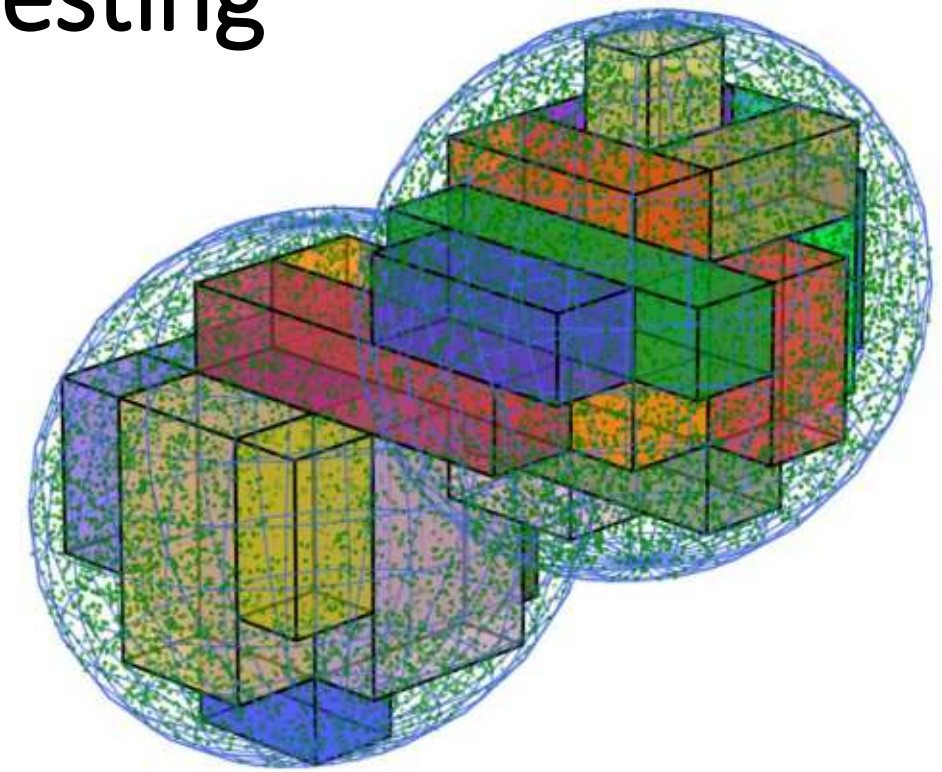
NIST Workshop on Privacy-Enhancing Cryptography 2026
Presented on 27 October 2026

Privacy-preserving Proximity Testing

from Geometric Fuzzy Matching

Florias Papadopoulos

University of St. Gallen, Switzerland



*This research is supported by armasuisse Science and Technology

Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra

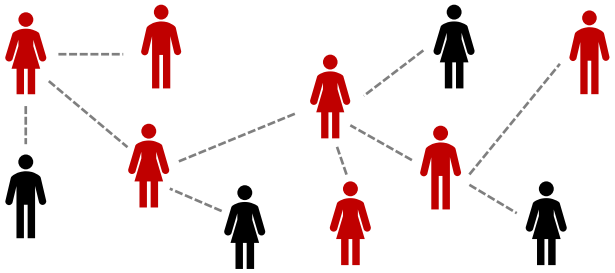
armasuisse Science and Technology
Cyber-Defence Campus

University of St.Gallen
Institute of Computer Science

Based on: *Privacy-preserving Proximity Testing from Geometric Fuzzy Matching*
I. Katis, K. Mitrokotsa, **F. Papadopoulos** (Published in ACM AsiaCCS 2026)

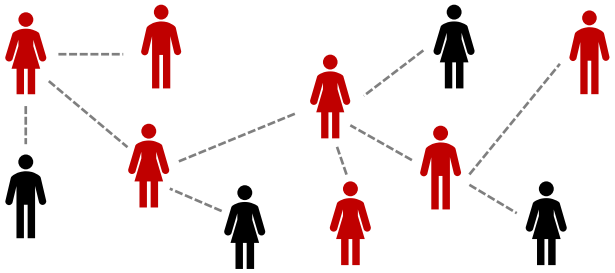
Scenarios Demanding Privacy in Proximity Testing

Contact Tracing



Scenarios Demanding Privacy in Proximity Testing

Contact Tracing



Privacy-preserving Ridesharing



Scenario #2: Privacy-preserving Ridesharing

- Can such an application be useful for the military too, directly?

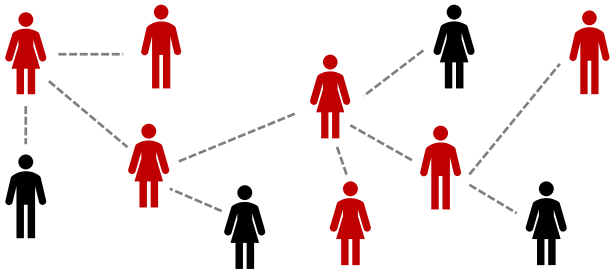


What is uberMILITARY?

UberMILITARY is a product offering available in Honolulu that matches riders with driver-partners who have base access, meaning you can begin and end your trip on base.

Scenarios Demanding Privacy in Proximity Testing

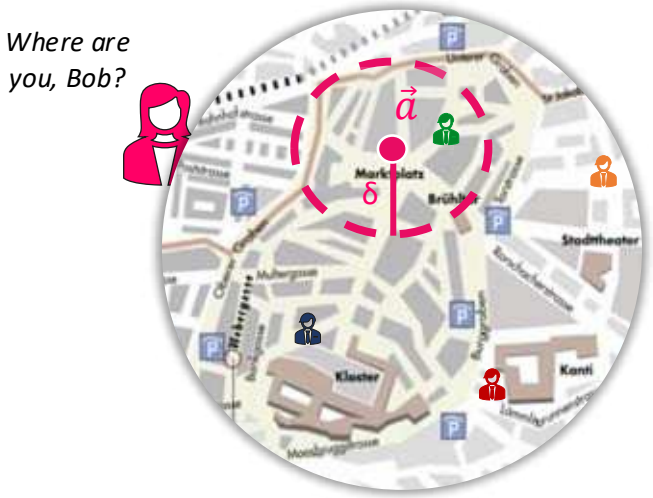
Contact Tracing



Privacy-preserving Ridesharing



Nearby friends' discovery



Nearby friends' discovery

- Current privacy-preserving methods

Have weak security models

Are inefficient

Rely on strong trust assumptions



Nearby friends' discovery

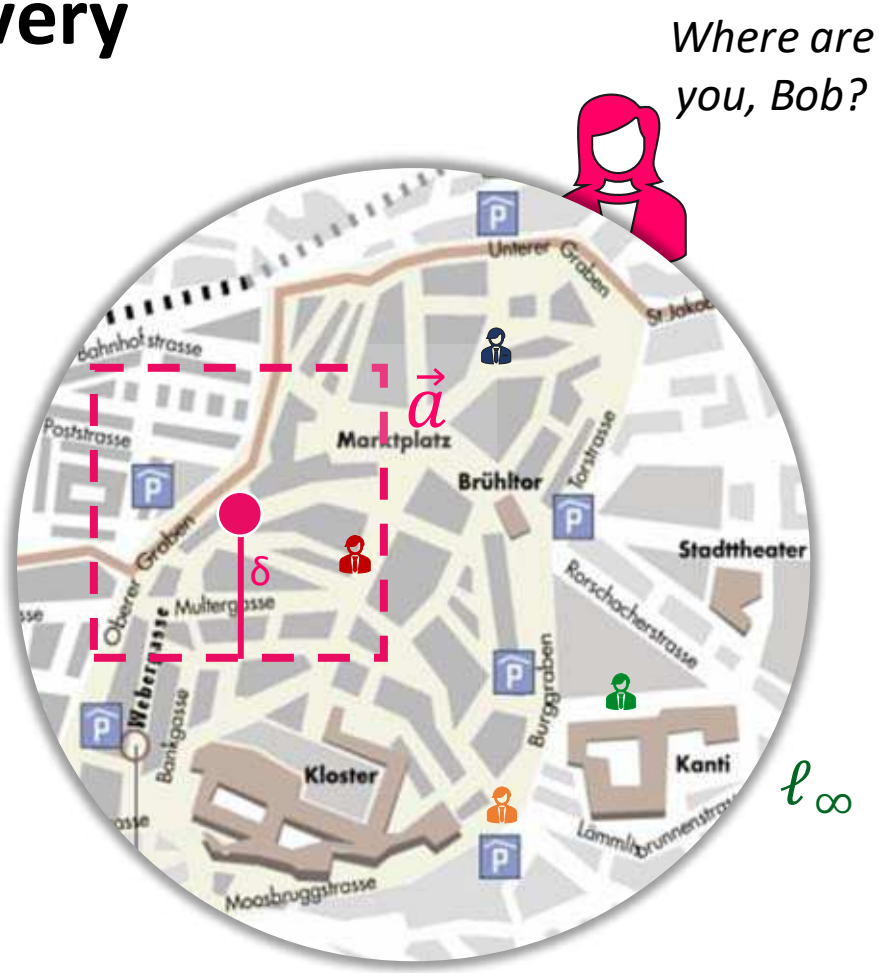
- Current privacy-preserving methods

Have weak security models

Are inefficient

Rely on strong trust assumptions

- In practice, companies rely on insecure systems that endanger user privacy and security.



Nearby friends' discovery

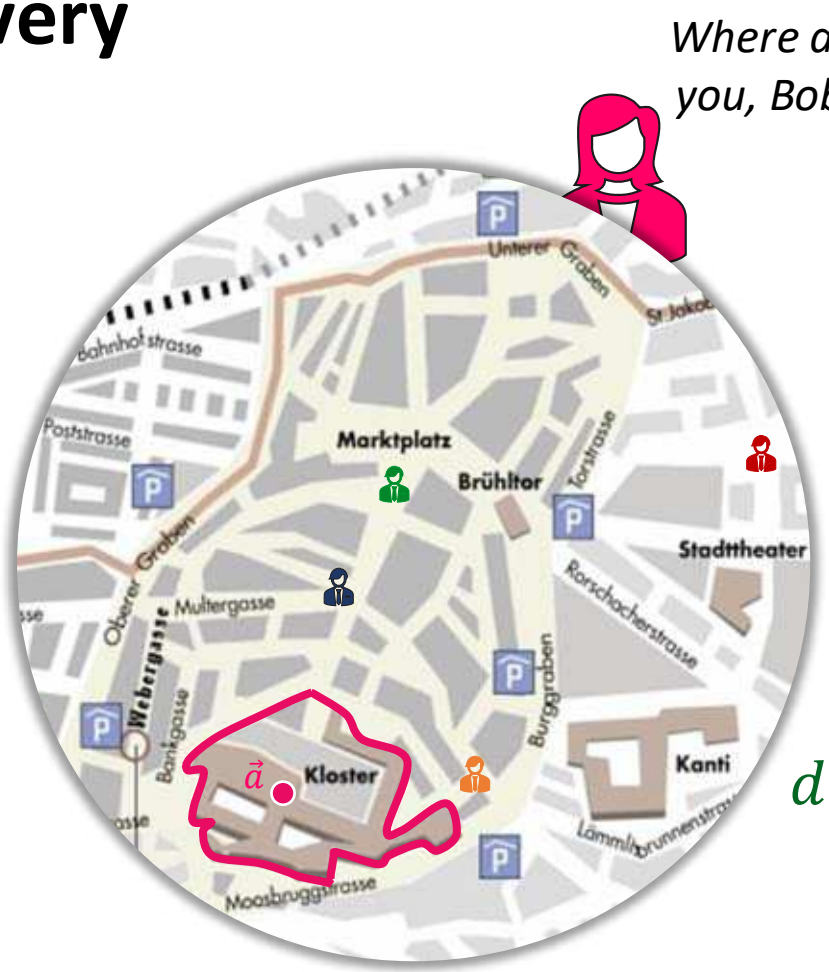
- Current privacy-preserving methods

Have weak security models

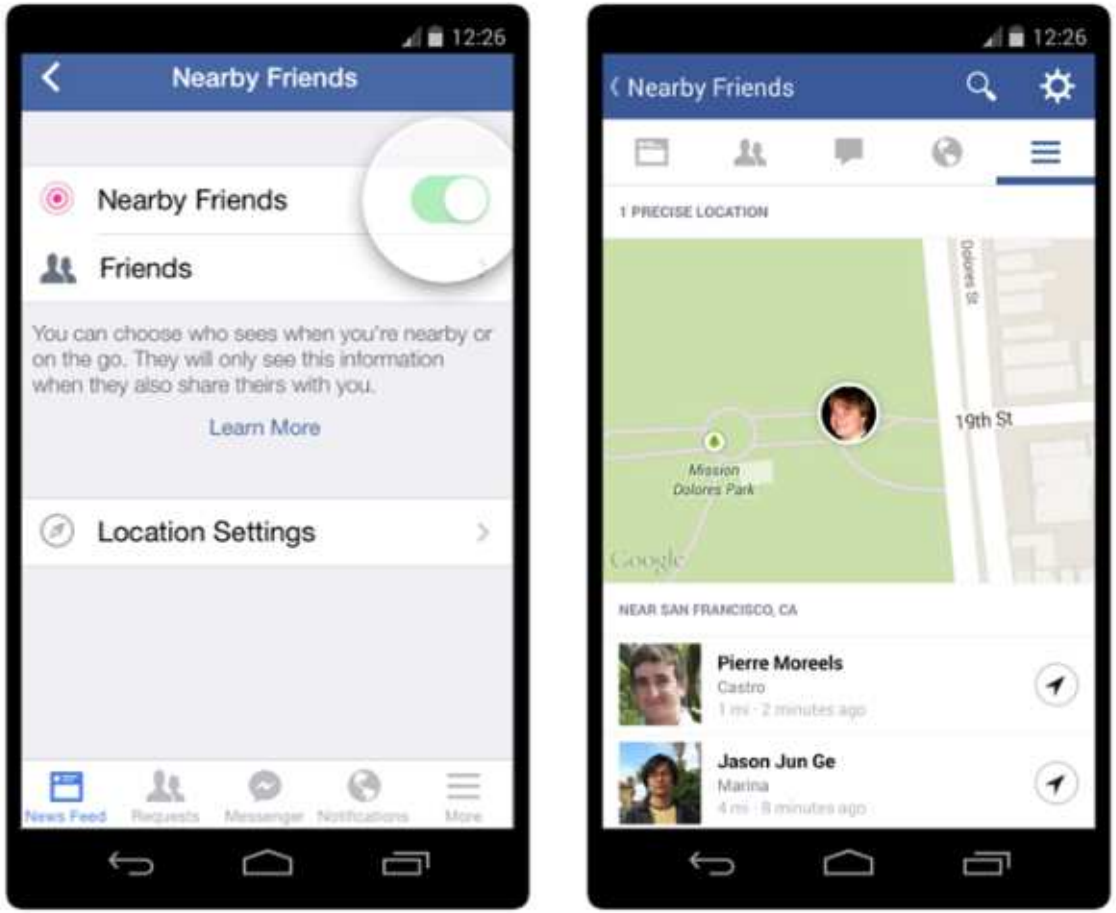
Rely on strong trust assumptions

Are inefficient

- In practice, companies rely on insecure systems that endanger user privacy and security.



Efficient, privacy-preserving, and accurate proximity discovery at scale remains an open problem.



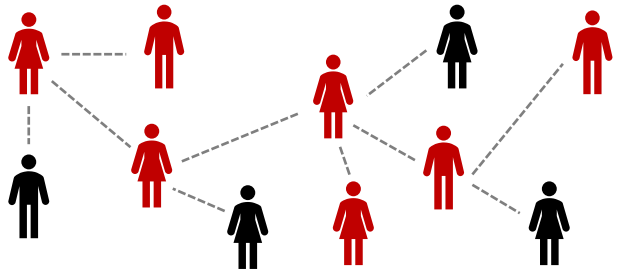
Facebook launches friend-tracking feature

Heather Kelly, CNN

3 minute read · Updated 12:18 PM EDT, Fri April 18, 2014

Scenarios Demanding Privacy in Proximity Testing

Contact Tracing



Privacy-preserving Ridesharing



Nearby friends' discovery

Where are you, Bob?



Drone Proximity & Collision Avoidance



*

*Tedeschi et al., ACSAC '23

Drone Proximity & Collision Avoidance

- Main difference from PPPT:

  "Users" are moving

  ↓

  Solution

  "Acceptable delays" added to the PPPT protocols

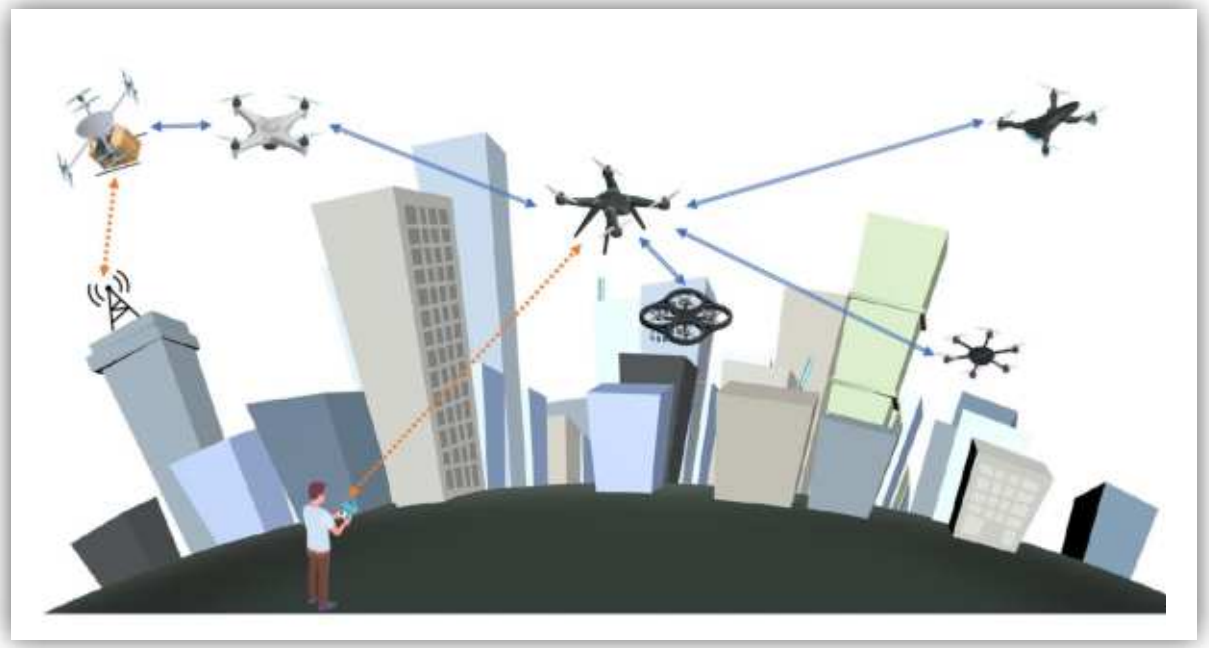


Fig.1: PPPT for remotely-controlled drones and UAVs

Drone Proximity & Collision Avoidance

- Main difference from PPPT:

  "Users" are moving

  ↓

  Solution

  "Acceptable delays" added to the PPPT protocols

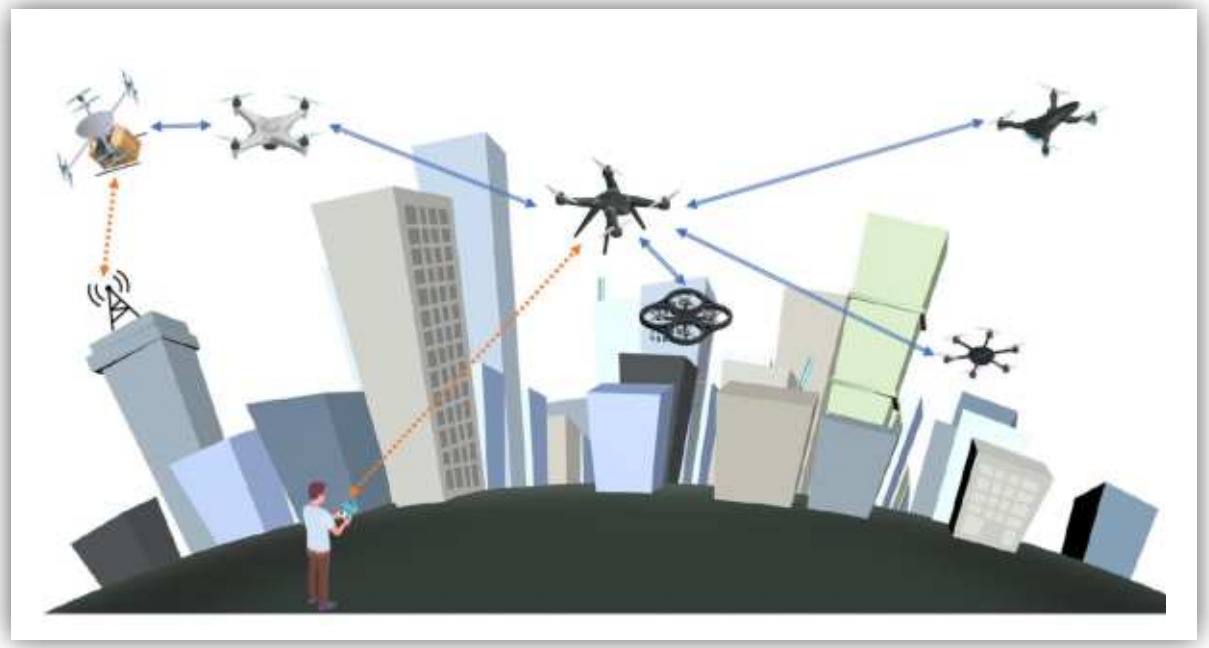


Fig.1: PPPT for remotely-controlled drones and UAVs

- Current privacy-preserving methods

  - Possible Leakage
  - Efficient
  - Non-standard security proof

Drone Proximity & Collision Avoidance

- Main difference from PPPT:

  "Users" are moving

  ↓

  Solution

  "Acceptable delays" added to the PPPT protocols

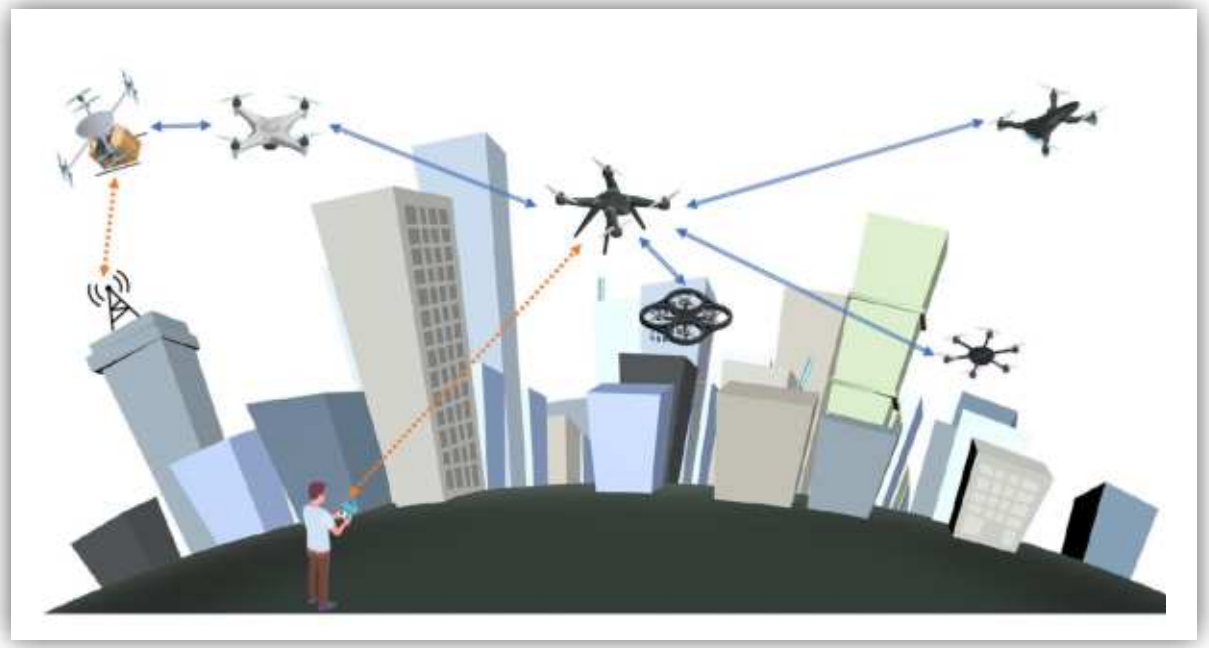


Fig.1: PPPT for remotely-controlled drones and UAVs

- Current privacy-preserving methods

  - Possible Leakage
  - Efficient
  - Non-standard security proof

Efficient, privacy-preserving, and secure collision avoidance for UAVs remains an open problem.

PPPT from Geometric Fuzzy Matching

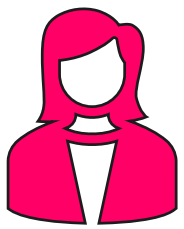
Related work on Fuzzy Matching

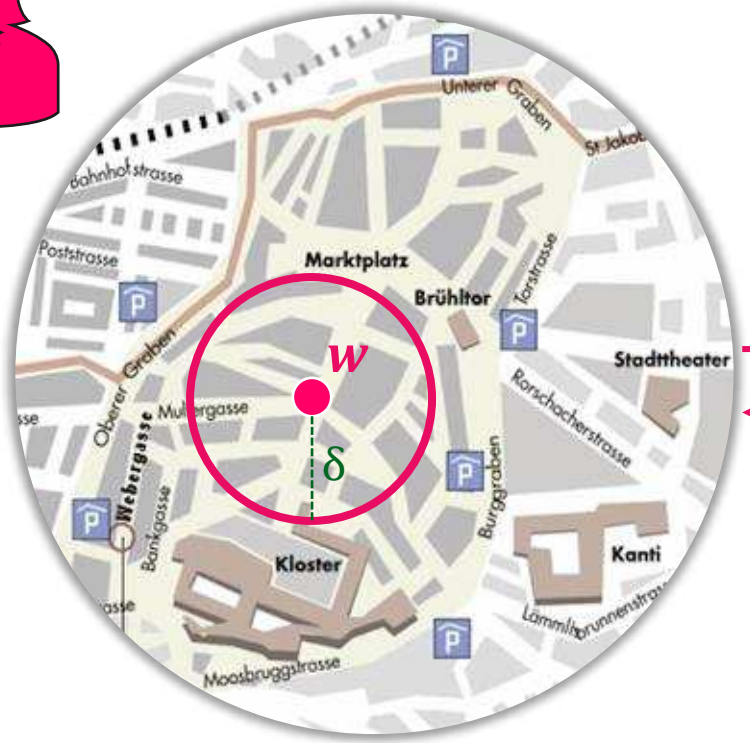
Geometric Fuzzy Matching

Applications of GFM

Related Work on Fuzzy Matching

Fuzzy Matching





$\boldsymbol{w} \in \mathbb{Z}^{\boldsymbol{d}}$

$1\ or\ 0$

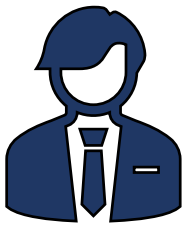
FM

$\boldsymbol{q} \in \mathbb{Z}^{\boldsymbol{d}}$

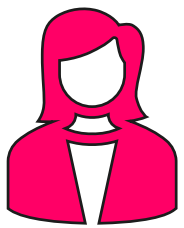
δ: threshold

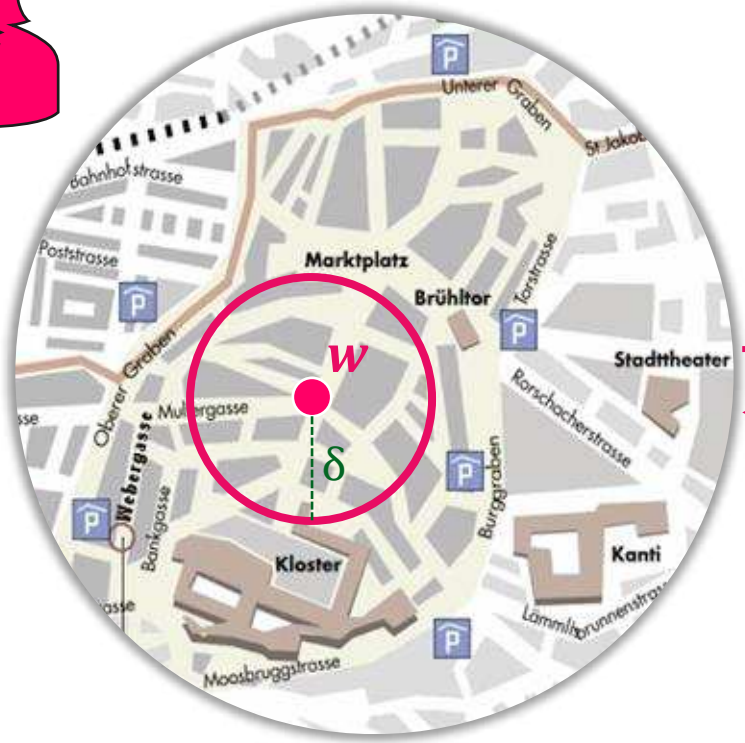
$dist(\cdot,\cdot)$: distance function





Fuzzy Matching





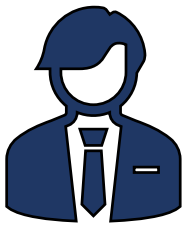
$\boldsymbol{w} \in \mathbb{Z}^{\boldsymbol{d}}$

$1\ or\ 0$

FM

$\boldsymbol{q} \in \mathbb{Z}^{\boldsymbol{d}}$





δ: threshold

$dist(\cdot,\cdot)$: distance function

ℓ_∞

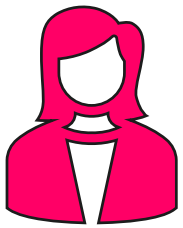
$$dist_\infty(\boldsymbol{w},\boldsymbol{q}) = \max_i(|w_i - q_i|)$$

Hamming

ℓ_p

$$dist_\mathrm{p}(\boldsymbol{w},\boldsymbol{q}) = \left(\sum_{i=1}^{n} |w_i - q_i|^\mathrm{p}\right)^{1/\mathrm{p}}$$

Fuzzy Matching





$w \in \mathbb{Z}^d$

FM

$1\ or\ 0$

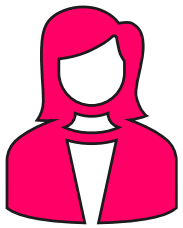
$q \in \mathbb{Z}^d$





	[UCK+21] USENIX	[GRS22] CRYPTO	[CFR23] USENIX	[vBP24] EUROCRYPT	[GQL+24] ASIACRYPT	[vBP25] ASIACRYPT	[SPK+25] IEEE Access	[PST+25] CCS	[DZL25] CCS
dist	Hamming	ℓ_∞	Hamming, $\ell_1\ (d=1)$	$\ell_{\mathrm{p},\ \mathrm{p}\in[1,\infty)}$	Hamming, $\ell_{\mathrm{p},\ \mathrm{p}\in[1,\infty)}$	$\ell_{\mathrm{p},\ \mathrm{p}\in[1,\infty)}$	Cosine Similarity	$\ell_\infty, \ell_1, \ell_2 (d=2)$	$\ell_{\mathrm{p},\ \mathrm{p}\in[1,\infty)}$

Fuzzy Matching





$w \in \mathbb{Z}^d$

1 or 0

FM

$q \in \mathbb{Z}^d$





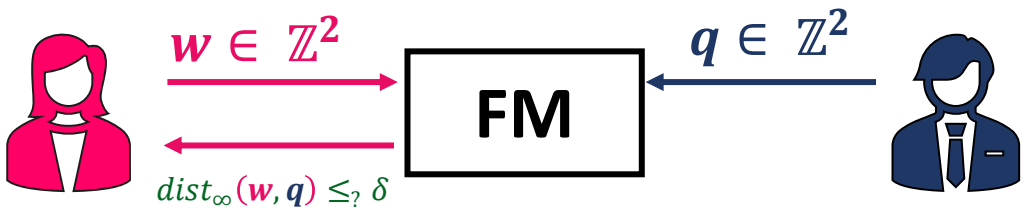
focus was fuzzy PSI*

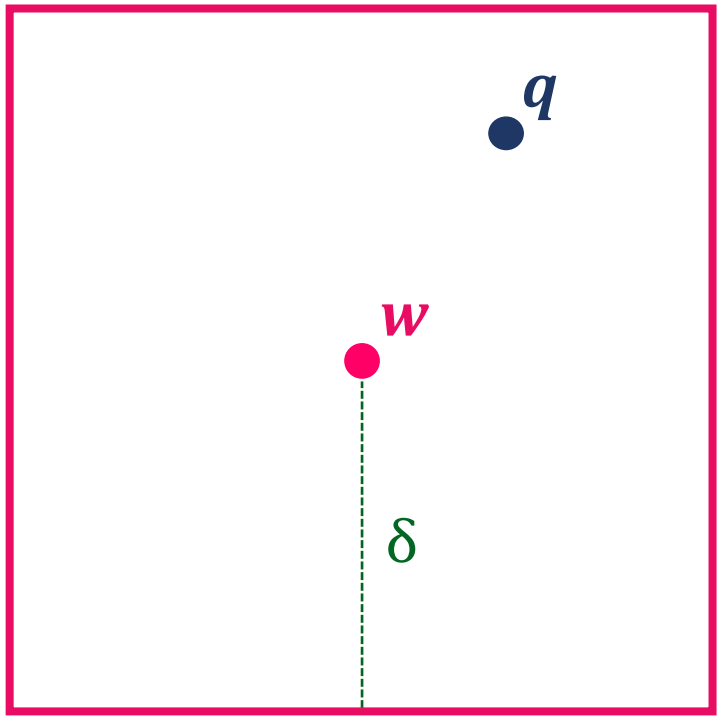
concurrent/later work

	[UCK+21] USENIX	[GRS22] CRYPTO	[CFR23] USENIX	[vBP24] EUROCRYPT	[GQL+24] ASIACRYPT	[vBP25] ASIACRYPT	[SPK+25] IEEE Access	[PST+25] CCS	[DZL25] CCS
dist	Hamming	ℓ_∞	Hamming, ℓ_1 $(d = 1)$	$\ell_{\mathrm{p},\ \mathrm{p}\in[1,\infty)}$	Hamming, $\ell_{\mathrm{p},\ \mathrm{p}\in[1,\infty)}$	$\ell_{\mathrm{p},\ \mathrm{p}\in[1,\infty)}$	Cosine Similarity	$\ell_\infty, \ell_1,$ $\ell_2 (d = 2)$	$\ell_{\mathrm{p},\ \mathrm{p}\in[1,\infty)}$

*Private Set Intersection (PSI)

Fuzzy Matching for ℓ_∞

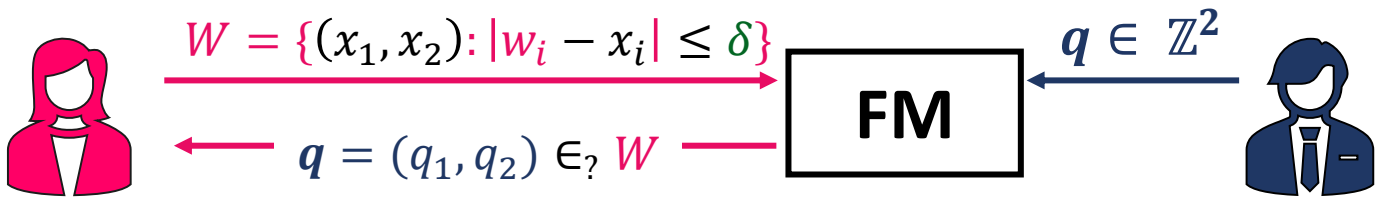


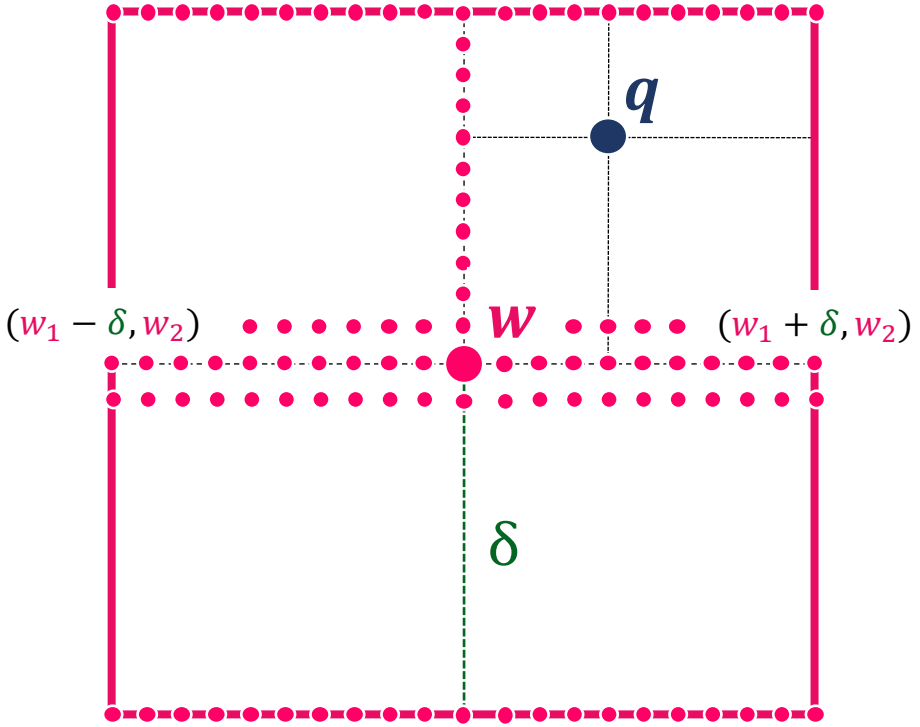


Fuzzy Matching for ℓ_∞



Idea #1

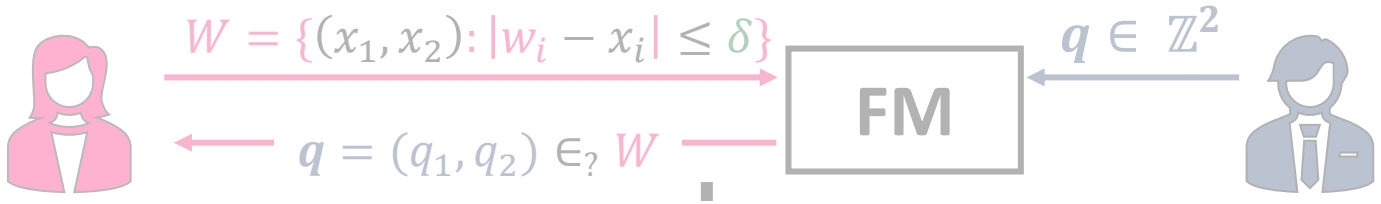




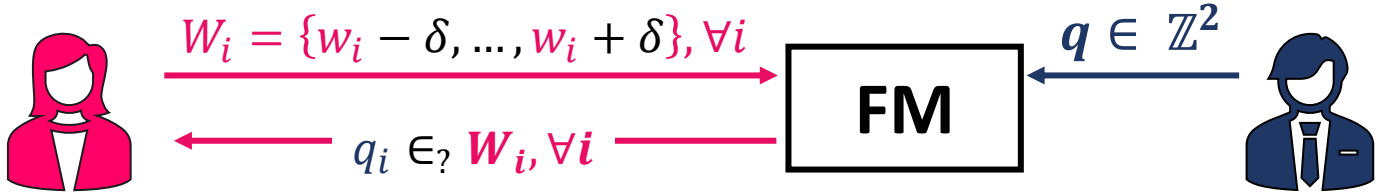
Fuzzy Matching for ℓ_∞

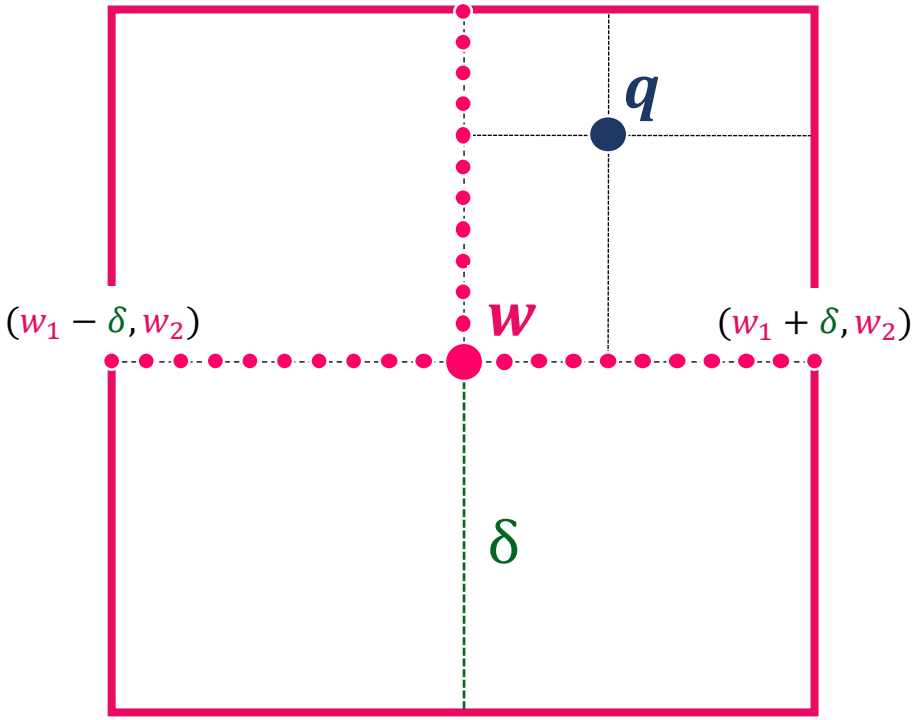


Idea #1

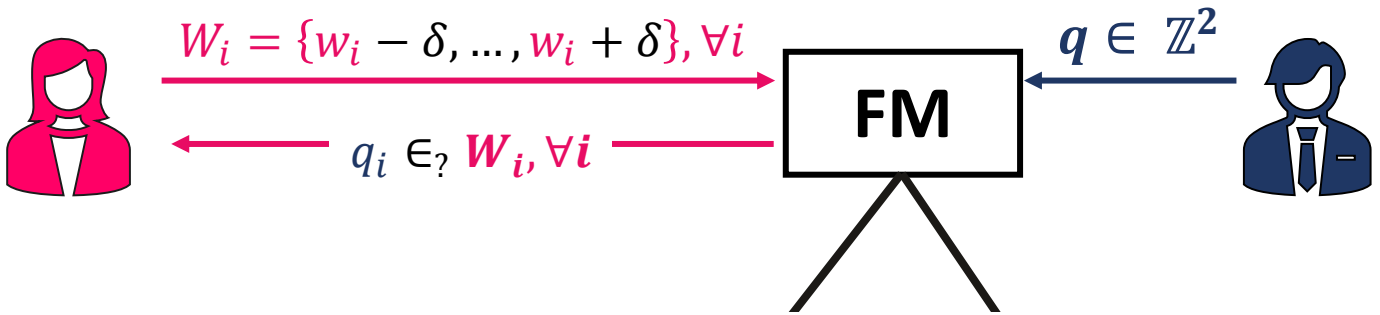


Idea #2





Fuzzy Matching for $\ell_{\mathrm{p},\,\mathrm{p}\in[1,\infty]}$ (from [vBP24])



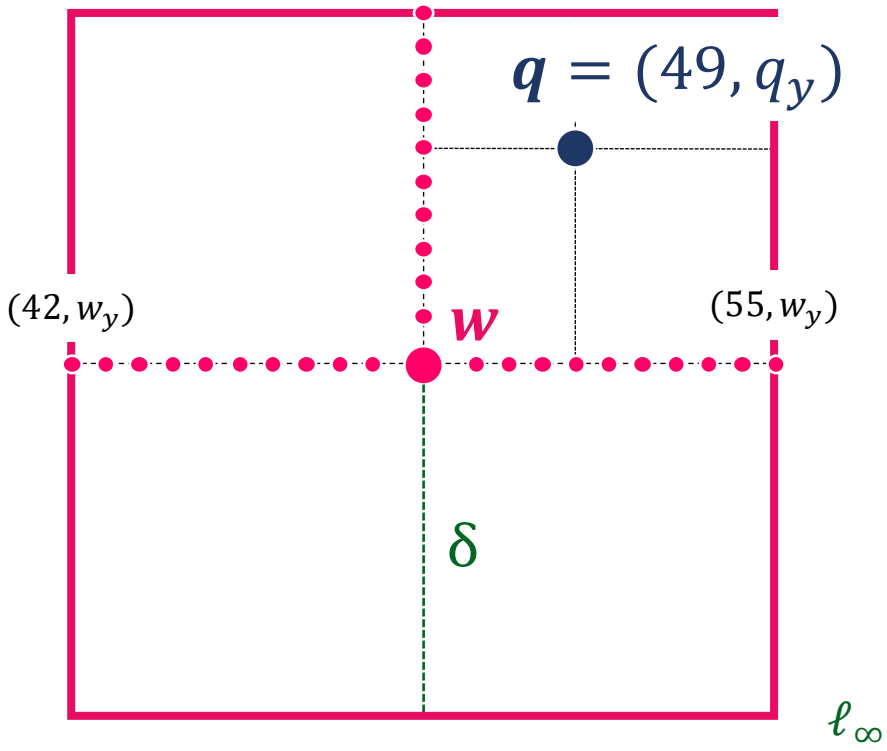
	ℓ_∞	$\ell_{\mathrm{p},\mathrm{p}\in[1,\infty)}$
Total Comm.:	$\sim O(\delta d)$	$\sim O(\delta d + \delta^{\mathrm{p}})$
Alice's Comp.:	$\sim O(\delta d)$	$\sim O(\delta d)$
Bob's Comp.:	$\sim O(d)$	$\sim O(d + \delta^{\mathrm{p}})$

Geometric Fuzzy Matching

(Geometric) Fuzzy Matching for ℓ_∞ (GFM_∞)

Goal: Get more efficient Fuzzy Matching for the ℓ_∞ distance.

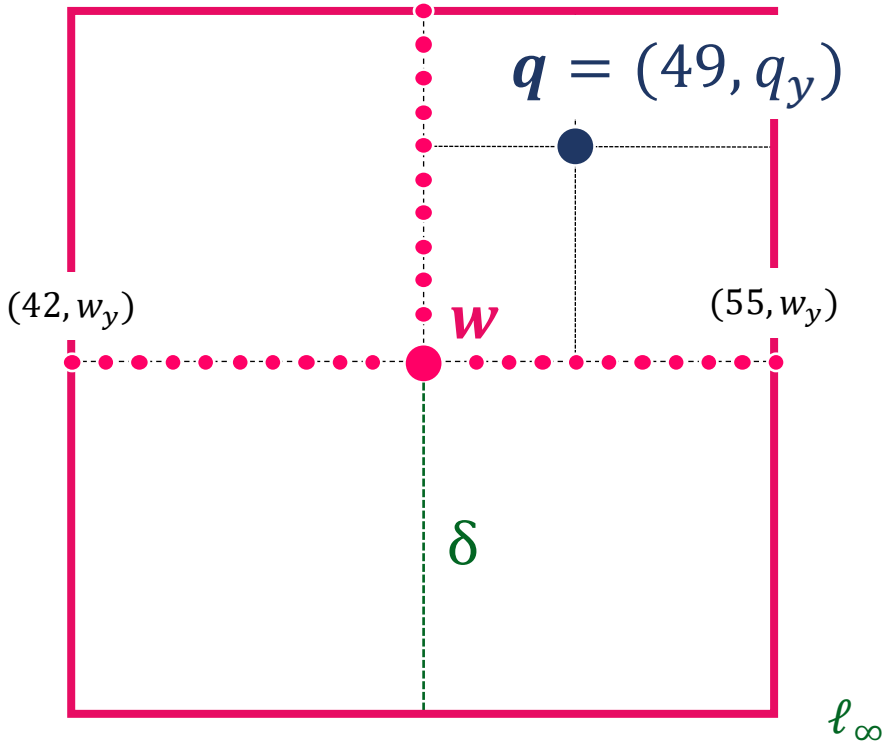
How?



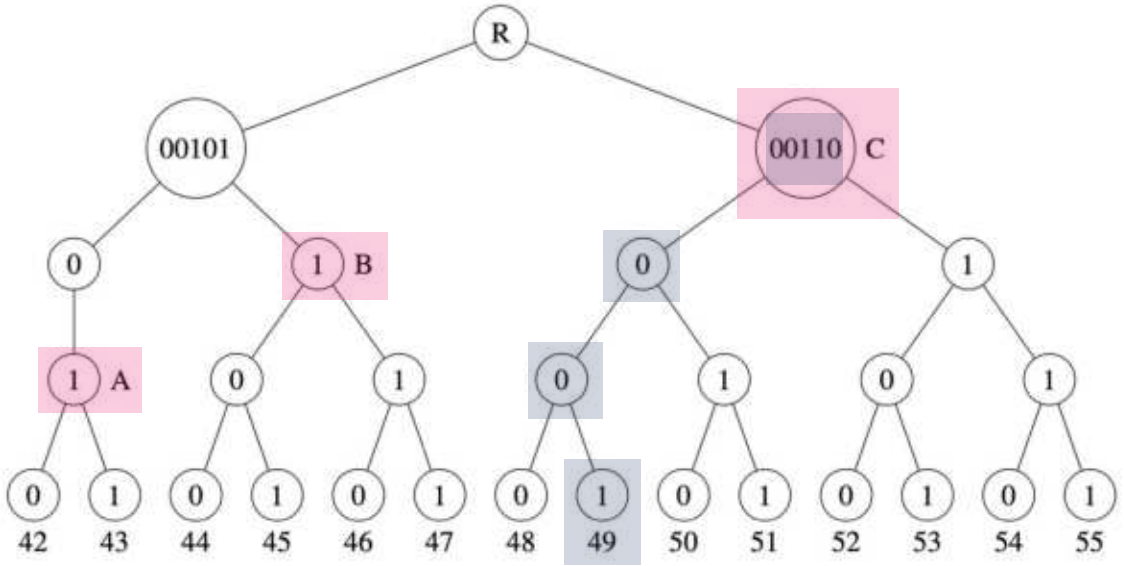
(Geometric) Fuzzy Matching for ℓ_∞ (GFM_∞)

Goal: Get more efficient Fuzzy Matching for the ℓ_∞ distance.

How? FM from [vBP24] + **Distance-Aware Set Augmentation** technique [CFR23]



Represent interval $[w - \delta, w + \delta]$ with $O(\log \delta)$ strings



$$[42{,}55] \leftrightarrow \{0010101*, 001011**, 00110***\}$$

$$49 \leftrightarrow \{00110001, 0011000*, 001100**, 00110***, 0011****\}$$

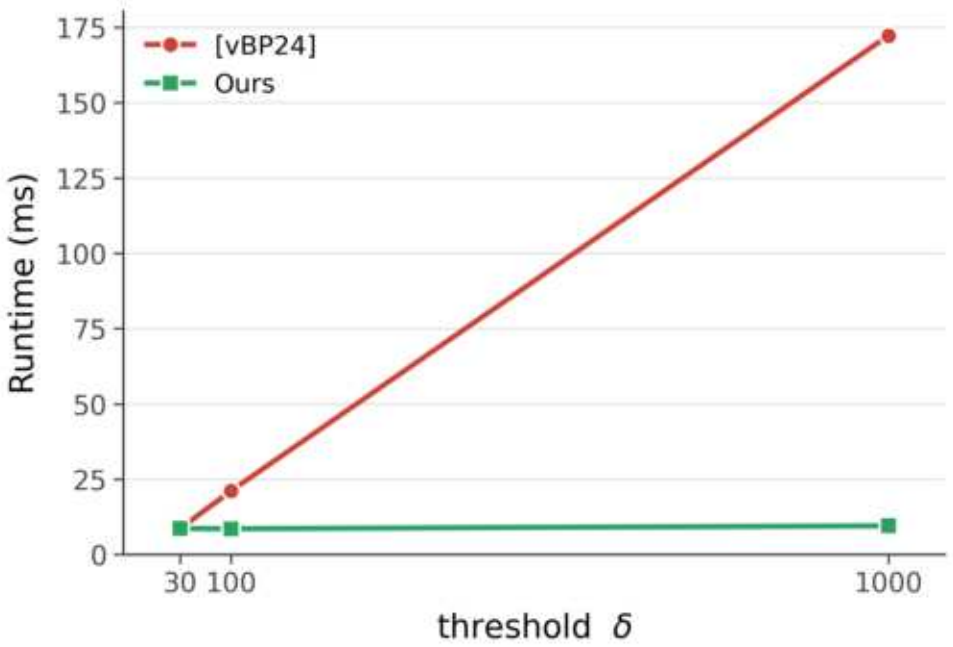
$\mathbf{GFM}_{\infty}$ – Complexities & Comparison

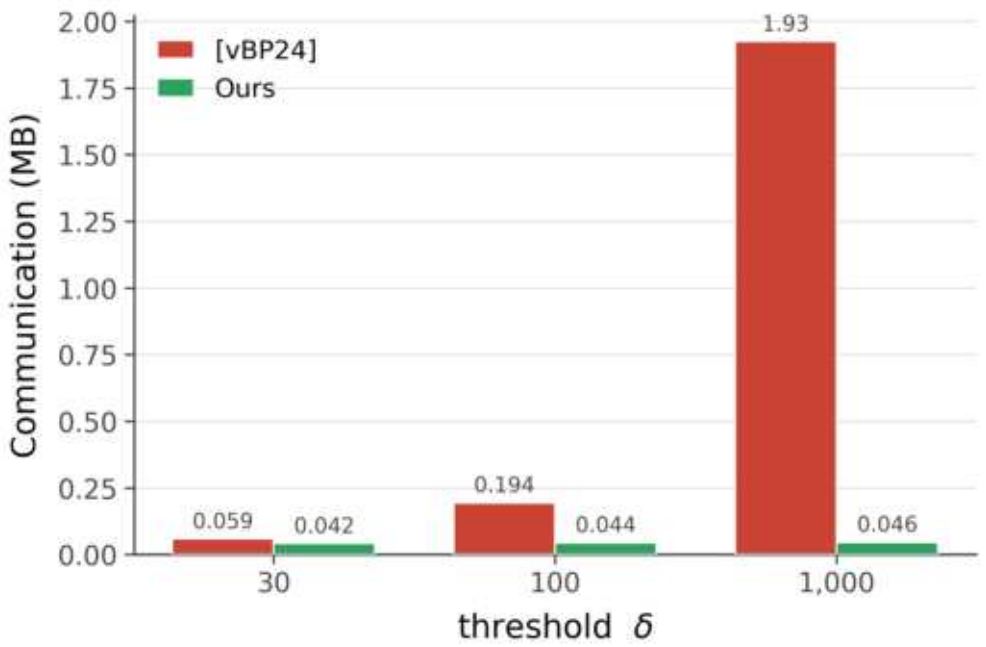
ℓ_∞	Total Communication	Receiver's Computations	Sender's Computations
[vBP24]	$O(\delta d)$	$O(\delta d)$	$O(d)$
Ours	$O(2\delta \log \delta + \log^{\mathrm{d}} \delta)$	$O(2\delta \log \delta + \log^{\mathrm{d}} \delta)$	$O(\mathrm{d} \log^{\mathrm{d}} \delta)$

$\mathbf{GFM}_{\infty}$ – Complexities & Comparison

ℓ_∞	Total Communication	Receiver's Computations	Sender's Computations
[vBP24]	$O(\delta d)$	$O(\delta d)$	$O(d)$
Ours	$O(2\delta \log \delta + \log^{\mathrm{d}}\delta)$	$O(2\delta \log \delta + \log^{\mathrm{d}}\delta)$	$O(\mathrm{d} \log^{\mathrm{d}}\delta)$

$d = 2$

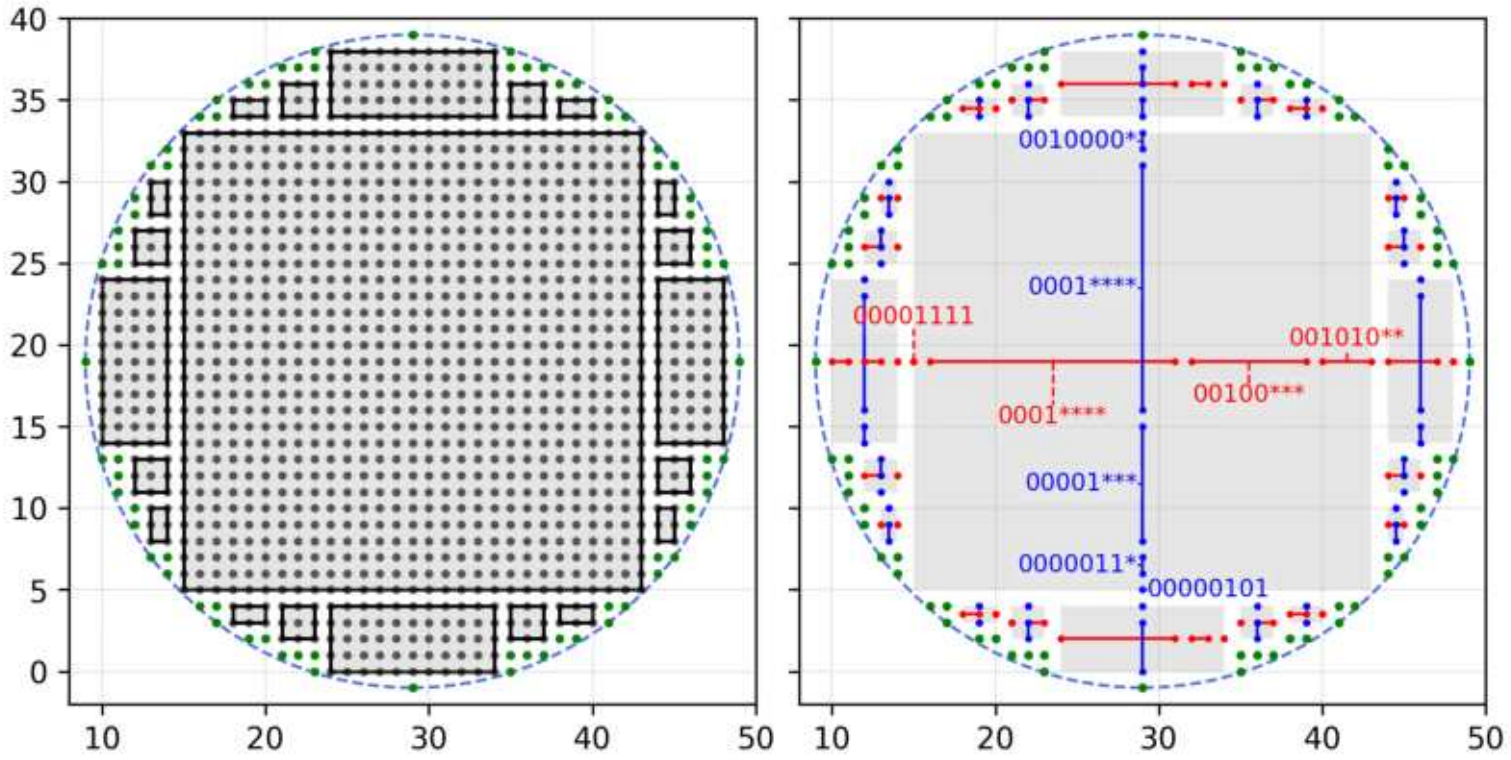




Geometric Fuzzy Matching

Goal: Think geometrically! Get expressive Fuzzy Matching for any region.

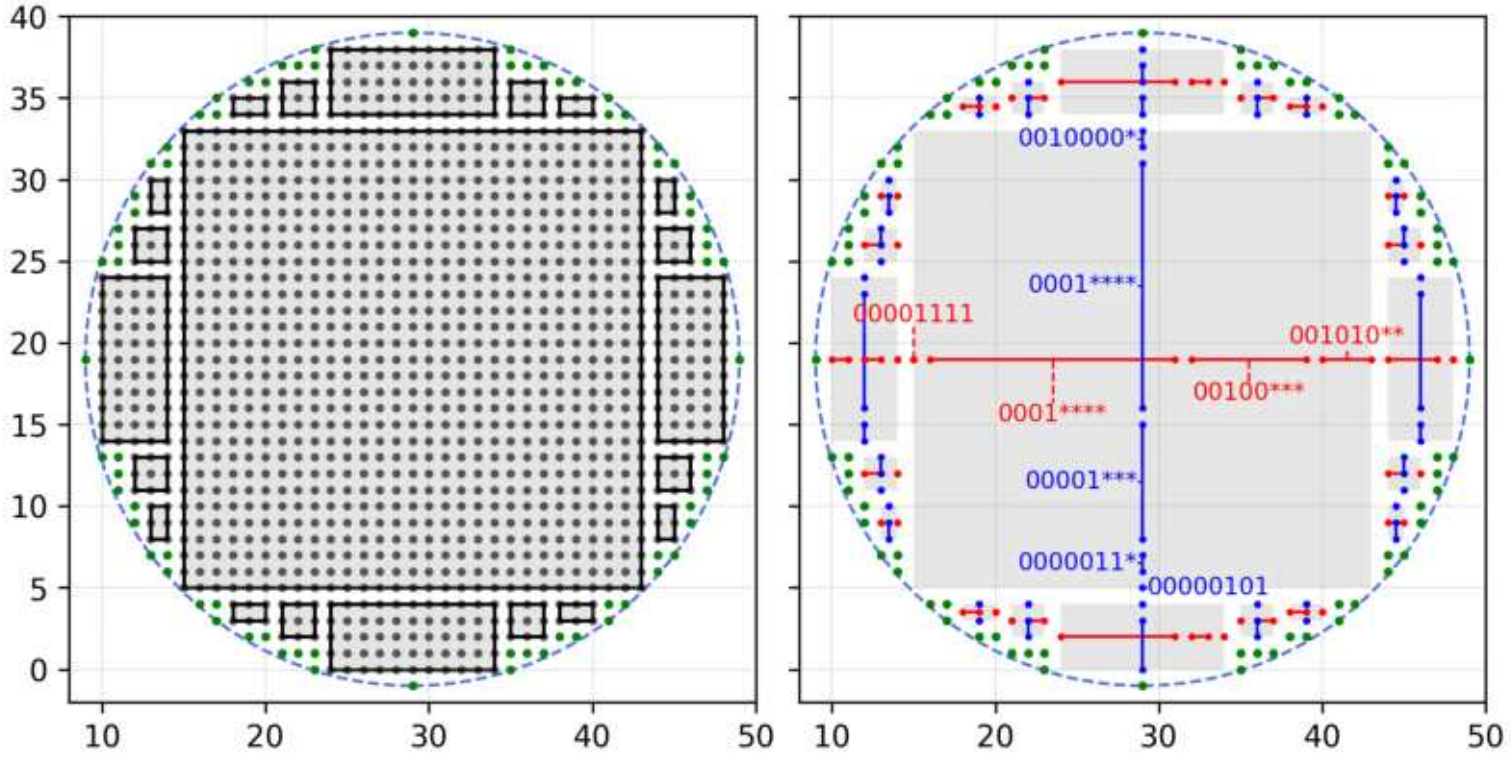
How? GFM_∞ **+** Partitioning Algorithms

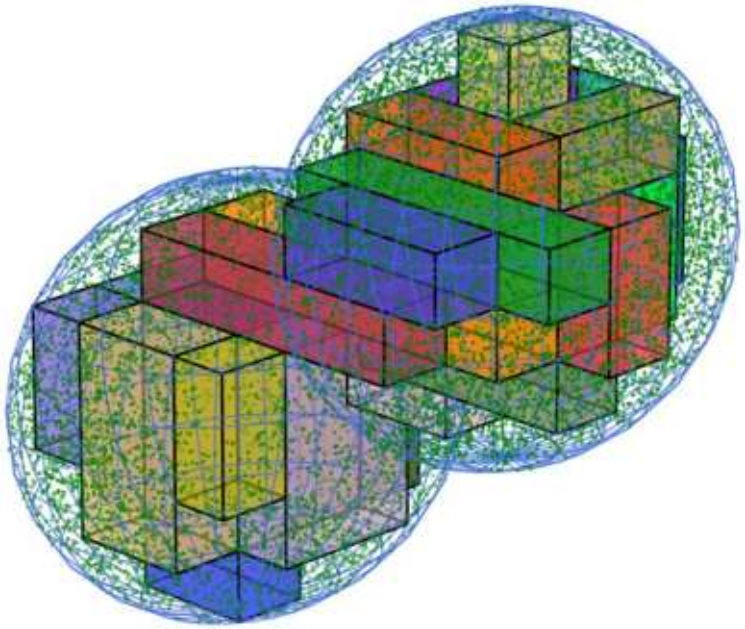


Geometric Fuzzy Matching

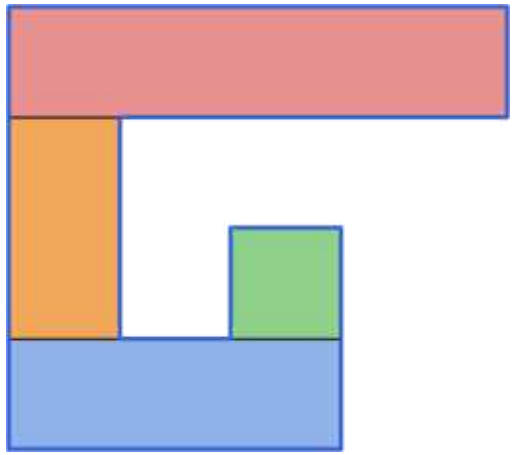
Goal: Think geometrically! Get expressive Fuzzy Matching for any region.

How? GFM_∞ **+** Partitioning Algorithms



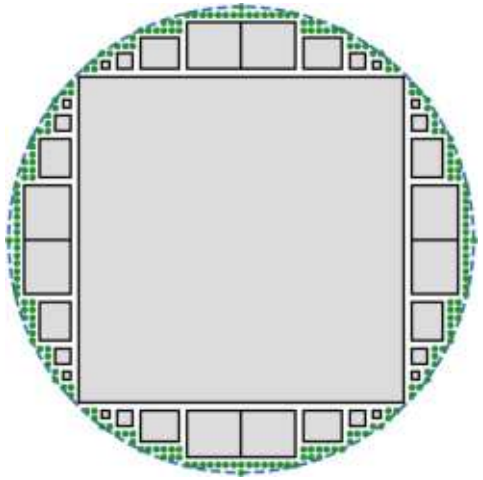


Partitioning: It Depends on the Goal



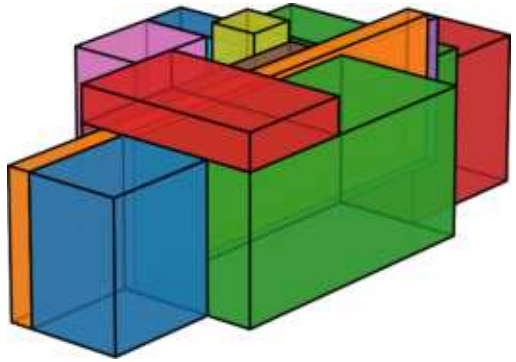
Rectilinear polygons

Exact rectangles
min. cut length [LPRS82]
min. #rectangles [WS94]



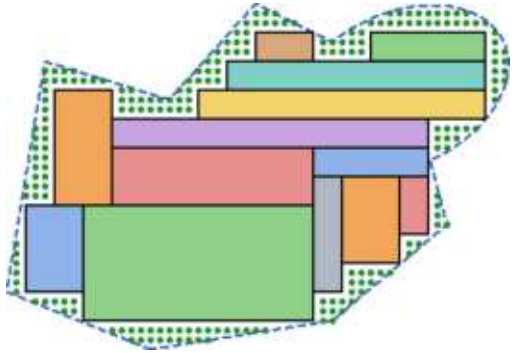
Discs

Square packing [Gla14]
+ residual points



Unions of boxes

Disjoint boxes, exact
cut and fuse [HMKC22]



Any region

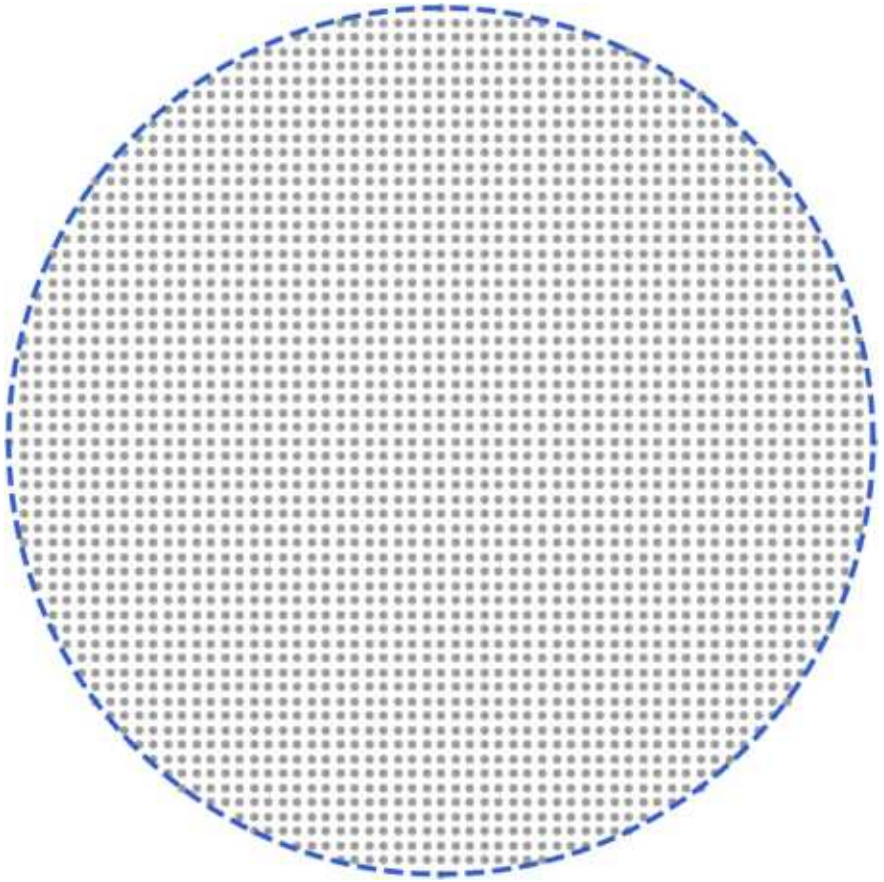
CutAndFuse **(ours)**
any dimension
+ residual points

Our goal: cover every integer point, balancing sender and receiver work

[LPRS82] Lingas, Pinter, Rivest, Shamir, Allerton '82 [WS94] Wu, Sahni, VLSI Design '94
[Gla14] Glassner, Square Packing in a Circle '14 [HMKC22] Holub, Martel, Kuhn, Couch, xrf '22

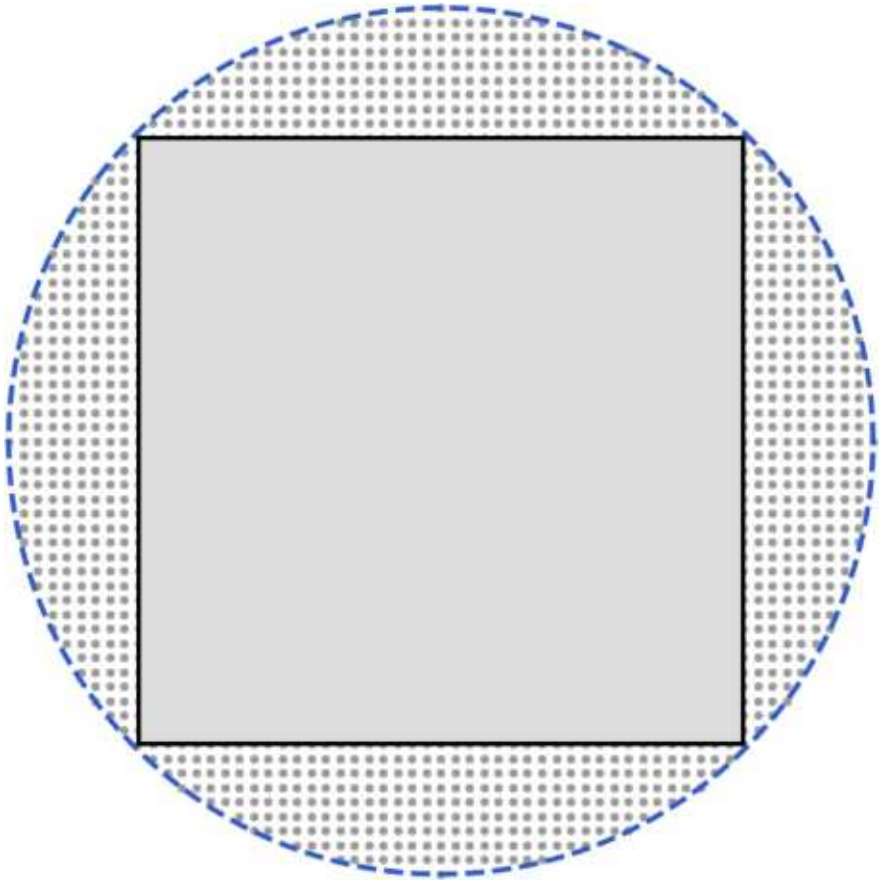
Partitioning Discs

Goal: cover all integer points of a disc of radius δ



Partitioning Discs

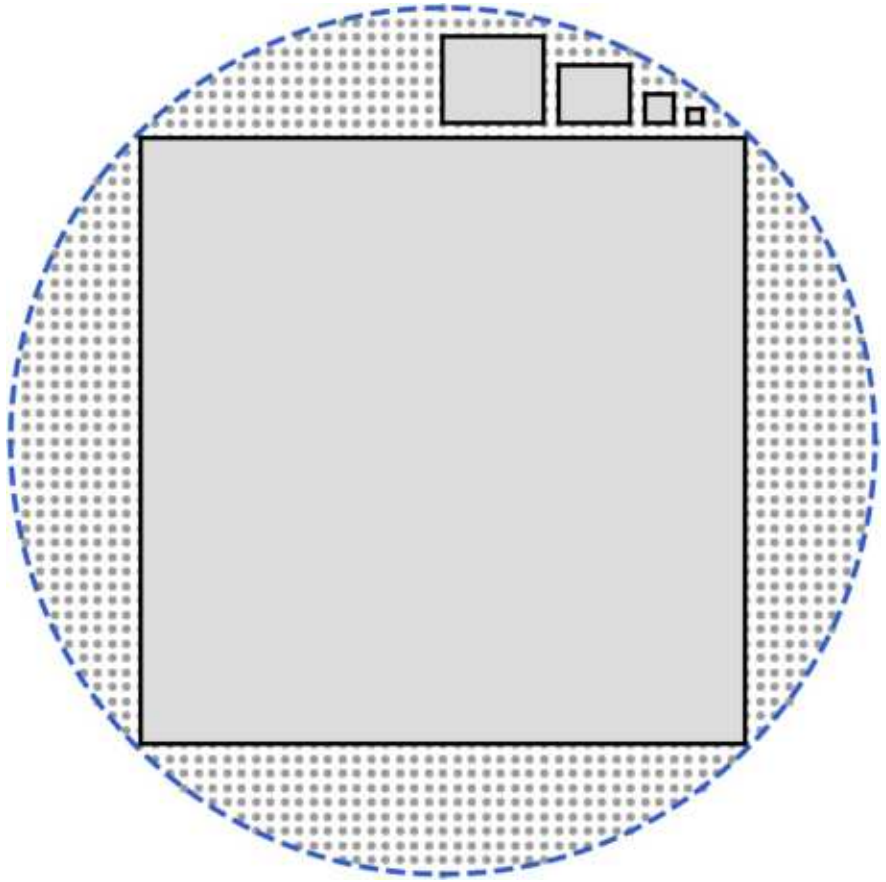
Goal: cover all integer points of a disc of radius δ



1 Largest square in the middle

Partitioning Discs

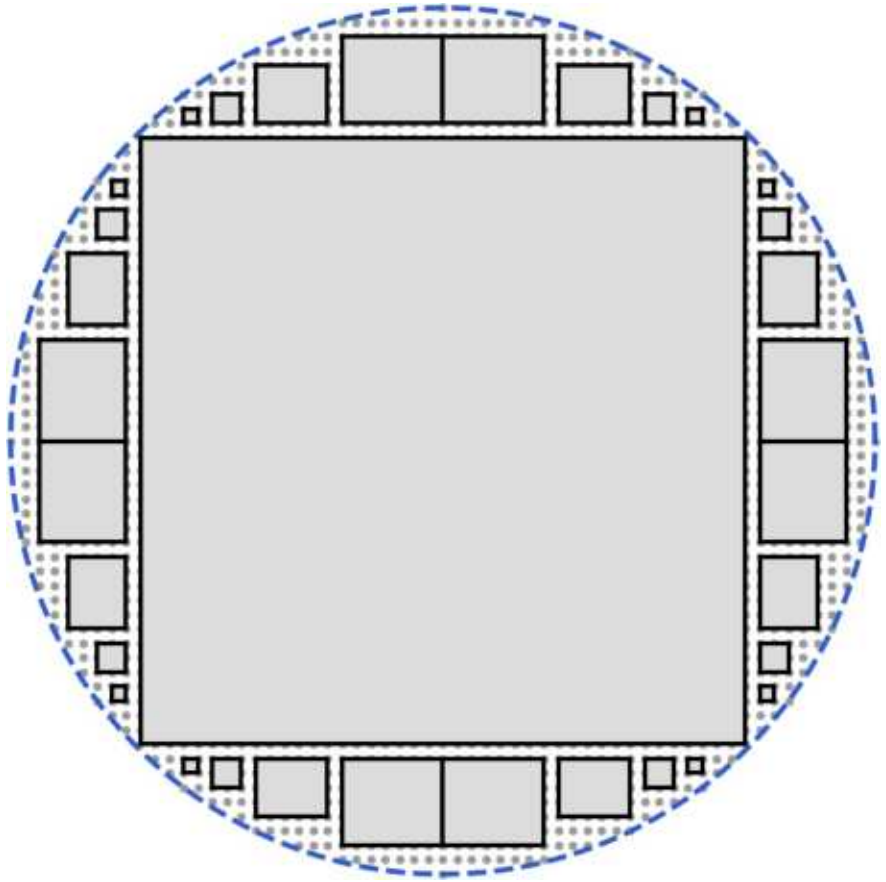
Goal: cover all integer points of a disc of radius δ



1. Largest square in the middle
2. Grow squares into one octant

Partitioning Discs

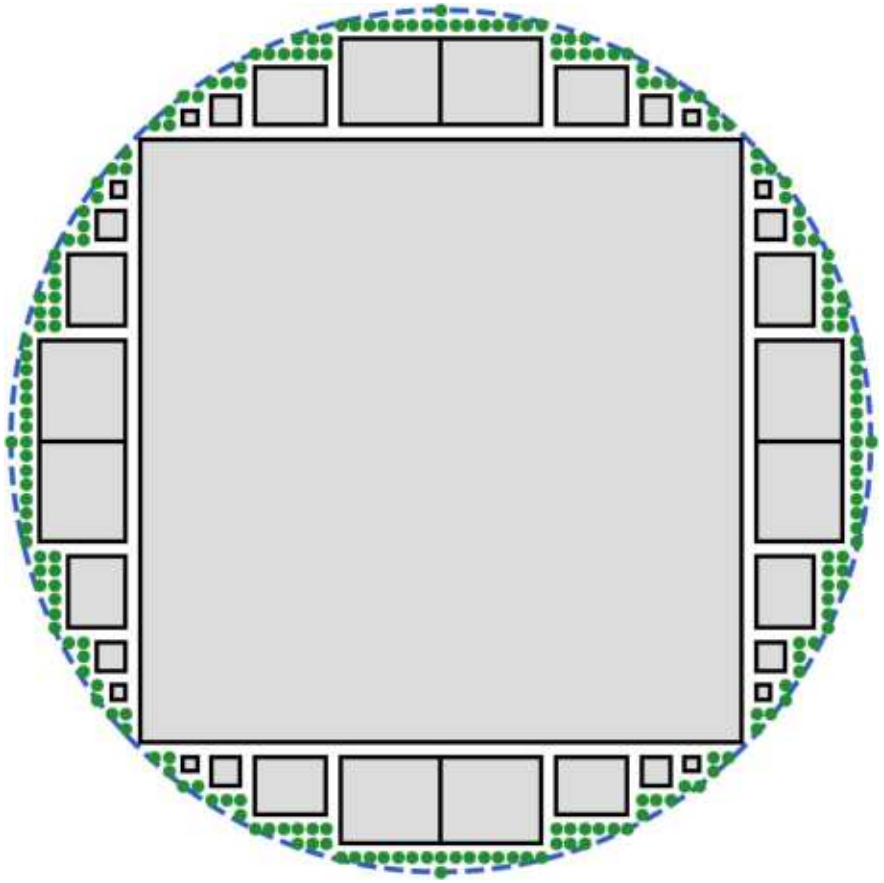
Goal: cover all integer points of a disc of radius δ



1. Largest square in the middle
2. Grow squares into one octant
3. Mirror into all 8 octants

Partitioning Discs

Goal: cover all integer points of a disc of radius δ

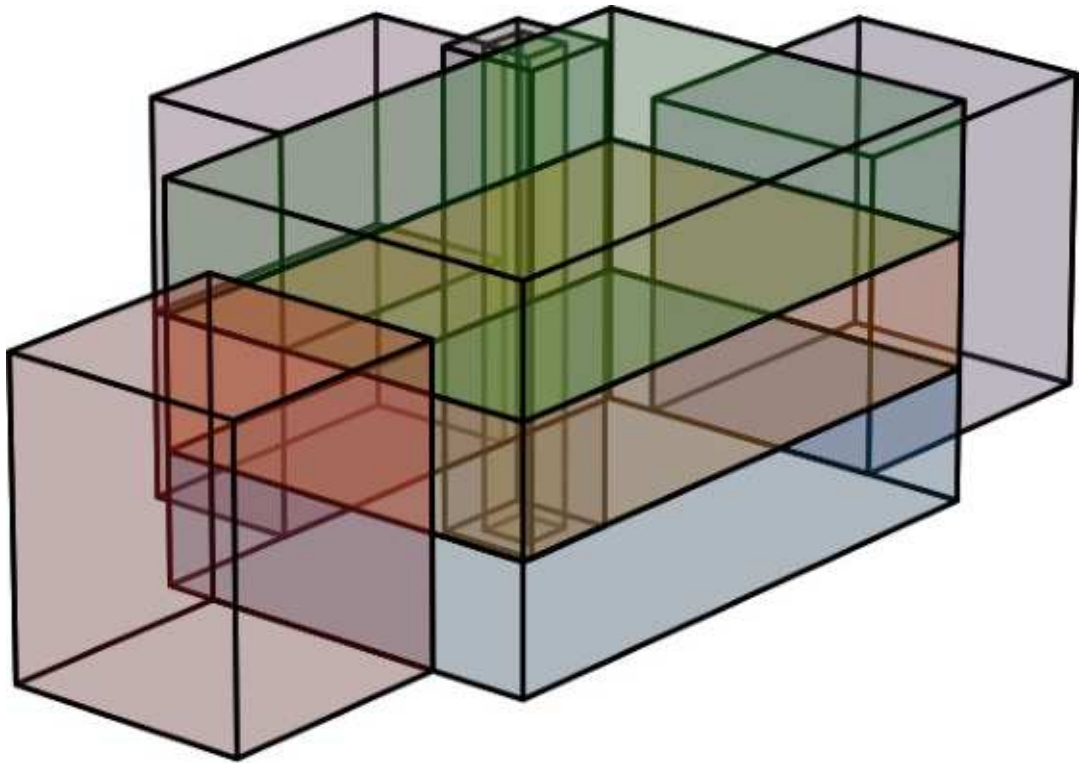


1. Largest square in the middle
2. Grow squares into one octant
3. Mirror into all 8 octants
4. Leftover points: residuals

δ = 30: 33 squares + 208 residual points

Partitioning Unions of Hyperrectangles

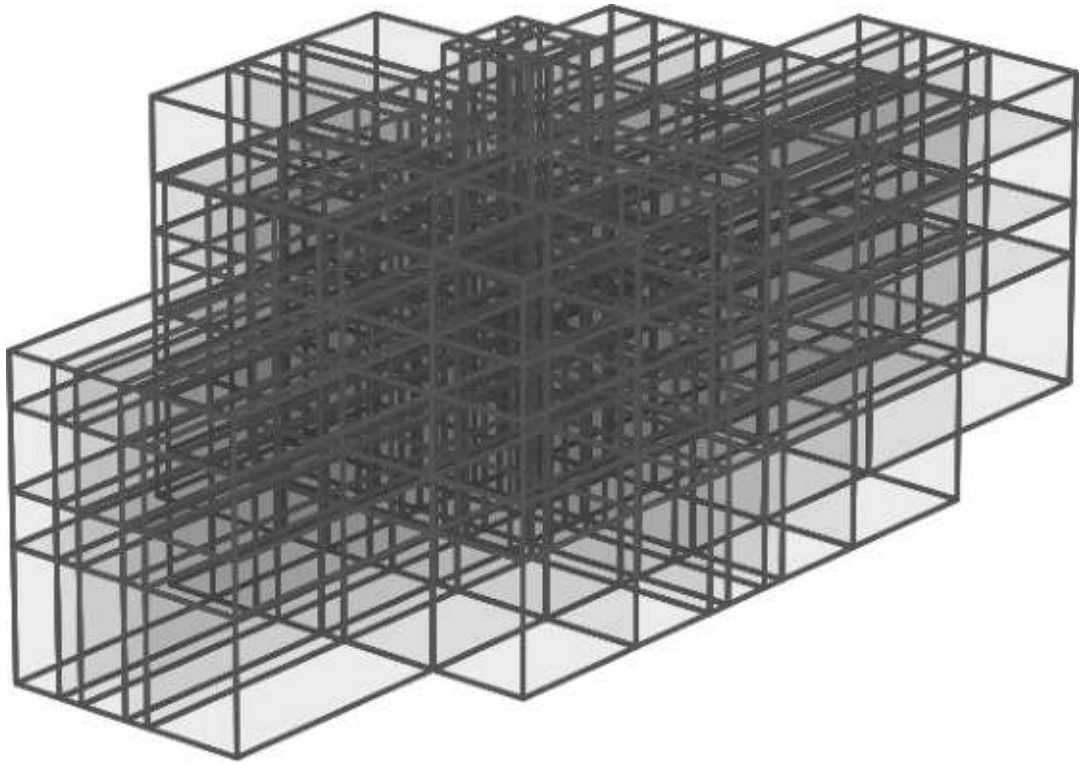
Goal: 3D regions such as a mall or a building (Demo 3)



1 Input: overlapping boxes

Partitioning Unions of Hyperrectangles

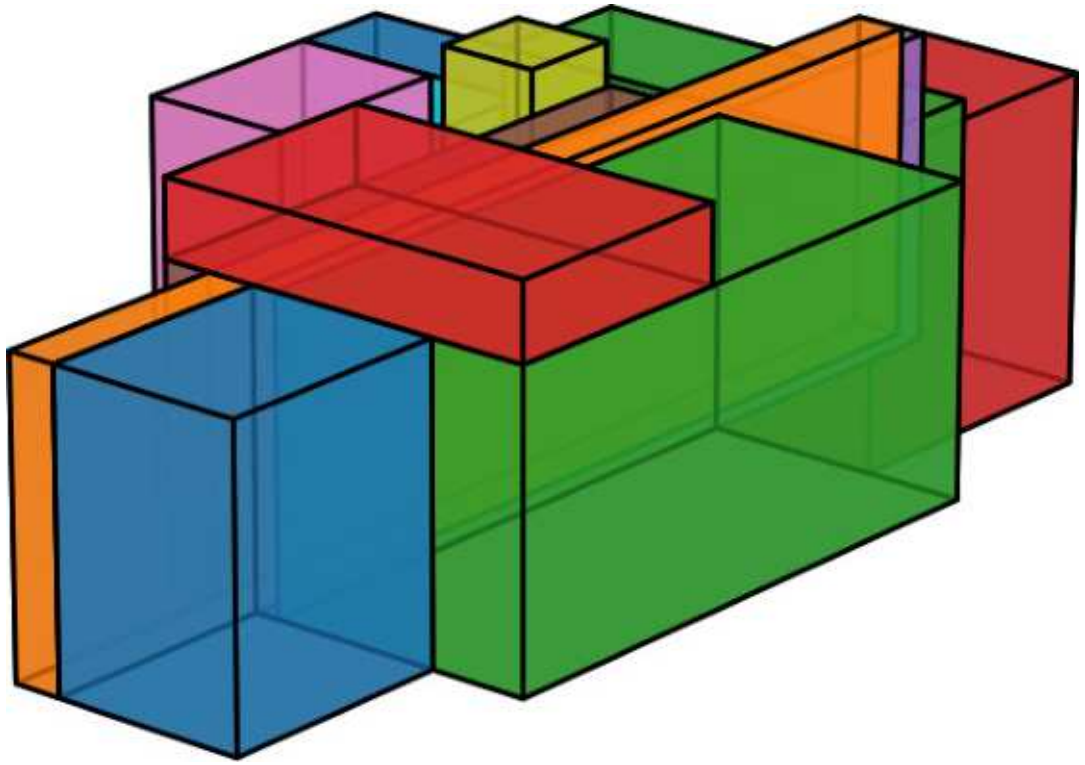
Goal: 3D regions such as a mall or a building (Demo 3)



1. Input: overlapping boxes
2. Grid from the boxes' edges

Partitioning Unions of Hyperrectangles

Goal: 3D regions such as a mall or a building (Demo 3)



1. Input: overlapping boxes
2. Grid from the boxes' edges
3. Fuse into disjoint boxes

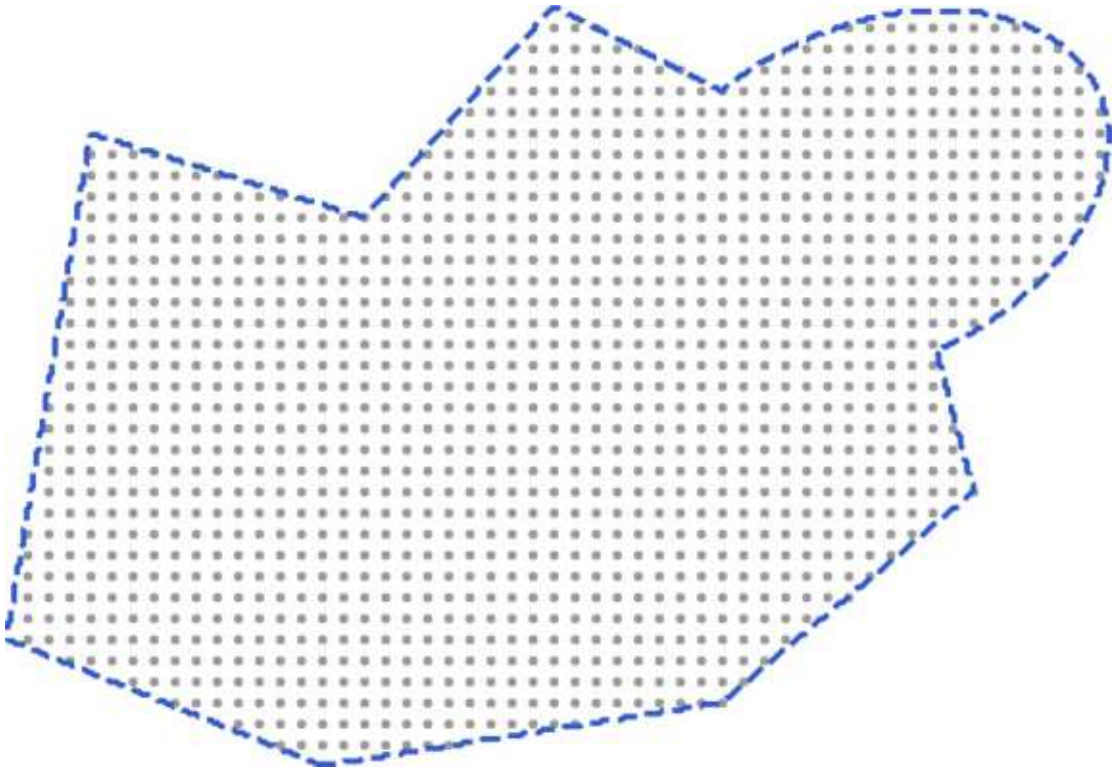
9 boxes → 319 cells → 14 distinct boxes

Exact: no residual points

Cut-and-fuse idea from [HMKC22] Holub, Martel, Kuhn, Couch, xrf '22

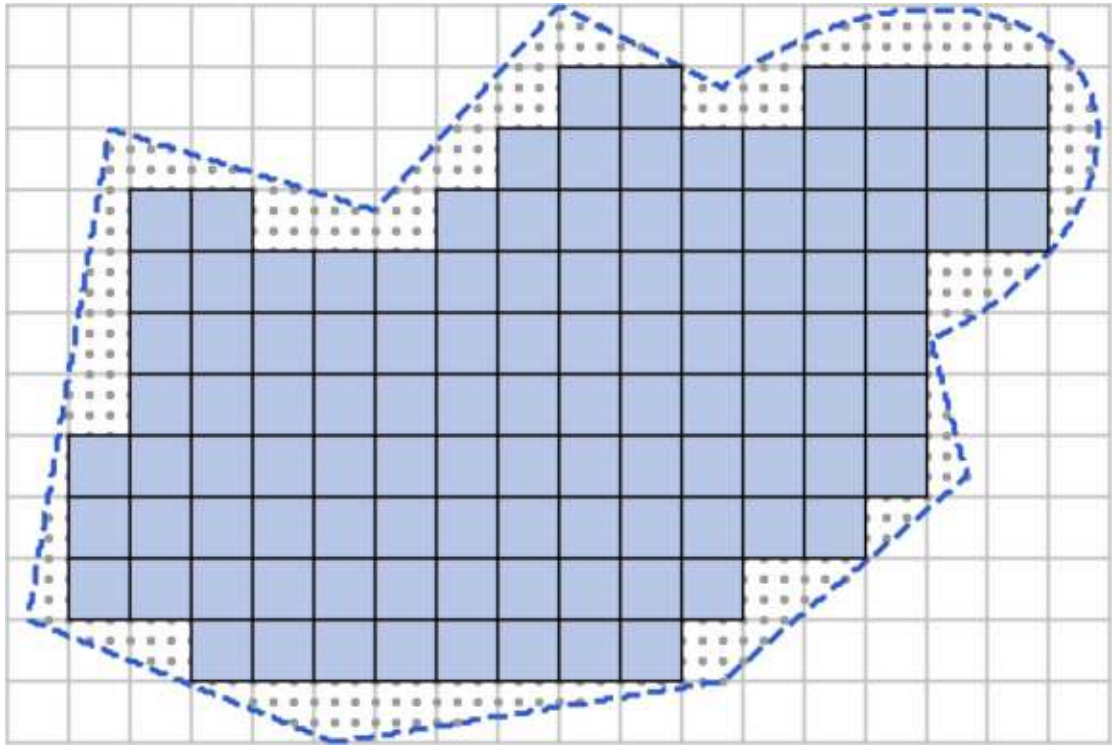
Generic Partitioner: CutAndFuse

Goal: any *closed and compact* surface S, in any dimension d



Generic Partitioner: CutAndFuse

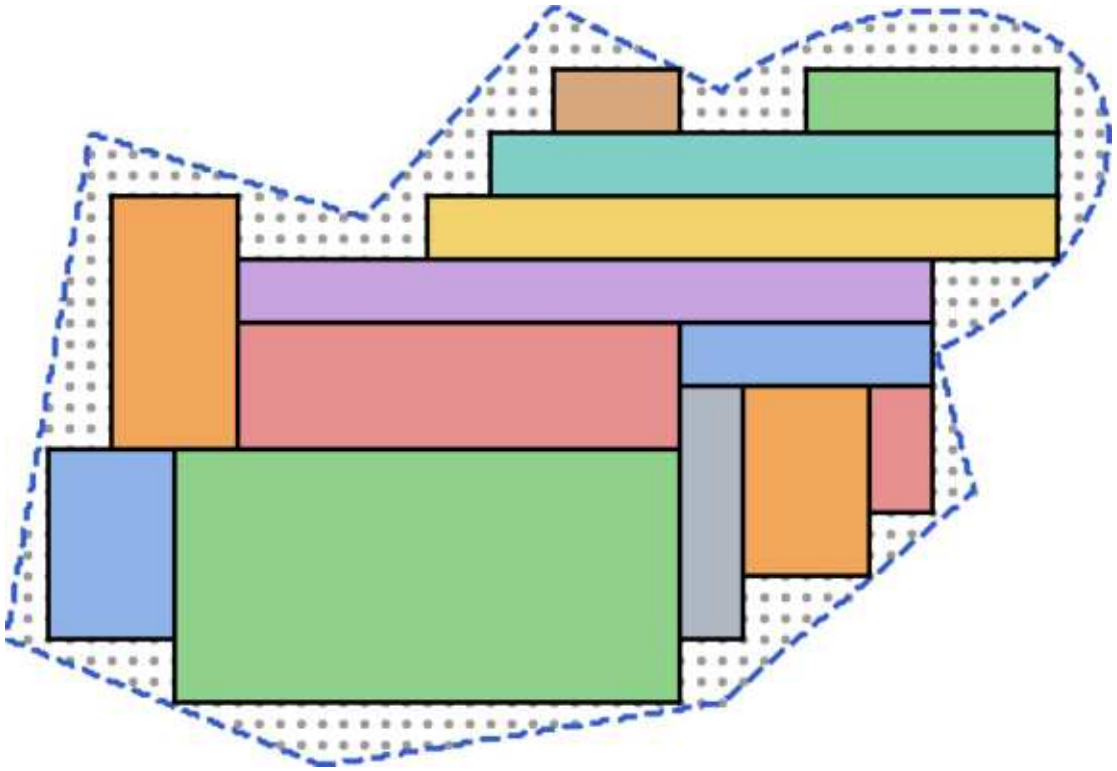
Goal: any *closed and compact* surface S, in any dimension d



Cut: r × r grid, keep inner cells

Generic Partitioner: CutAndFuse

Goal: any *closed and compact* surface S, in any dimension d

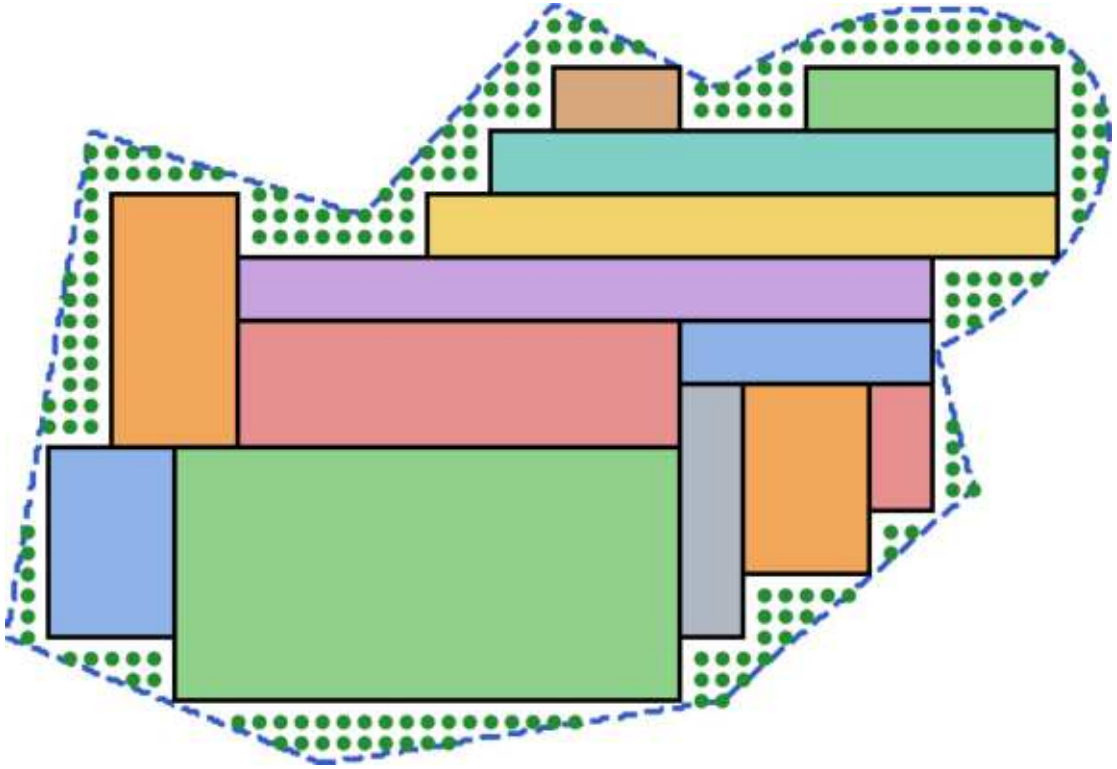


Cut: r × r grid, keep inner cells

Fuse: merge neighbouring cells

Generic Partitioner: CutAndFuse

Goal: any *closed and compact* surface S, in any dimension d



Cut: r × r grid, keep inner cells

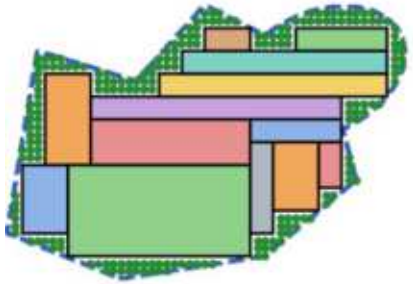
Fuse: merge neighbouring cells

Residuals: leftover integer points

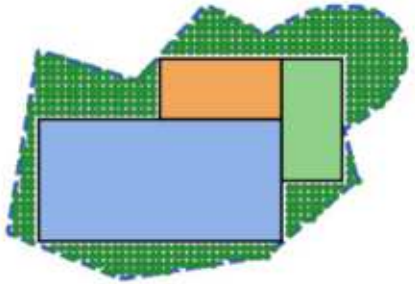
r = 3: 112 cells → 13 rectangles
+ 202 residual points

Generic Partitioner: Choosing r*

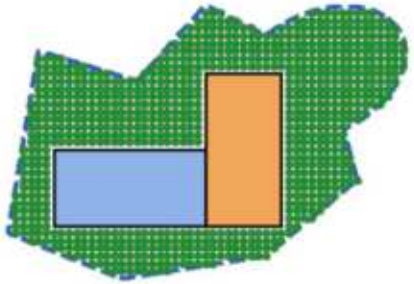
r = 3
13 rectangles
202 residual points

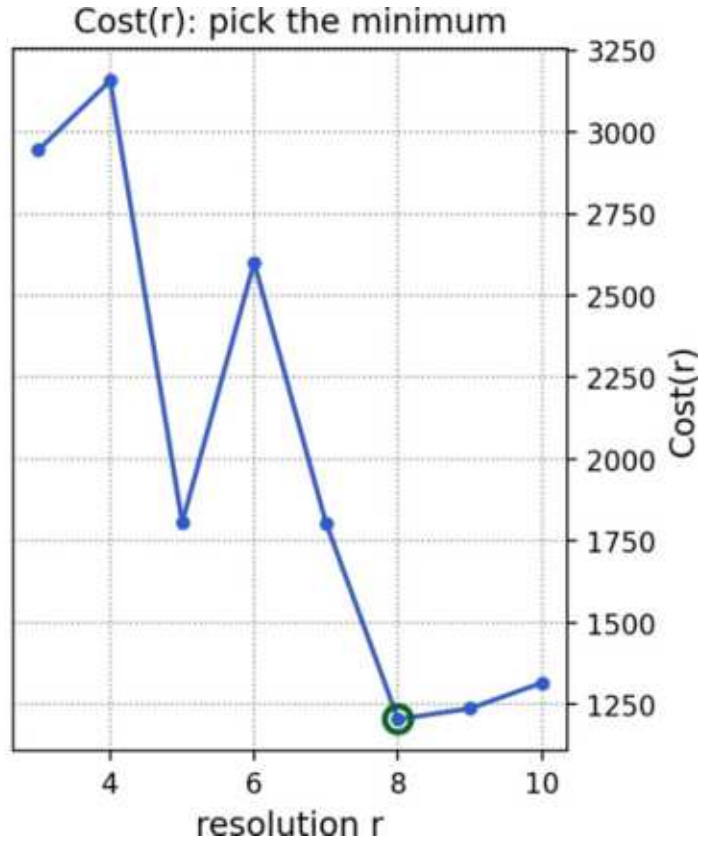


r = 8 (r*)
3 rectangles
462 residual points



r = 10
2 rectangles
844 residual points



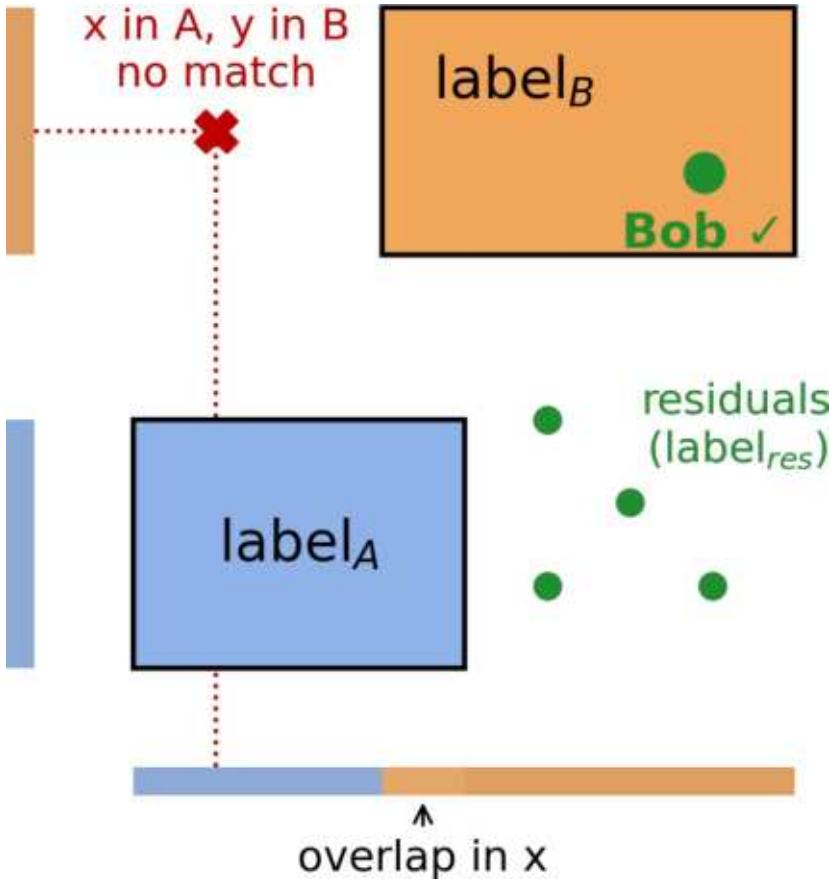


Small r: many rectangles, more work for the sender

Large r: many residual points, more work for the receiver

Cost(r) is not monotone: try every r, keep the minimum

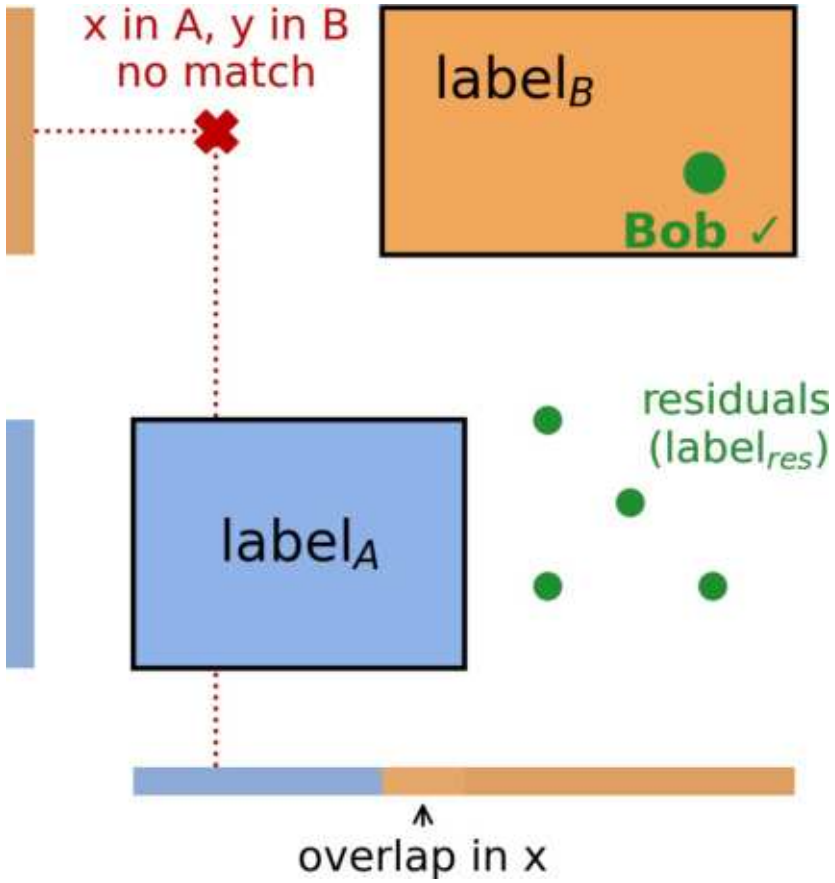
GFM: From Boxes to One Bit



One GFM_∞ instance per hyperrectangle k

→ OKVS keys $H(label_k \| i \| prefix)$

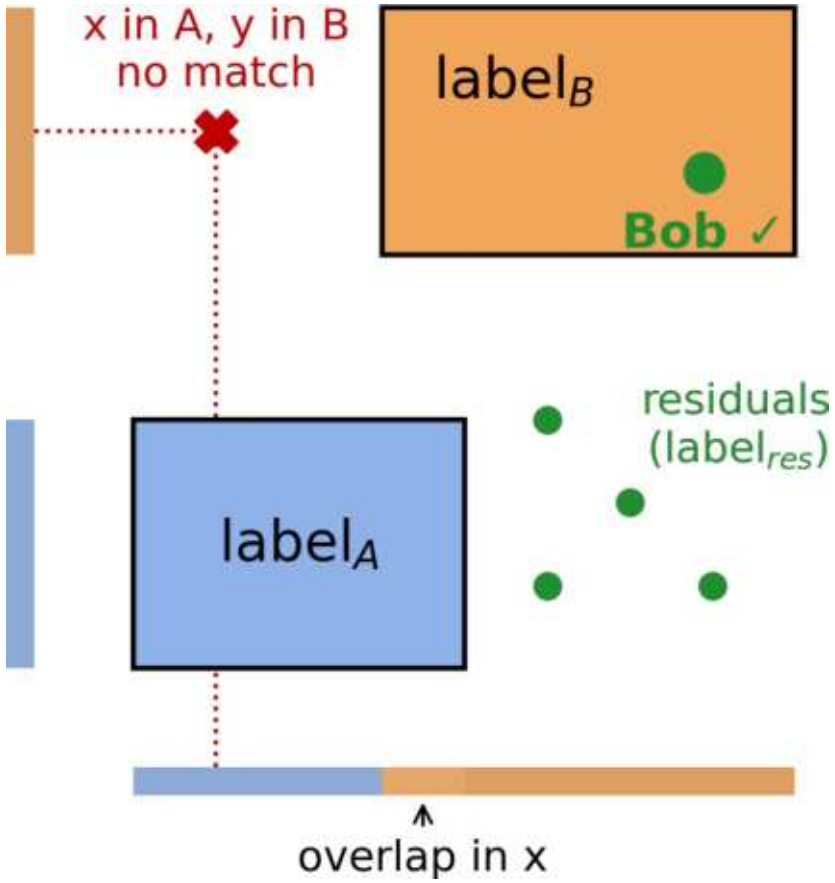
GFM: From Boxes to One Bit



One GFM_∞ instance per hyperrectangle k
→ OKVS keys $H(label_k \| i \| prefix)$

Residuals taken care of separately

GFM: From Boxes to One Bit



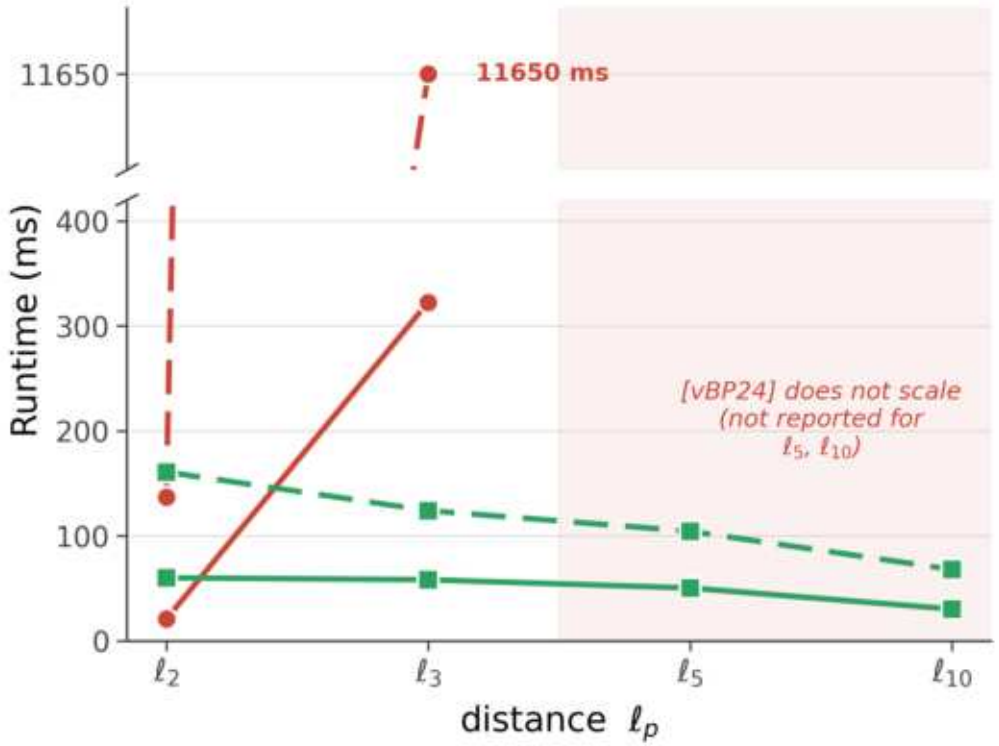
One GFM_∞ instance per hyperrectangle k
→ OKVS keys $H(label_k \| i \| prefix)$

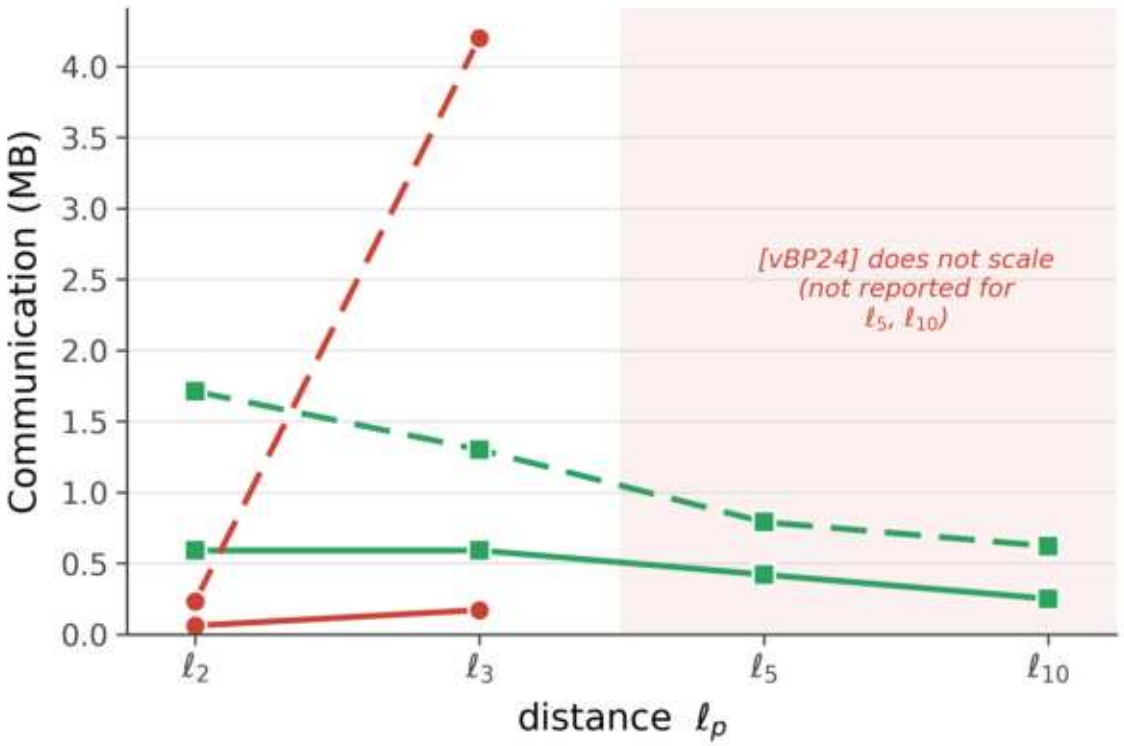
Residuals taken care of separately

Additive Homomorphic Encryption
merges everything together

GFM – Complexities & Comparison

$d = 2$





- Benchmarks for ℓ_p are representative of our framework's behavior across different geometric regions.

[vBP24], $\delta = 30$ (solid)
[vBP24], $\delta = 100$ (dashed)
Ours, $\delta = 30$ (solid)
Ours, $\delta = 100$ (dashed)

Nearby Friends’ Discovery

Geometric Fuzzy Matching

Nearby Friends' Finder from GFM

DEMO 01

L_2 Distance Proximity

Alice sets her location and a matching radius δ. She sends Msg_1. Bob places himself and sees δ around him. Protocol reveals only inside/outside.

`Our Partitioner algorithm`

DEMO 02

2D Polygon Region

Alice draws any region on a map. Bob's point is tested privately. Draw any shape, drag Bob live.

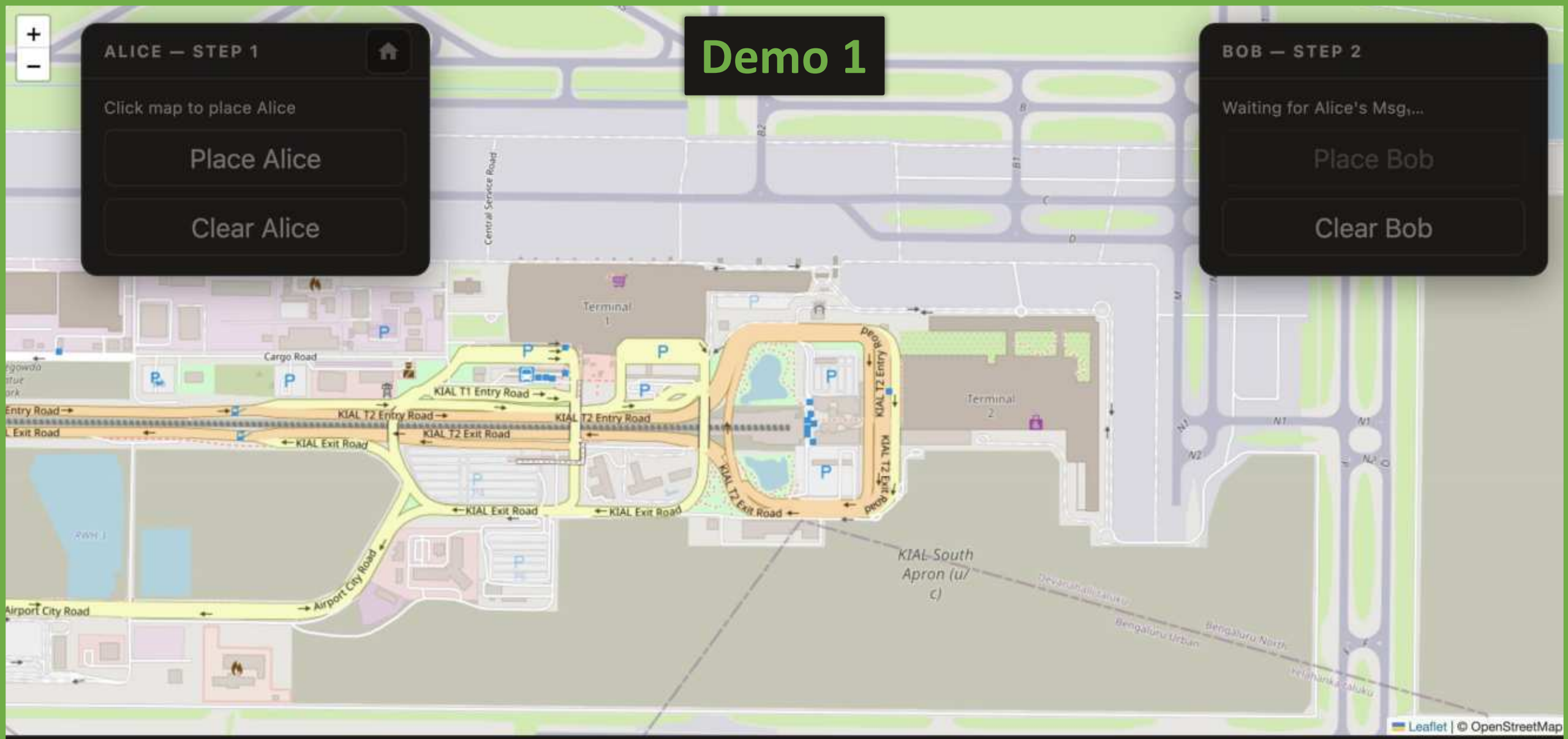
`Our Partitioner algorithm`

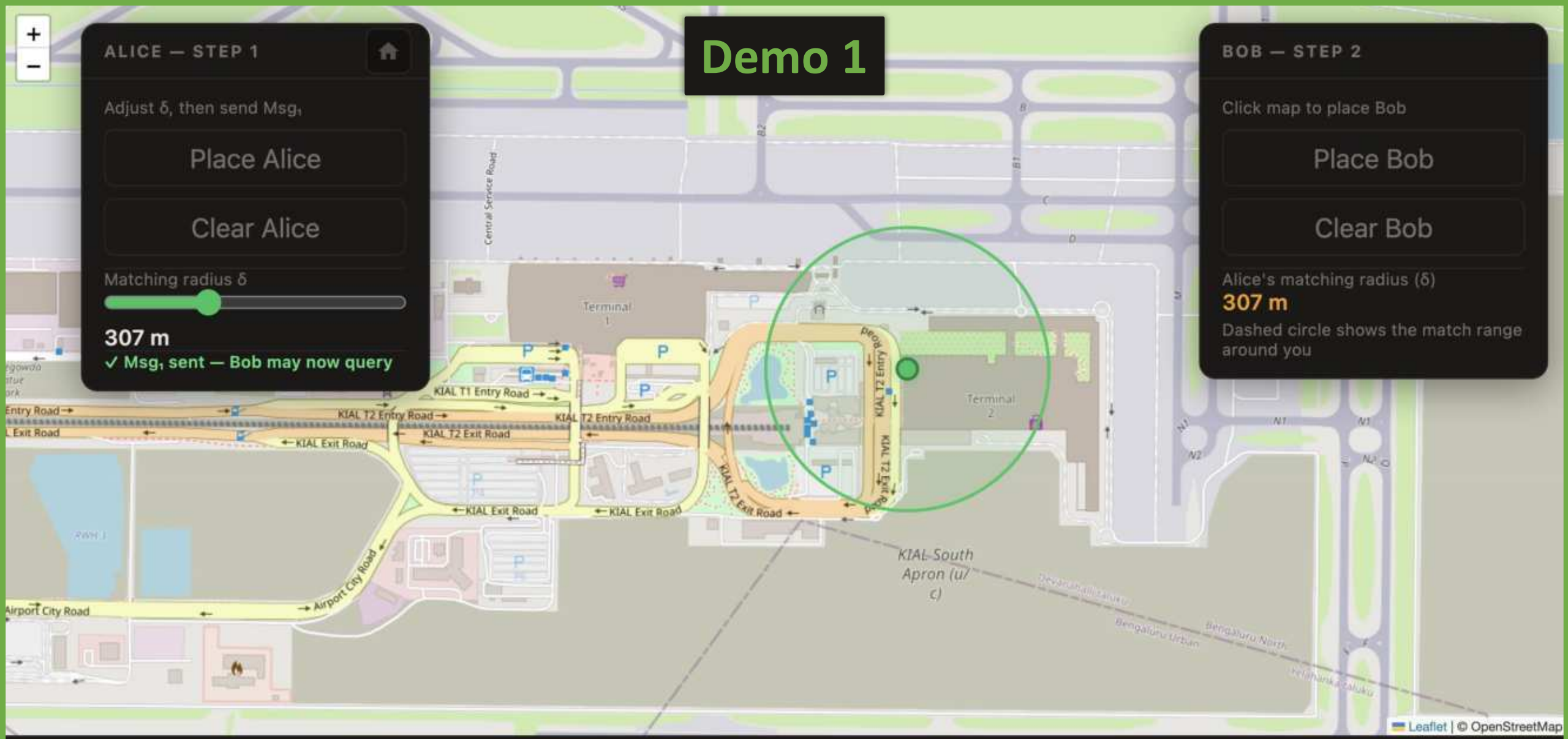
DEMO 03

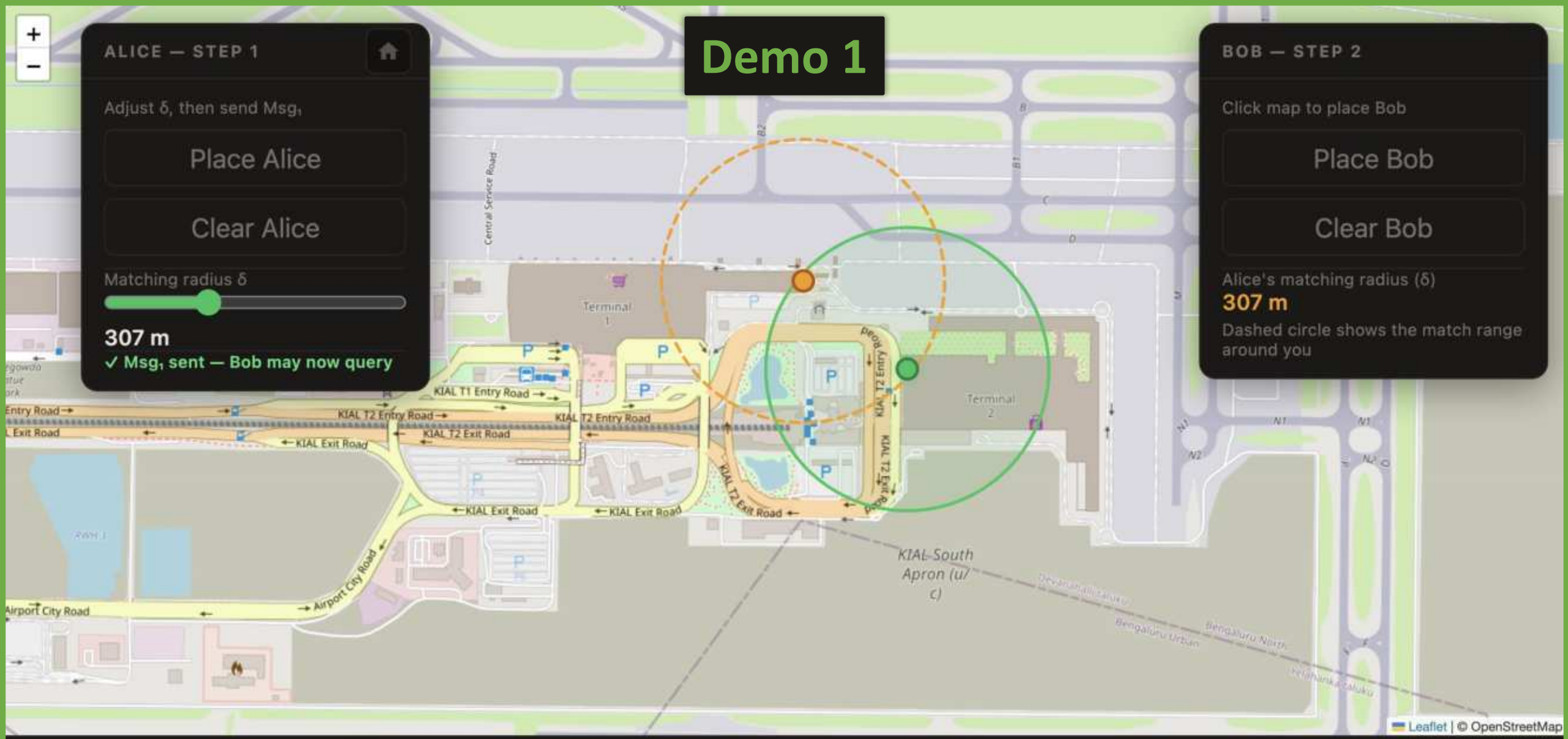
3D Building (Indoor)

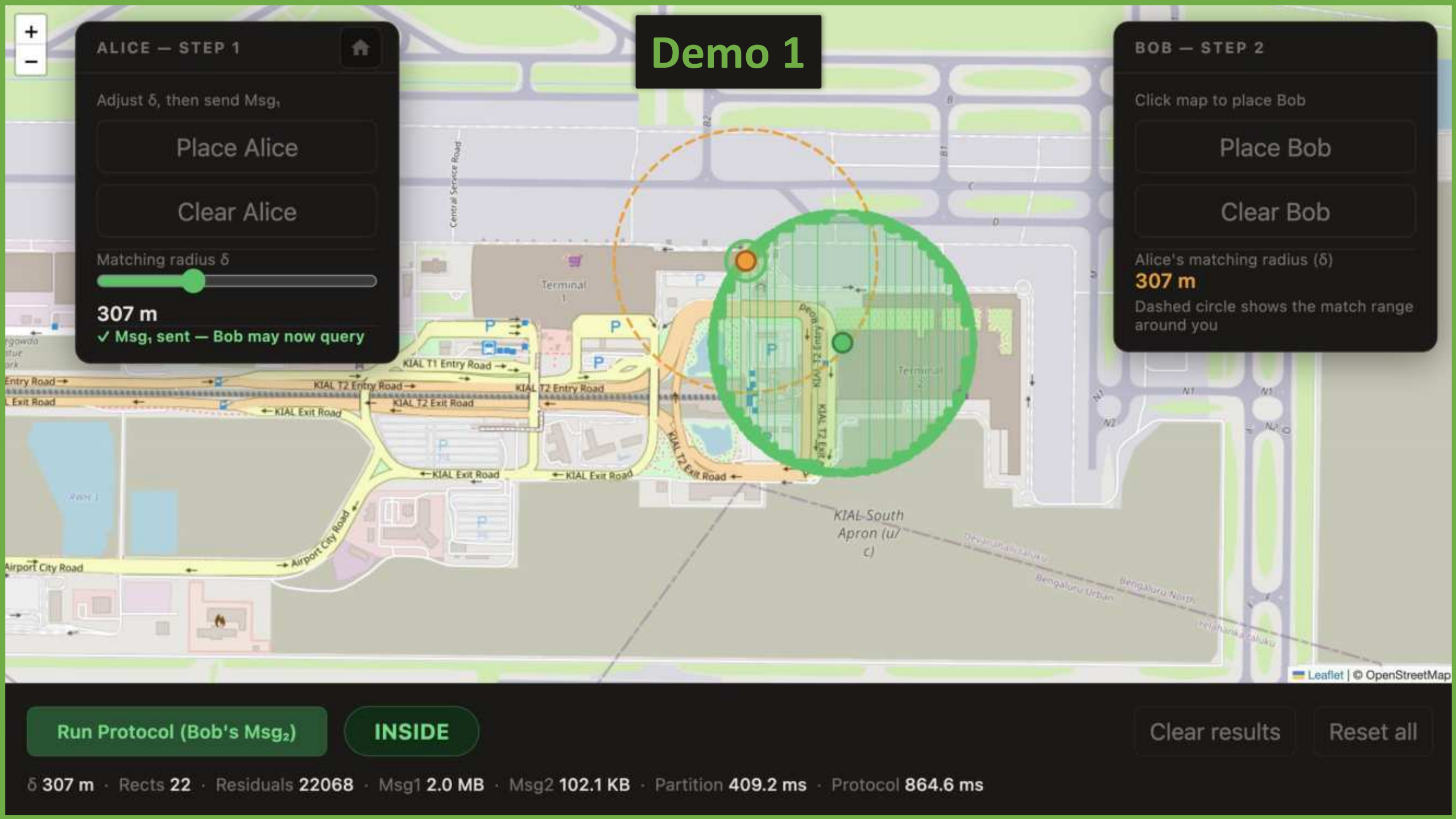
Alice authorizes a full floor of Skybridge Campus. Bob is placed in any room.

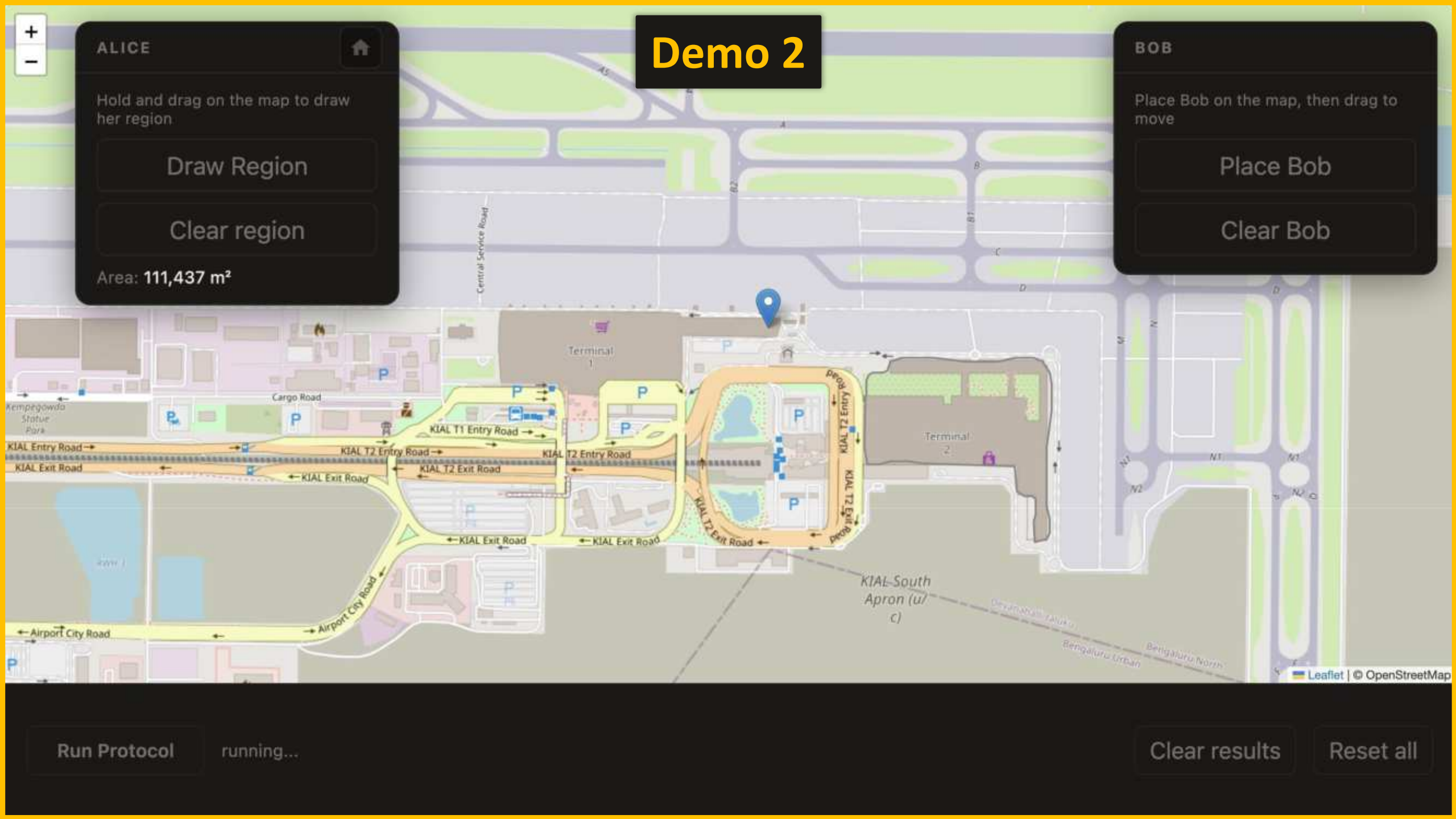
`Composite hyperrectangle region (Holub et al.)`

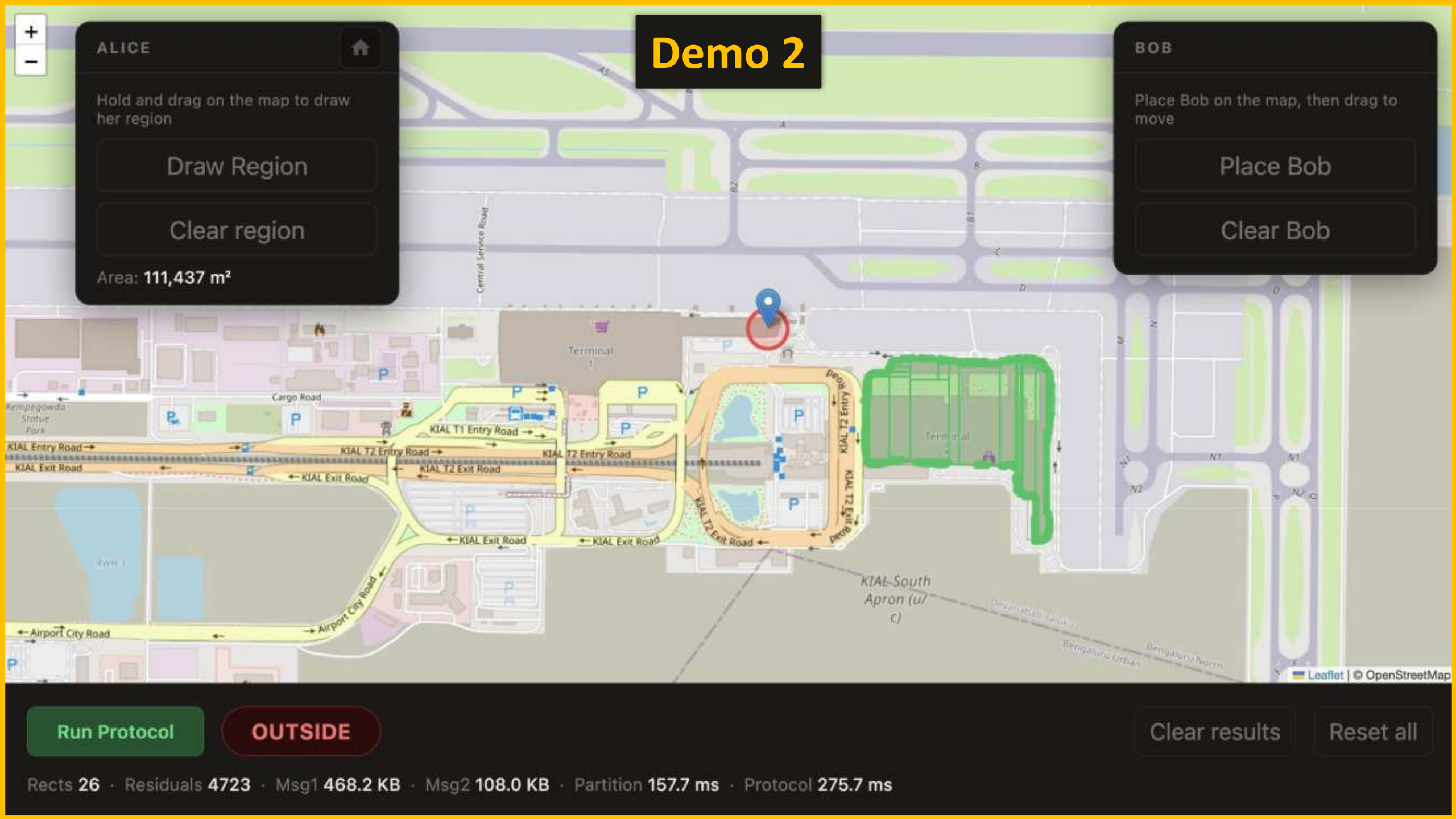

Demo 1
ALICE — STEP 1
Click map to place Alice
Place Alice
Clear Alice
BOB — STEP 2
Waiting for Alice's Msg₁...
Place Bob
Clear Bob
Terminal 1
Terminal 2
Cargo Road
KIAL T1 Entry Road
KIAL T2 Entry Road
KIAL T2 Exit Road
KIAL Exit Road
Airport City Road
KIAL South Apron (u/c)
Leaflet | © OpenStreetMap
Run Protocol (Bob's Msg₂)
Clear results
Reset all


Demo 1
ALICE — STEP 1
Adjust δ, then send Msg₁
Place Alice
Clear Alice
Matching radius δ
307 m
✓ Msg₁ sent — Bob may now query
BOB — STEP 2
Click map to place Bob
Place Bob
Clear Bob
Alice's matching radius (δ)
307 m
Dashed circle shows the match range around you
Terminal 1
Terminal 2
KIAL T1 Entry Road
KIAL T2 Entry Road
KIAL T2 Exit Road
KIAL Exit Road
Airport City Road
KIAL South Apron (u/c)
Leaflet | © OpenStreetMap
Run Protocol (Bob's Msg₂)
Clear results
Reset all


Demo 1
ALICE — STEP 1
Adjust δ, then send Msg₁
Place Alice
Clear Alice
Matching radius δ
307 m
✓ Msg₁ sent — Bob may now query
BOB — STEP 2
Click map to place Bob
Place Bob
Clear Bob
Alice's matching radius (δ)
307 m
Dashed circle shows the match range around you
Terminal 1
Terminal 2
KIAL T1 Entry Road
KIAL T2 Entry Road
KIAL T2 Exit Road
KIAL Exit Road
Airport City Road
KIAL South Apron (u/c)
Leaflet | © OpenStreetMap
Run Protocol (Bob's Msg₂)
Running...
Clear results
Reset all


Demo 1
ALICE — STEP 1
Adjust δ, then send Msg₁
Place Alice
Clear Alice
Matching radius δ
307 m
✓ Msg₁ sent — Bob may now query
BOB — STEP 2
Click map to place Bob
Place Bob
Clear Bob
Alice's matching radius (δ)
307 m
Dashed circle shows the match range around you
Run Protocol (Bob's Msg₂)
INSIDE
Clear results
Reset all
δ 307 m · Rects 22 · Residuals 22068 · Msg1 2.0 MB · Msg2 102.1 KB · Partition 409.2 ms · Protocol 864.6 ms
Leaflet | © OpenStreetMap


Demo 2
ALICE
Hold and drag on the map to draw her region
Draw Region
Clear region
Area: 111,437 m²
BOB
Place Bob on the map, then drag to move
Place Bob
Clear Bob
Terminal 1
Terminal 2
Cargo Road
KIAL Entry Road
KIAL Exit Road
KIAL T1 Entry Road
KIAL T2 Entry Road
KIAL T2 Exit Road
KIAL South Apron (u/c)
Airport City Road
Central Service Road
Leaflet | © OpenStreetMap
Run Protocol
running...
Clear results
Reset all


Demo 2
ALICE
Hold and drag on the map to draw her region
Draw Region
Clear region
Area: 111,437 m²
BOB
Place Bob on the map, then drag to move
Place Bob
Clear Bob
Terminal 1
Terminal 2
Cargo Road
KIAL T1 Entry Road
KIAL T2 Entry Road
KIAL Entry Road
KIAL Exit Road
KIAL T2 Exit Road
KIAL T2 Entry Road
Airport City Road
KIAL South Apron (u/c)
Leaflet | © OpenStreetMap
Run Protocol
OUTSIDE
Clear results
Reset all
Rects 26 · Residuals 4723 · Msg1 468.2 KB · Msg2 108.0 KB · Partition 157.7 ms · Protocol 275.7 ms


Demo 3
ALICE
Authorized floor -- all rooms included
Choose a floor...
No floor selected
BOB
Floor
Choose a floor...
Room (placed at center)
Select a floor first
No position set
Run Protocol
Clear & Reset


Demo 3
ALICE
Authorized floor -- all rooms included
F2
F2 -- 8 rooms -- 13,600 m²
BOB
Floor
Choose a floor...
Room (placed at center)
Select a floor first
No position set
Run Protocol
Clear & Reset


Demo 3
ALICE
Authorized floor -- all rooms included
F2
F2 -- 8 rooms -- 13,600 m²
BOB
Floor
F4
Room (placed at center)
Observatory B
F4 -- Observatory B
Run Protocol
Clear & Reset
OUTSIDE
Done.
Rects 3 · Residuals 0 · Msg1 11.7 KB · Msg2 27.6 KB · Partition 0 ms · Protocol 81 ms

Conclusion and Future Work

Our Results

Efficient FM for ℓ_∞ in low dimensions

Novel partitioning algorithms

Geometric Fuzzy Matching

new primitive for arbitrary geometric region membership

Privacy-Preserving Proximity Testing for complex layouts

Conclusion and Future Work

Our Results

Efficient FM for ℓ_∞ in low dimensions

Novel partitioning algorithms

Geometric Fuzzy Matching

new primitive for arbitrary geometric region membership

Privacy-Preserving Proximity Testing for complex layouts

Future Work

Dynamic environments for moving entities

Full hiding of *region layout* from the sender

Support for repeated queries

Conclusion and Future Work

Our Results

Efficient FM for ℓ_∞ in low dimensions

Novel partitioning algorithms

Geometric Fuzzy Matching

new primitive for arbitrary geometric region membership

Privacy-Preserving Proximity Testing for complex layouts

Future Work

Dynamic environments for moving entities

Full hiding of *region layout* from the sender

Support for repeated queries

Thank you!

