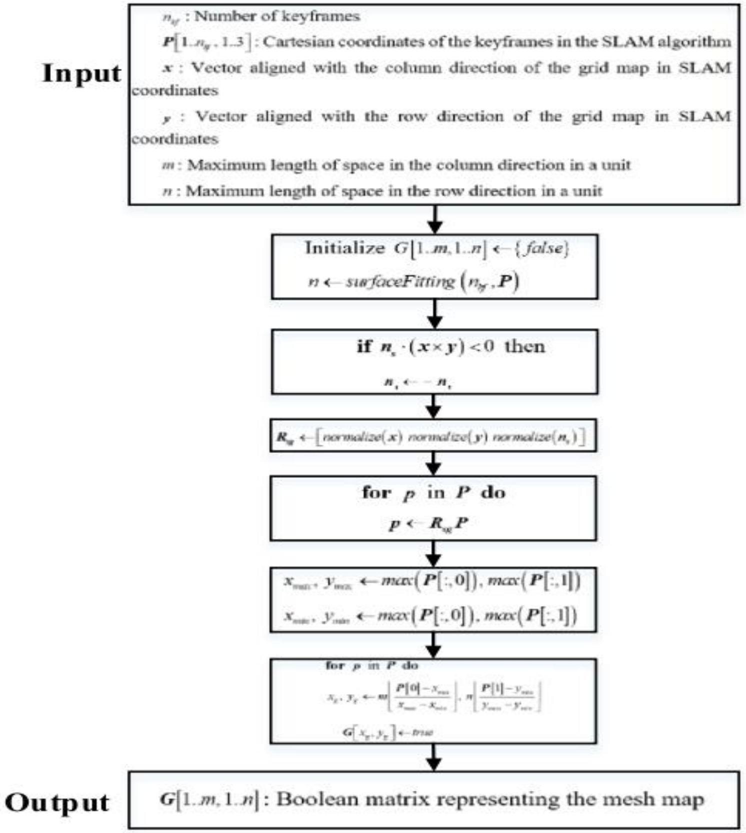
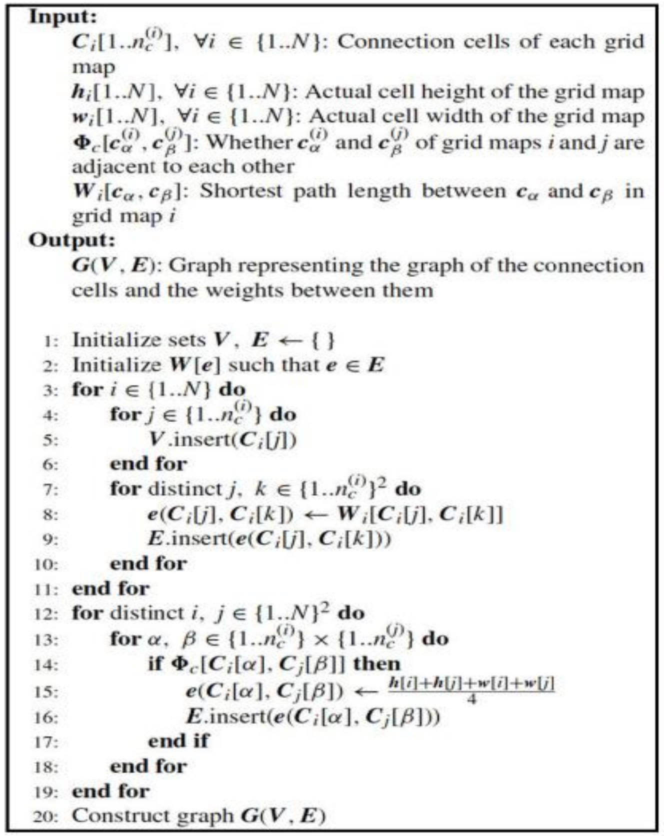

Input
n_{kf} : Number of keyframes
$P[1..n_{kf}, 1..3]$: Cartesian coordinates of the keyframes in the SLAM algorithm
x : Vector aligned with the column direction of the grid map in SLAM coordinates
y : Vector aligned with the row direction of the grid map in SLAM coordinates
m : Maximum length of space in the column direction in a unit
n : Maximum length of space in the row direction in a unit
Initialize $G[1..m, 1..n] \leftarrow \{false\}$
$n \leftarrow surfaceFitting(n_{kf}, P)$
if $n_s \cdot (x \times y) < 0$ then
$n_s \leftarrow -n_s$
$R_{sg} \leftarrow [normalize(x)\ normalize(y)\ normalize(n_s)]$
for p in P do
$p \leftarrow R_{sg} P$
$x_{max}, y_{max} \leftarrow max(P[:,0]), max(P[:,1])$
$x_{min}, y_{min} \leftarrow max(P[:,0]), max(P[:,1])$
for p in P do
$x_g, y_g \leftarrow m\left\lfloor \frac{P[0]-x_{min}}{x_{max}-x_{min}} \right\rfloor, n\left\lfloor \frac{P[1]-y_{min}}{y_{max}-y_{min}} \right\rfloor$
$G[x_g, y_g] \leftarrow true$
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 $c_\alpha$ and $c_\beta$ in grid map $i$
Output:
    $G(V, E)$: Graph representing the graph of the connection cells and the weights between them

Initialize sets $V$, $E \leftarrow \{\}$
Initialize $W[e]$ such that $e \in E$
for $i \in \{1..N\}$ do
    for $j \in \{1..n_c^{(i)}\}$ do
        $V$.insert($C_i[j]$)
    end for
    for distinct $j$, $k \in \{1..n_c^{(i)}\}^2$ do
        $e(C_i[j], C_i[k]) \leftarrow W_i[C_i[j], C_i[k]]$
        $E$.insert($e(C_i[j], C_i[k])$)
    end for
end for
for distinct $i$, $j \in \{1..N\}^2$ do
    for $\alpha$, $\beta \in \{1..n_c^{(i)}\} \times \{1..n_c^{(j)}\}$ do
        if $\Phi_c[C_i[\alpha], C_j[\beta]]$ then
            $e(C_i[\alpha], C_j[\beta]) \leftarrow \frac{h[i]+h[j]+w[i]+w[j]}{4}$
            $E$.insert($e(C_i[\alpha], C_j[\beta])$)
        end if
    end for
end for
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

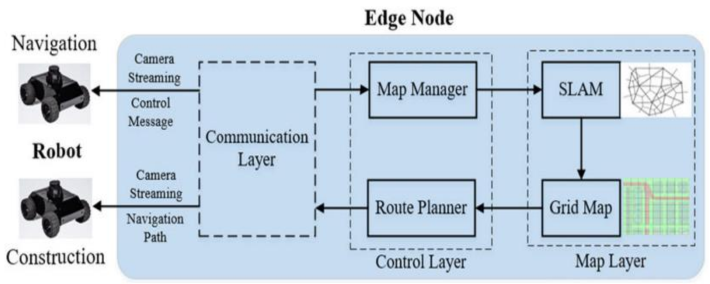
Methodology

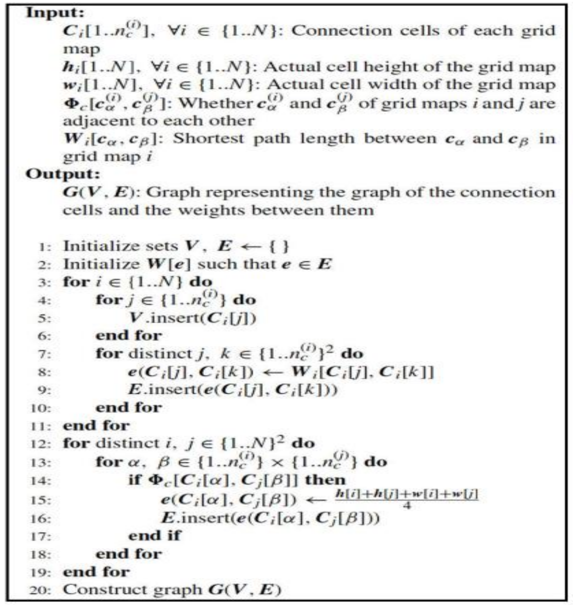
Conclusion

Summary

Motivation/purpose/ aims/hypothesis

Contribution





```
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 $c_\alpha$ and $c_\beta$ in grid map $i$
Output:
    $G(V, E)$: Graph representing the graph of the connection cells and the weights between them

Initialize sets $V$, $E \leftarrow \{\ \}$
Initialize $W[e]$ such that $e \in E$
for $i \in \{1..N\}$ do
    for $j \in \{1..n_c^{(i)}\}$ do
        $V$.insert($C_i[j]$)
    end for
    for distinct $j, k \in \{1..n_c^{(i)}\}^2$ do
        $e(C_i[j], C_i[k]) \leftarrow W_i[C_i[j], C_i[k]]$
        $E$.insert($e(C_i[j], C_i[k])$)
    end for
end for
for distinct $i, j \in \{1..N\}^2$ do
    for $\alpha, \beta \in \{1..n_c^{(i)}\} \times \{1..n_c^{(j)}\}$ do
        if $\Phi_c[C_i[\alpha], C_j[\beta]]$ then
            $e(C_i[\alpha], C_j[\beta]) \leftarrow \frac{h[i]+h[j]+w[i]+w[j]}{4}$
            $E$.insert($e(C_i[\alpha], C_j[\beta])$)
        end if
    end for
end for
Construct graph $G(V, E)$
```

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

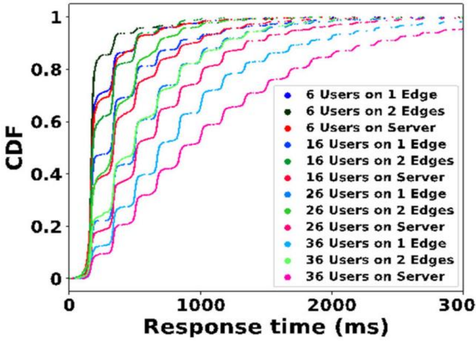


FIGURE 12. The CDF of the time intervals between responses. Note: User means a working robot.

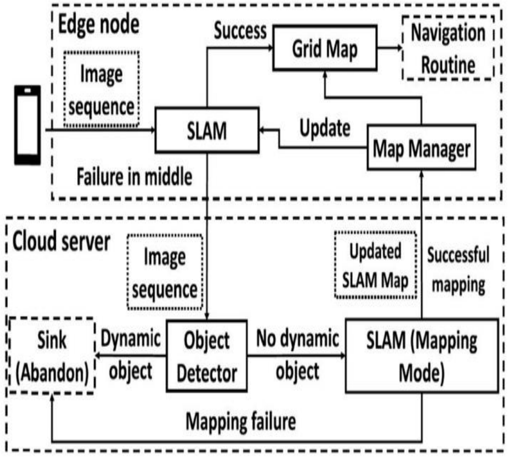


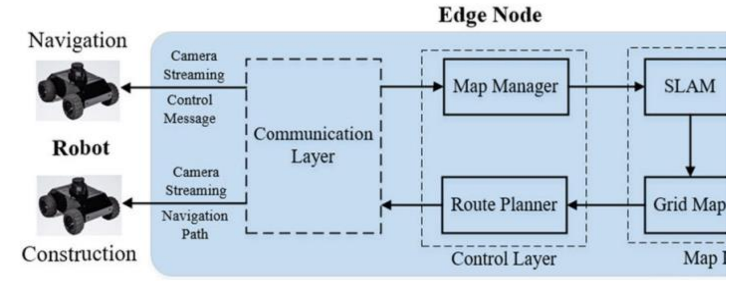
FIGURE 9. The flowchart of map maintenance.

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)



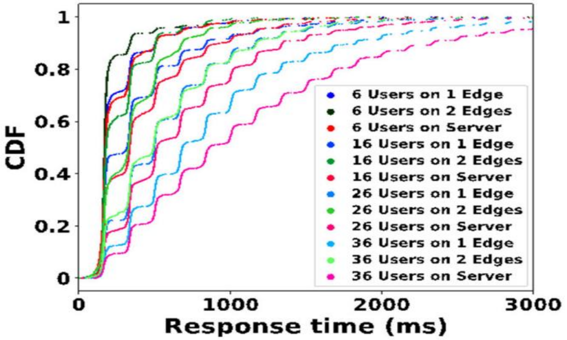
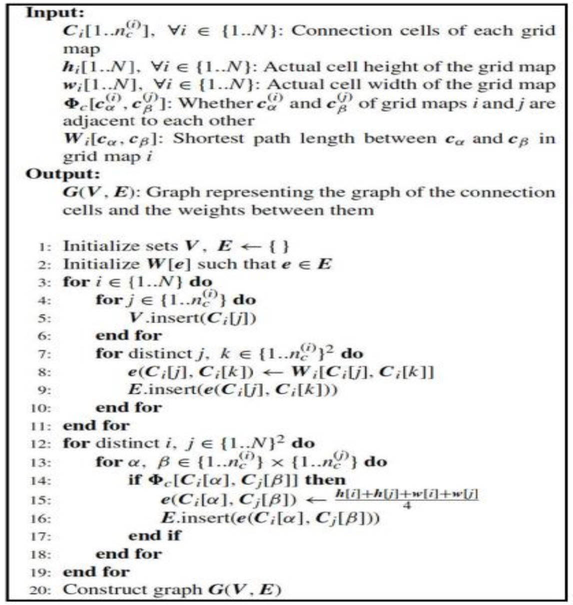


FIGURE 12. The CDF of the time intervals between responses. Note: User means a working robot.



```
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 $c_\alpha$ and $c_\beta$ in grid map $i$
Output:
    $G(V, E)$: Graph representing the graph of the connection cells and the weights between them

Initialize sets $V, E \leftarrow \{\}$
Initialize $W[e]$ such that $e \in E$
for $i \in \{1..N\}$ do
    for $j \in \{1..n_c^{(i)}\}$ do
        $V$.insert($C_i[j]$)
    end for
    for distinct $j, k \in \{1..n_c^{(i)}\}^2$ do
        $e(C_i[j], C_i[k]) \leftarrow W_i[C_i[j], C_i[k]]$
        $E$.insert($e(C_i[j], C_i[k])$)
    end for
end for
for distinct $i, j \in \{1..N\}^2$ do
    for $\alpha, \beta \in \{1..n_c^{(i)}\} \times \{1..n_c^{(j)}\}$ do
        if $\Phi_c[C_i[\alpha], C_j[\beta]]$ then
            $e(C_i[\alpha], C_j[\beta]) \leftarrow \frac{h[i]+h[j]+w[i]+w[j]}{4}$
            $E$.insert($e(C_i[\alpha], C_j[\beta])$)
        end if
    end for
end for
Construct graph $G(V, E)$
```

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

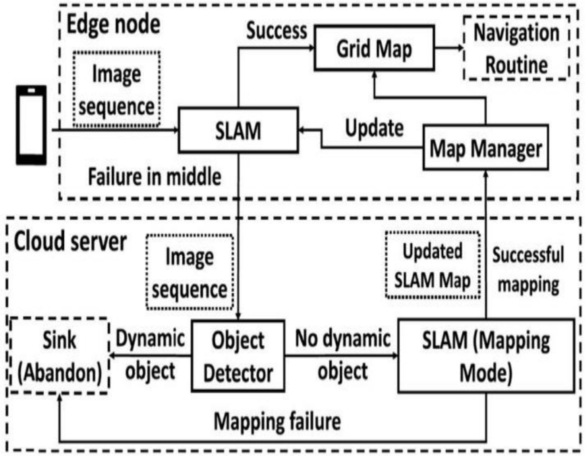


FIGURE 9. The flowchart of map maintenance.

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)

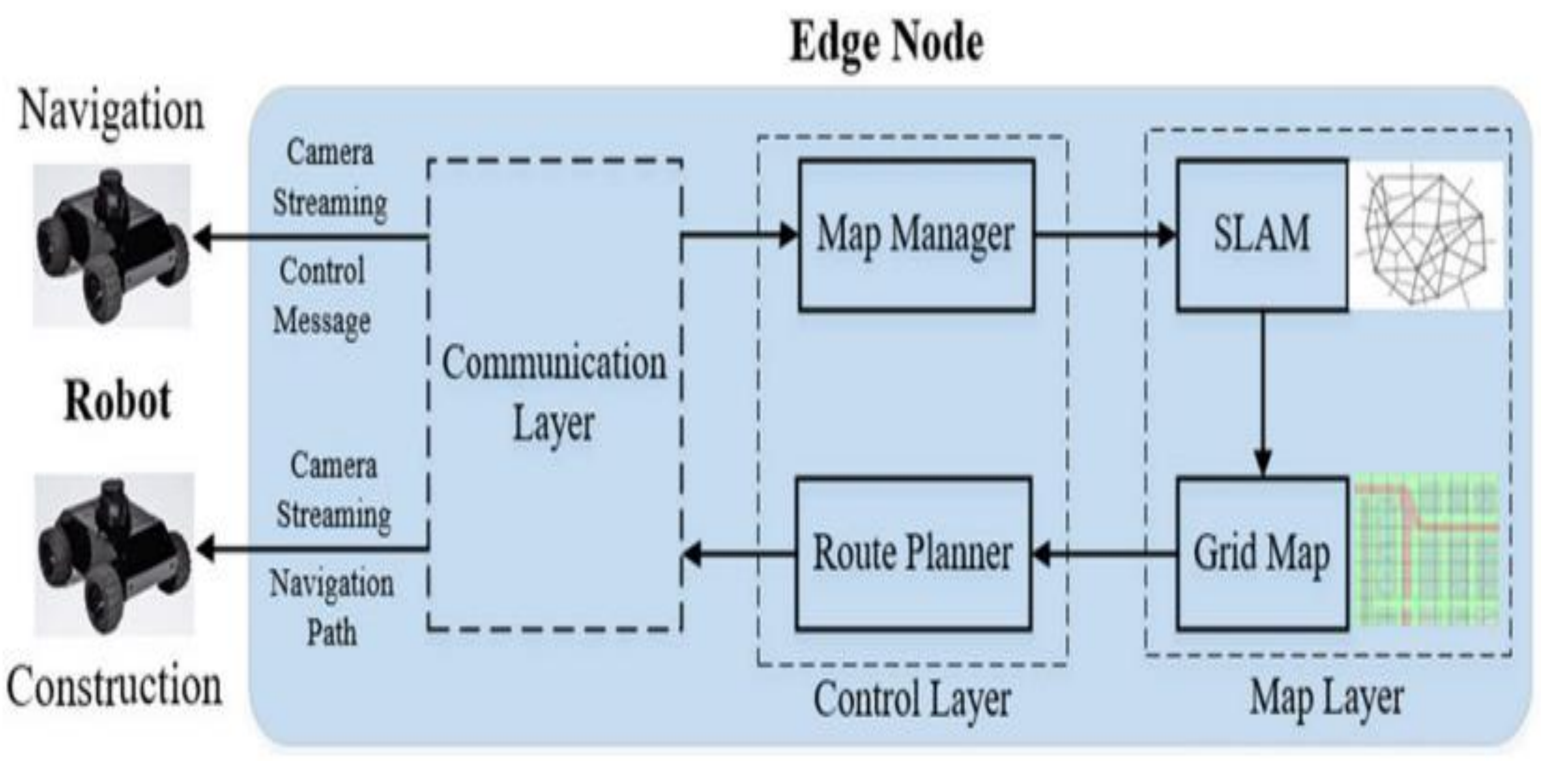
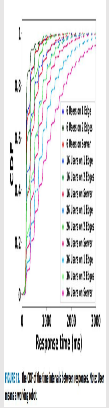
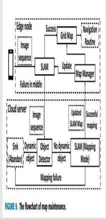
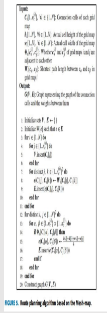


FIGURE 2. The Structure of the Edge Node Layer.



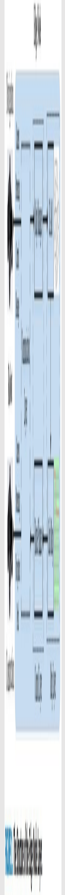


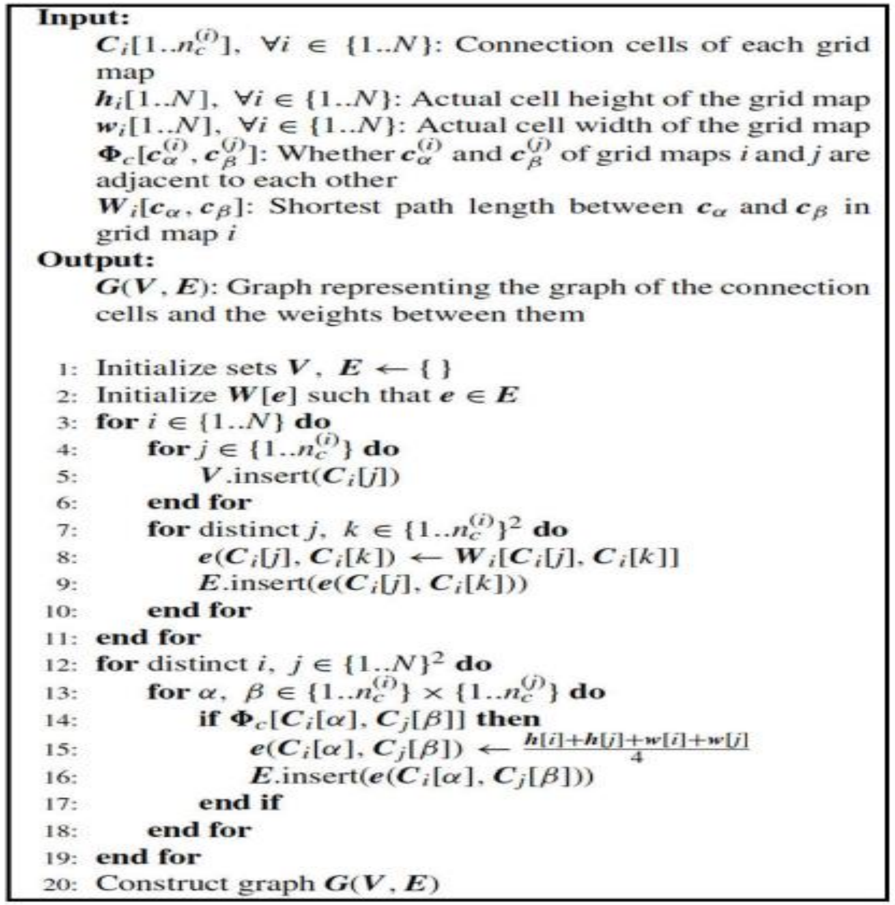


Summary

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 $c_\alpha$ and $c_\beta$ in grid map $i$
Output:
    $G(V, E)$: Graph representing the graph of the connection cells and the weights between them

Initialize sets $V, E \leftarrow \{\}$
Initialize $W[e]$ such that $e \in E$
for $i \in \{1..N\}$ do
    for $j \in \{1..n_c^{(i)}\}$ do
        $V$.insert($C_i[j]$)
    end for
    for distinct $j, k \in \{1..n_c^{(i)}\}^2$ do
        $e(C_i[j], C_i[k]) \leftarrow W_i[C_i[j], C_i[k]]$
        $E$.insert($e(C_i[j], C_i[k])$)
    end for
end for
for distinct $i, j \in \{1..N\}^2$ do
    for $\alpha, \beta \in \{1..n_c^{(i)}\} \times \{1..n_c^{(j)}\}$ do
        if $\Phi_c[C_i[\alpha], C_j[\beta]]$ then
            $e(C_i[\alpha], C_j[\beta]) \leftarrow \frac{h[i]+h[j]+w[i]+w[j]}{4}$
            $E$.insert($e(C_i[\alpha], C_j[\beta])$)
        end if
    end for
end for
Construct graph $G(V, E)$
```

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

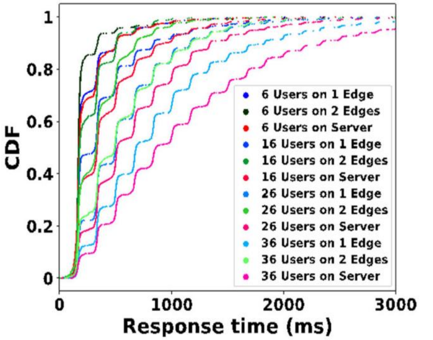


FIGURE 12. The CDF of the time intervals between responses. Note: User means a working robot.

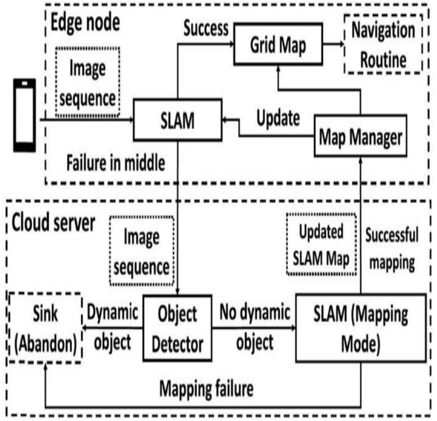
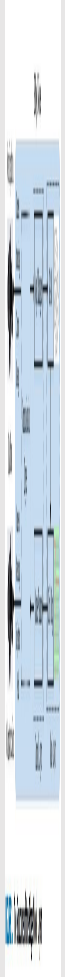


FIGURE 9. The flowchart of map maintenance.

Summary

Methodology

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



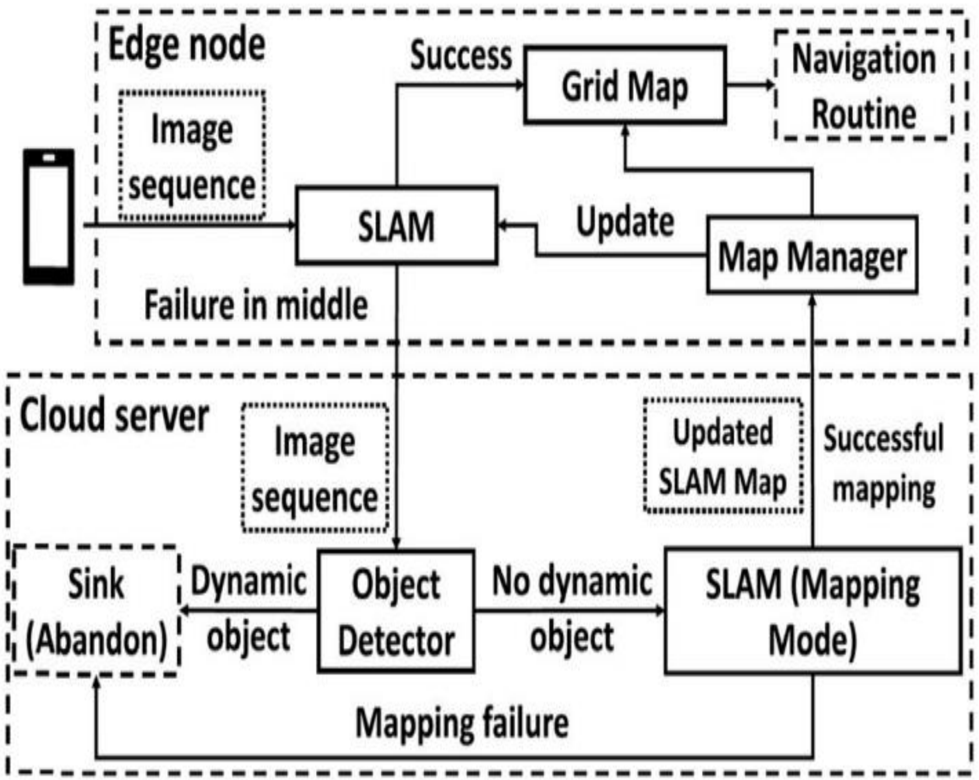


FIGURE 9. The flowchart of map maintenance.

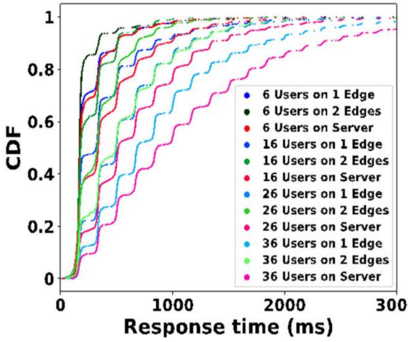


FIGURE 12. The CDF of the time intervals between responses. Note: User means a working robot.

Summary

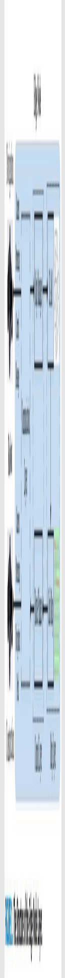
Methodology

•Monocular cameras, SLAM, mesh maps (Figs. 2, 5)

•Accuracy, CPU usage, localization

•experiments (Figs. 9, 12)

•Real-time mapping for precision agriculture



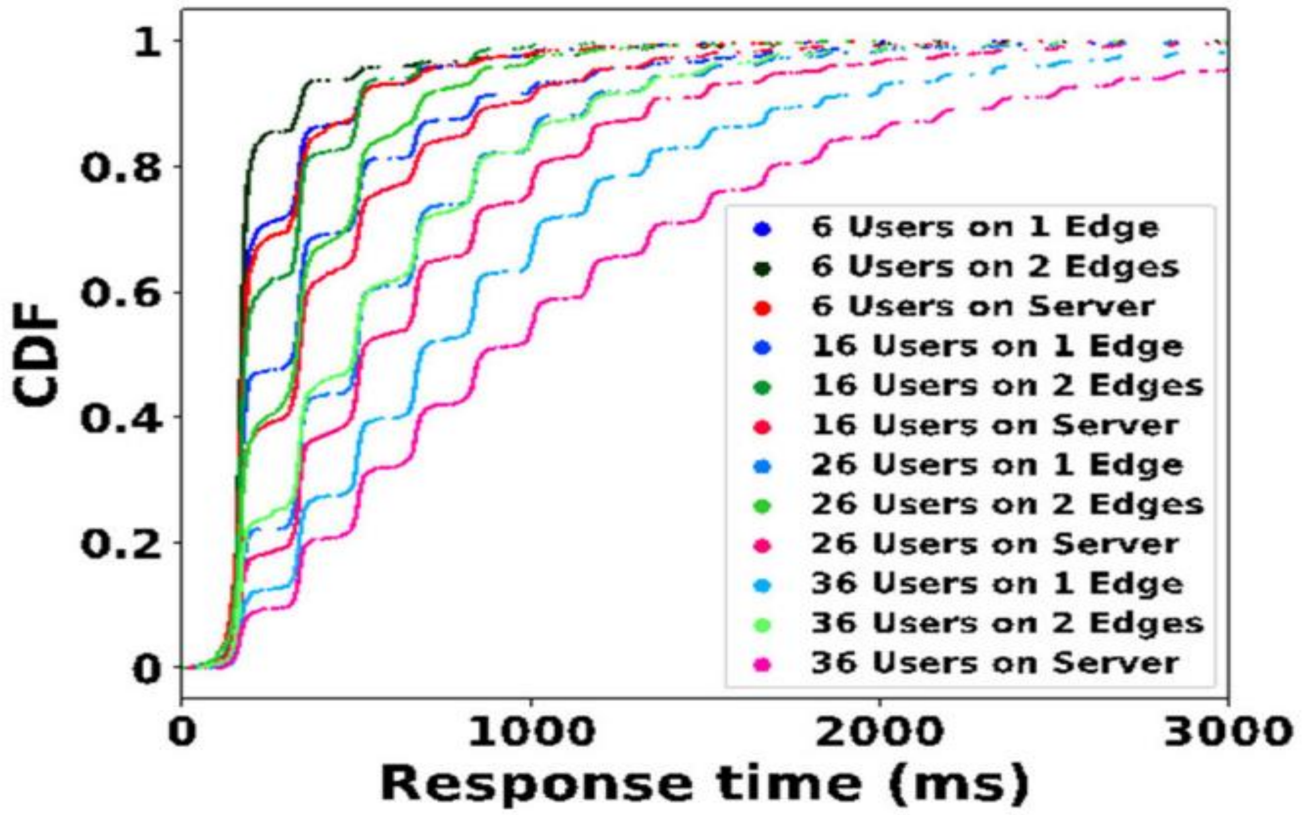


FIGURE 12. The CDF of the time intervals between responses. Note: User means a working robot.

Summary

Motivation/purpose/ aims/hypothesis

Contribution

Methodology

Conclusion

Summary

Motivation/purpose/ aims/hypothesis

Contribution

Methodology

Conclusion

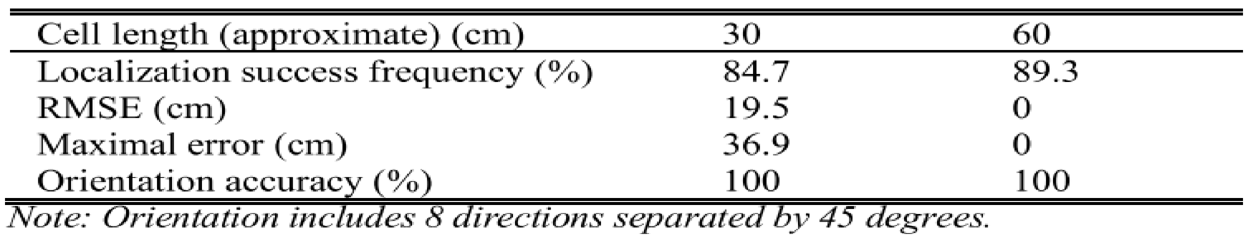
Summary

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.

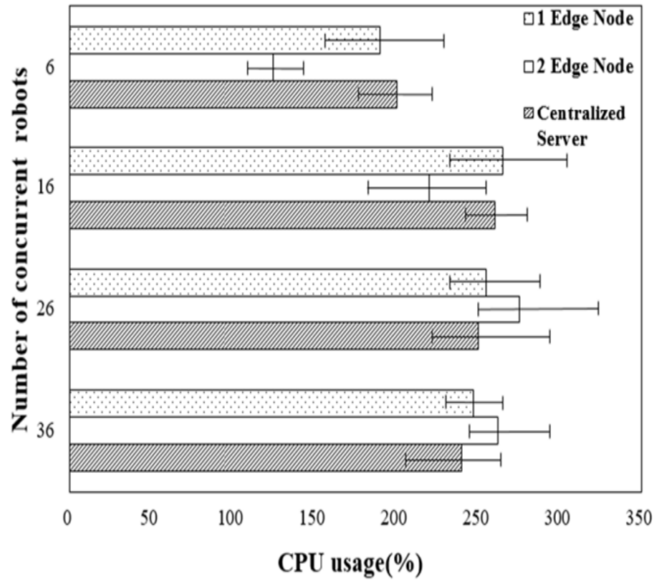
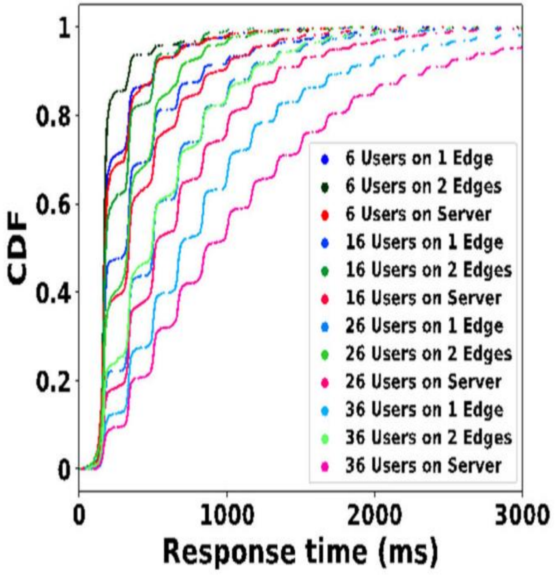


FIGURE 13. CPU usages in each experiment configuration.



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

Summary

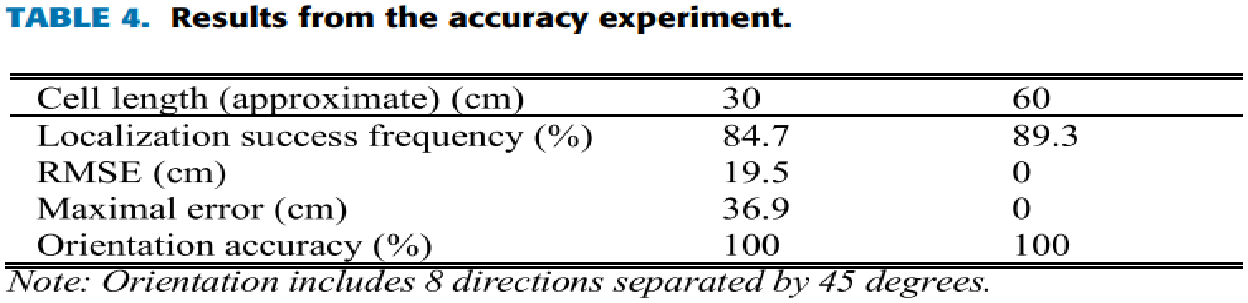


TABLE 4. Results from the accuracy experiment.

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.

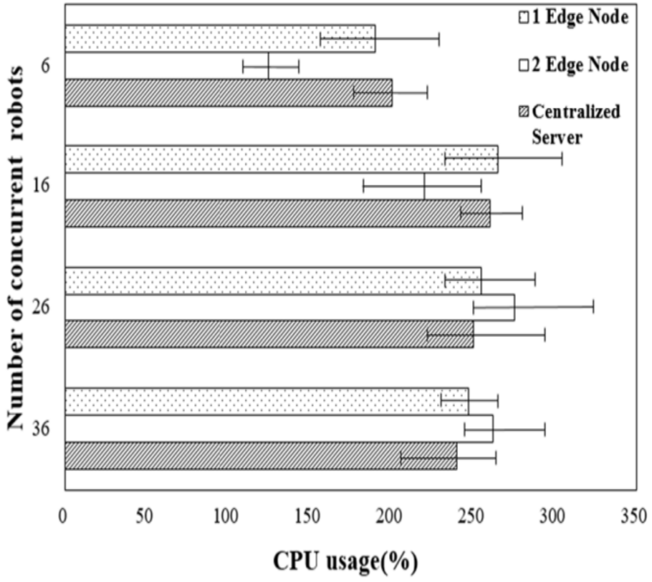
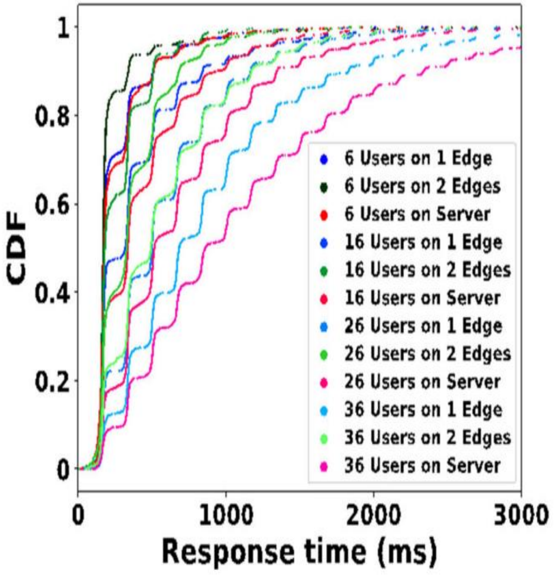


FIGURE 13. CPU usages in each experiment configuration.



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

Conclusion

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

Summary

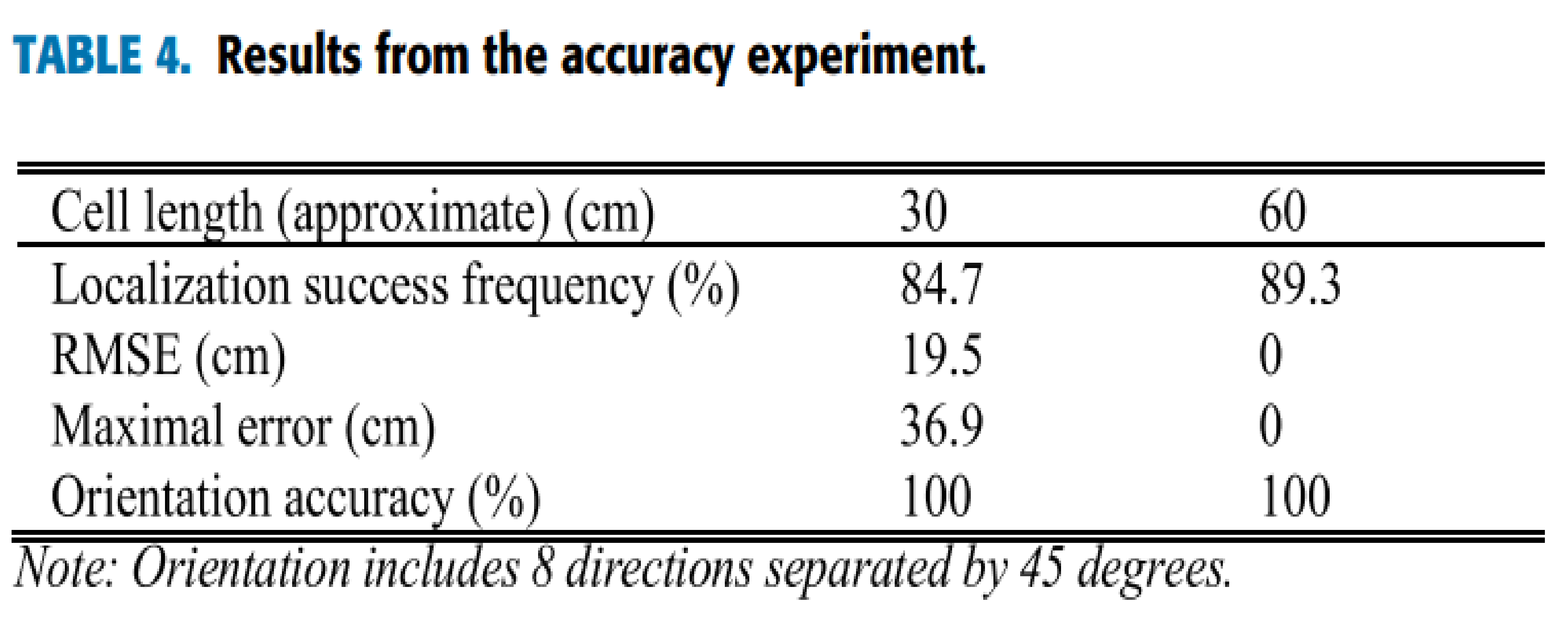


TABLE 4. Results from the accuracy experiment.

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.

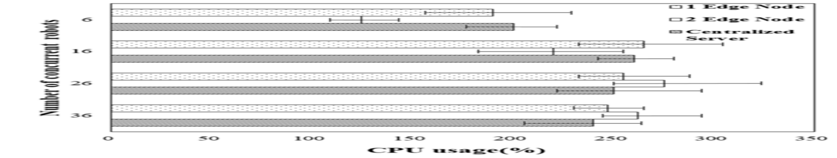


FIGURE 13. CPU usages in each experiment configuration.

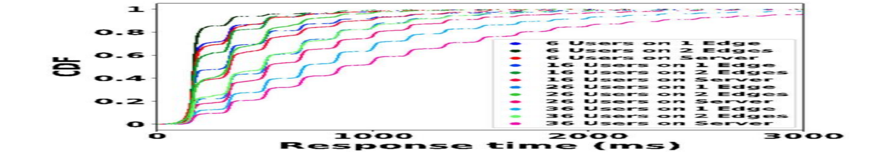


FIGURE 12. The CDF of the time intervals between responses. Note: User means a working robot.

Conclusion

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

Summary

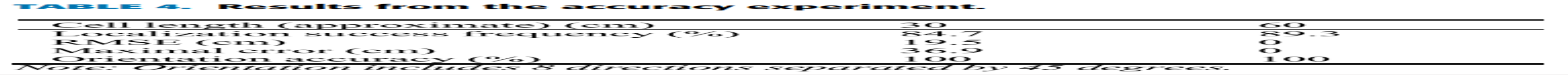


TABLE 4. Results from the accuracy experiment.

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.

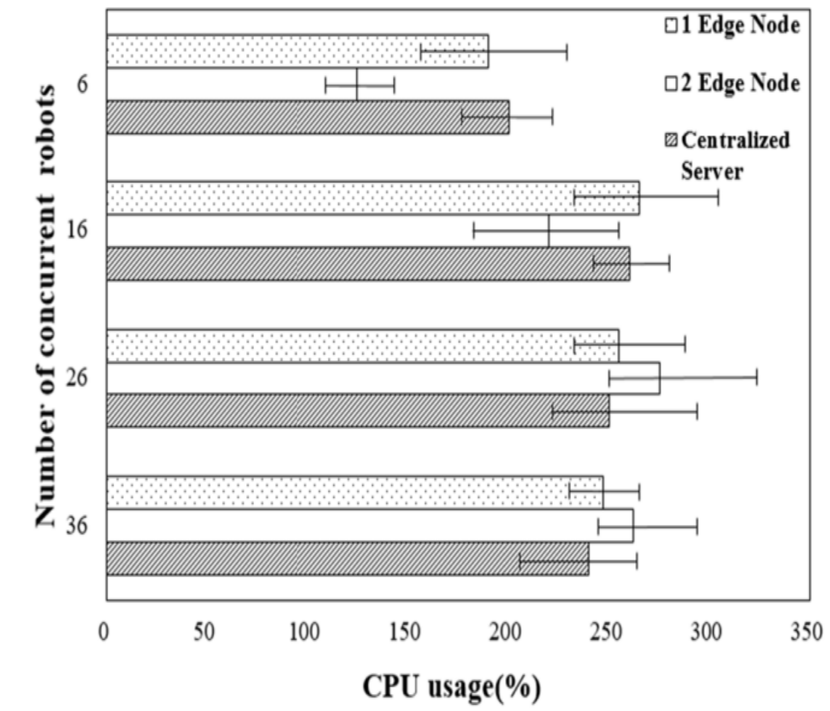


FIGURE 13. CPU usages in each experiment configuration.

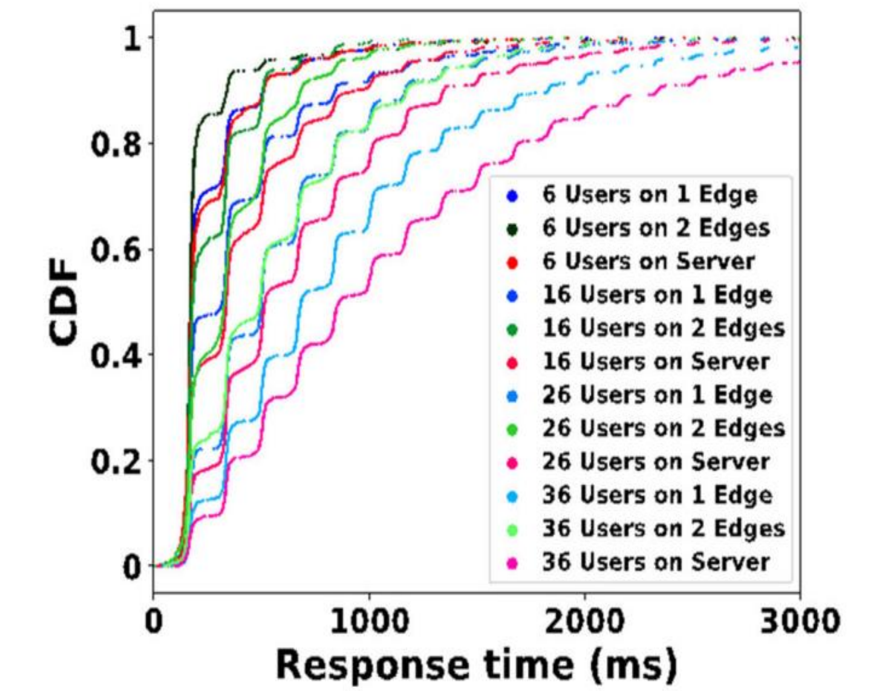


FIGURE 12. The CDF of the time intervals between responses. Note: User means a working robot.

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

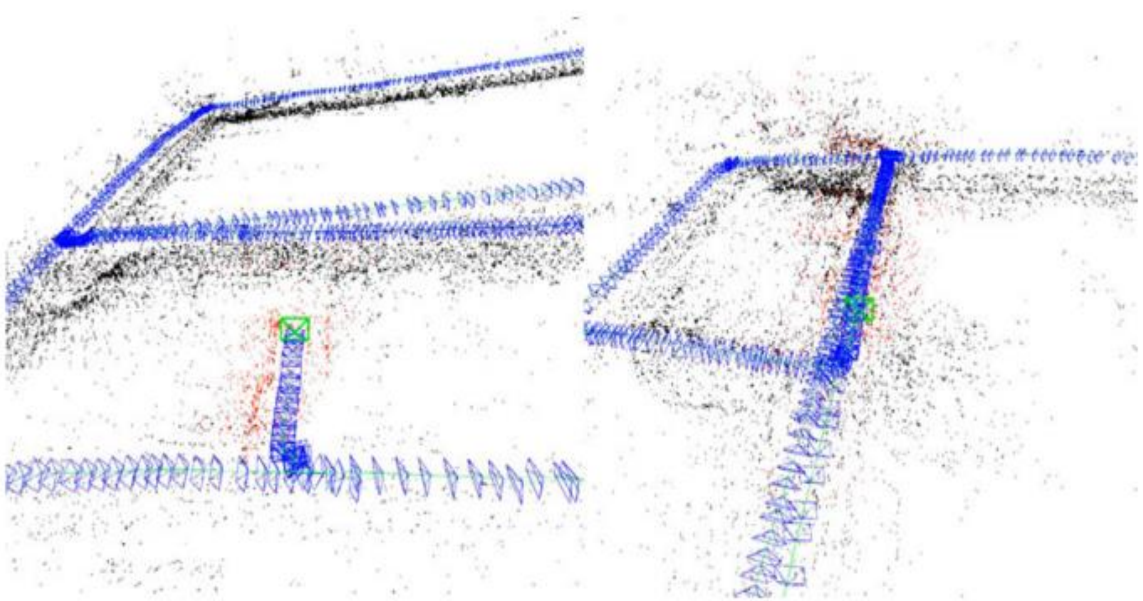


FIGURE 11. Building the SLAM map including (left) the map before path merging and (right) after path merging.

Limitation 2

SLAM technology limitations

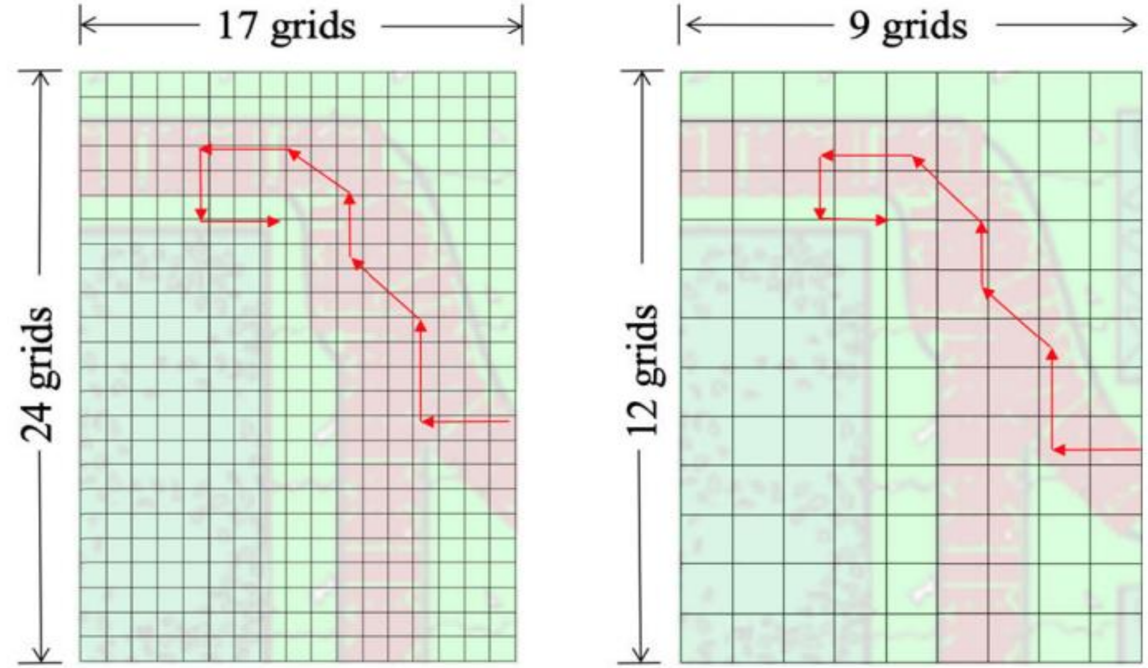


FIGURE 10. The configuration of the accuracy experiment with a unit of 30cm-cells (left) and 60cm-cells (right).

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).