Autonome Systeme

Der Homing-Algorithmus

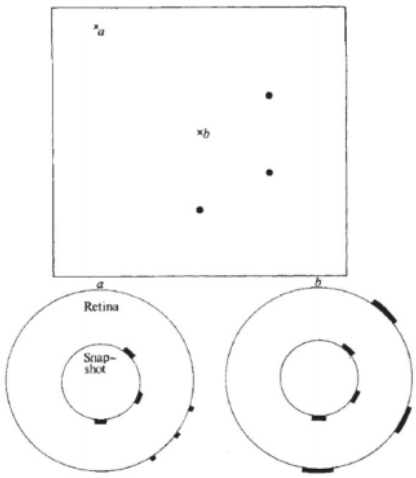
Chunrong Yuan

Labor für Autonome Systeme

Technology
Arts Sciences
TH Köln

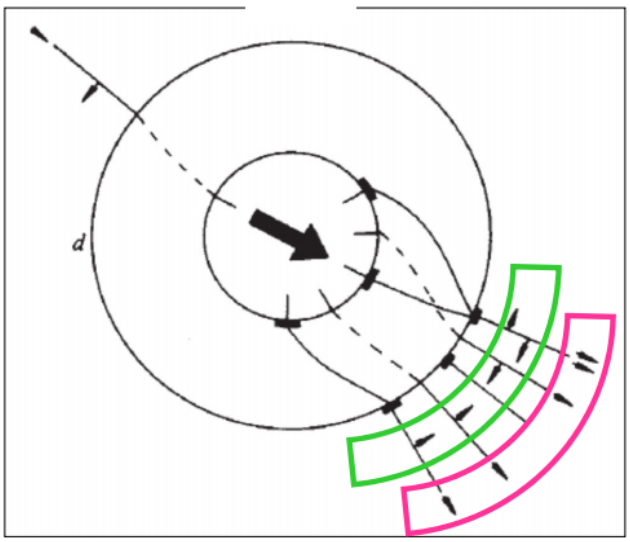
Technische Hochschule Köln

How honey bees use landmarks to guide their return to a food source



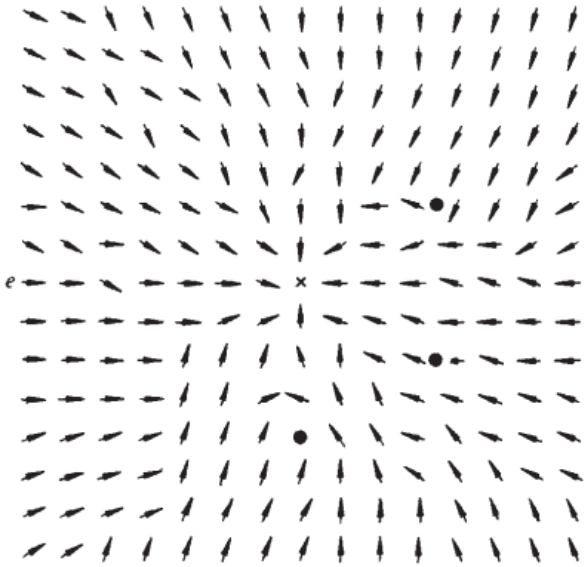
Top: configuration with three landmarks and two marked positions

Bottom: inner circle shows panoramic snapshot obtained at position *b*, outer circle shows current image visible at positions *a* and *b*, resp.

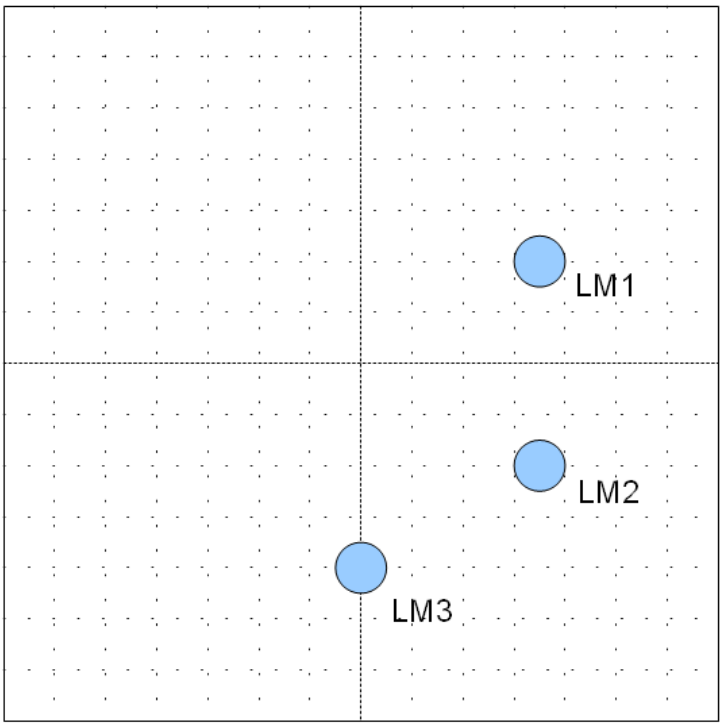


Snapshot algorithm. Features (black sectors) and gaps are matched. If, in current image, feature or gap is smaller than in snapshot, approach vector is activated. If position is wrong, turn vector is activated. Agent moves in direction of average vector.

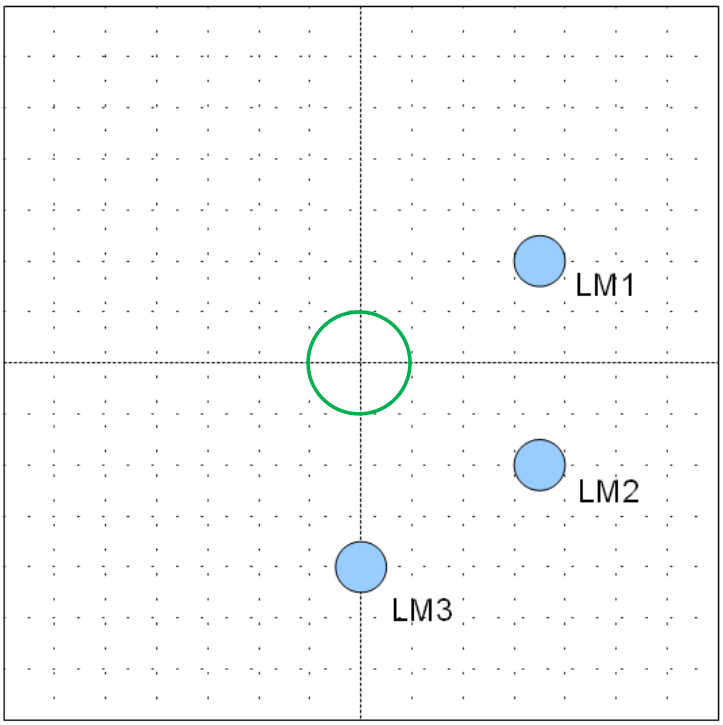
How honey bees use landmarks to guide their return to a food source



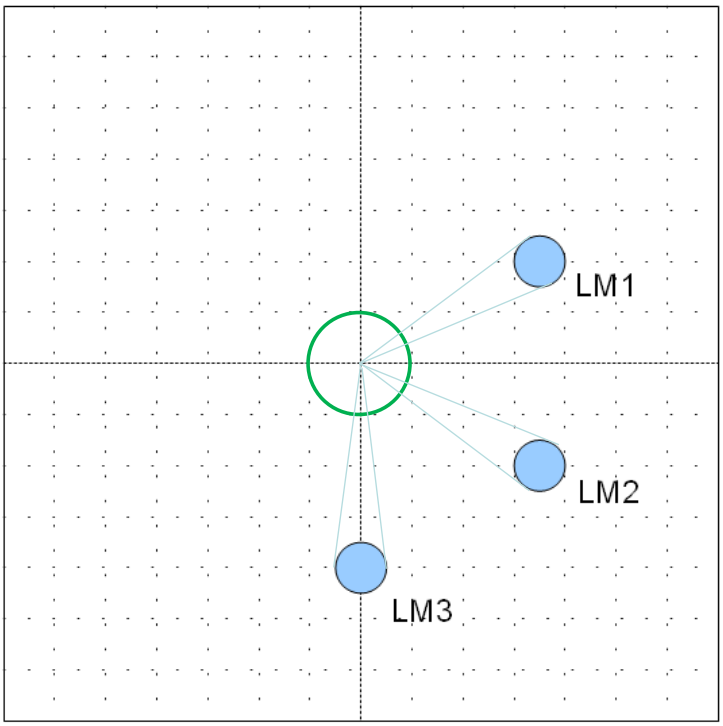
Homing with three Landmarks



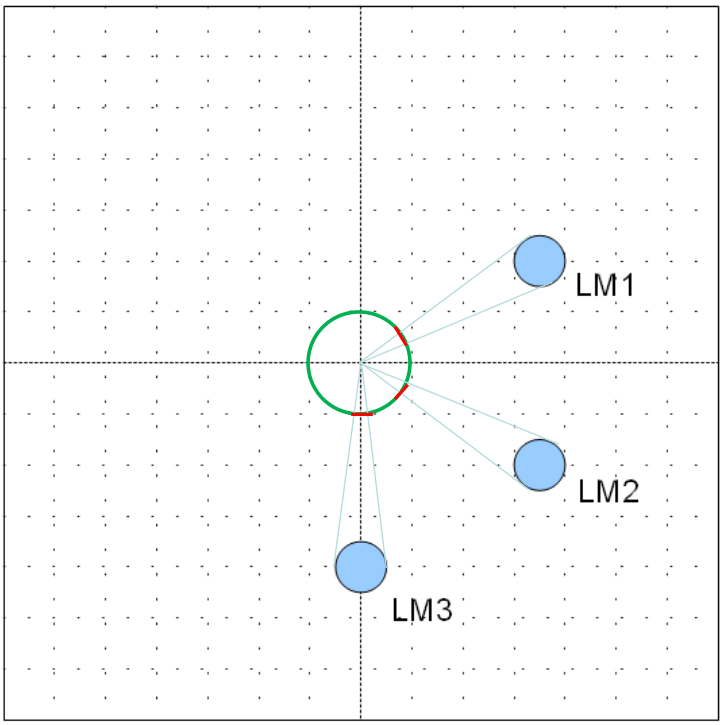
Snapshot



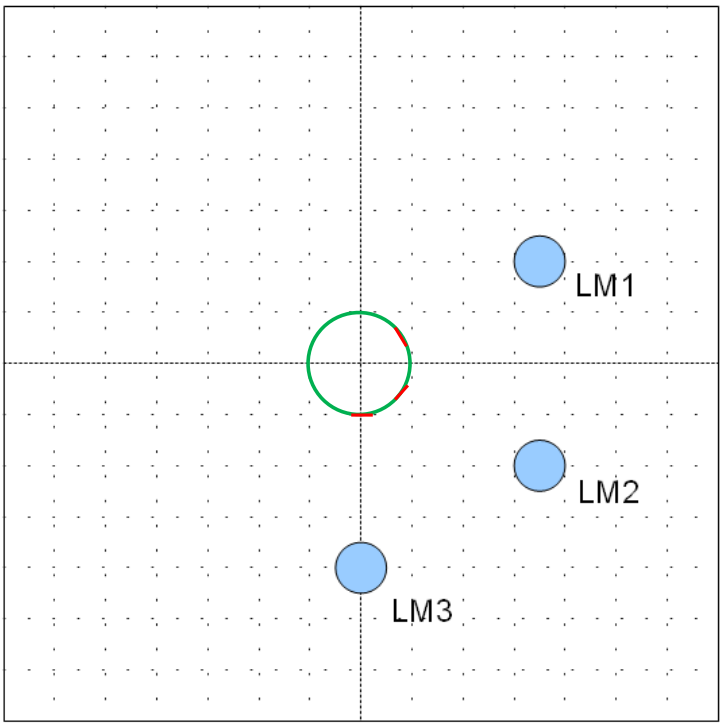
Map the landmarks onto snapshot



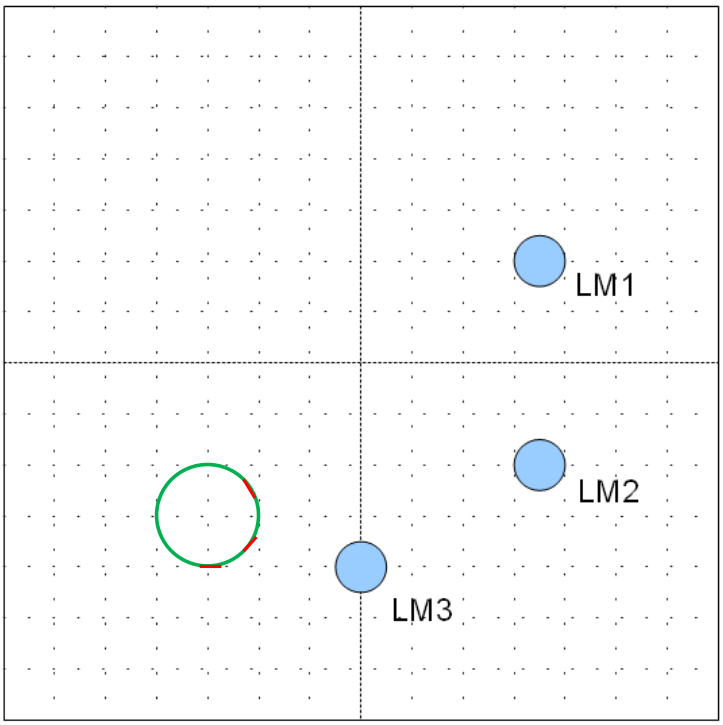
Map the landmarks onto snapshot resulting in six model features



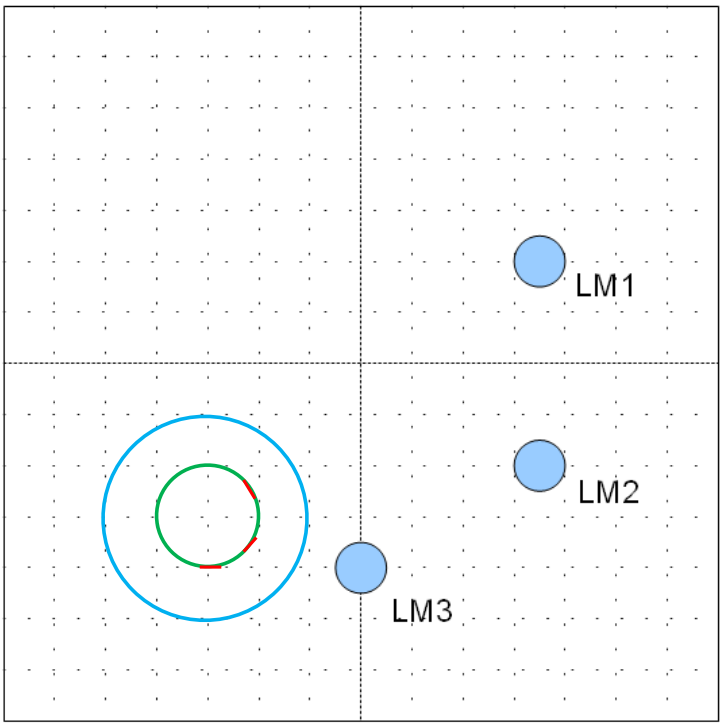
Map the landmarks onto snapshot resulting in six model features



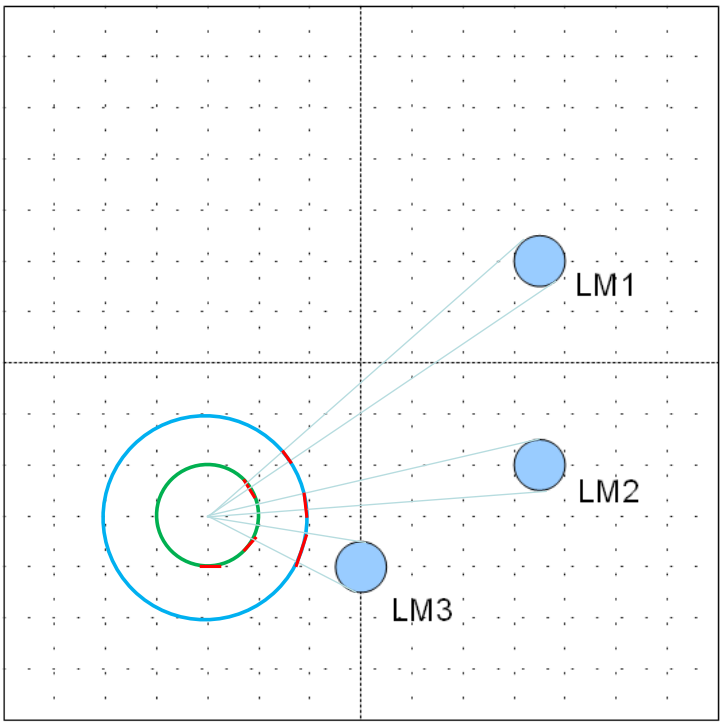
Shift the snapshot to a new position (x,y)



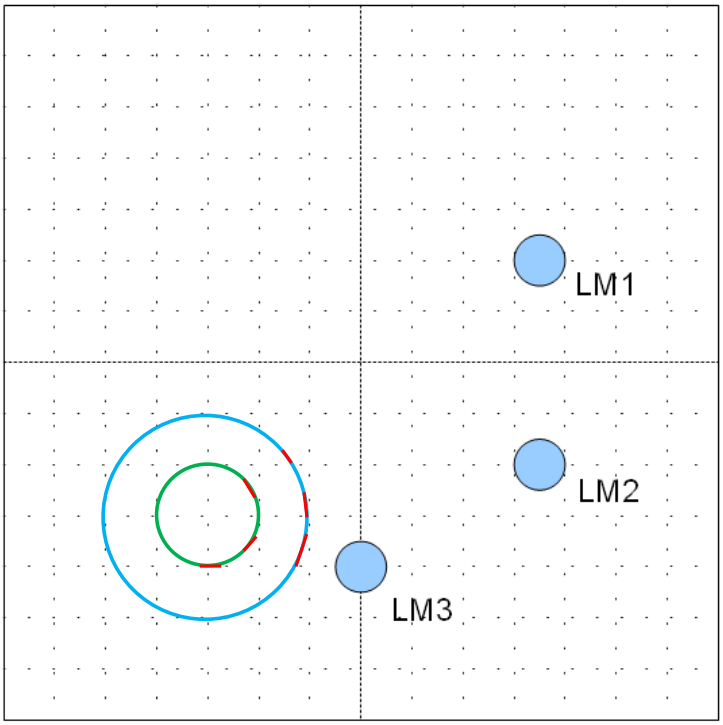
Map the landmarks onto retina



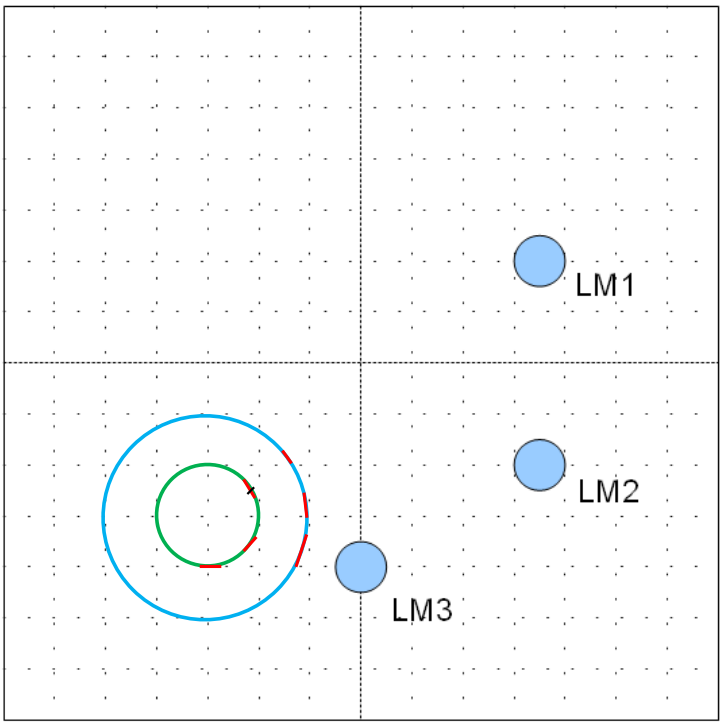
Map the landmarks onto retina



Map the landmarks onto retina resulting in six retina features

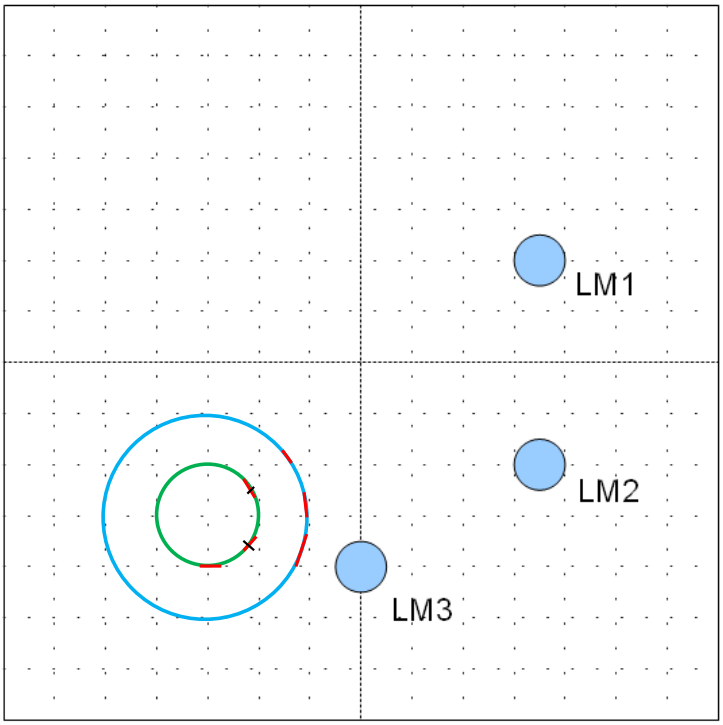


Calculate the model features



(O_m1, L_m1)

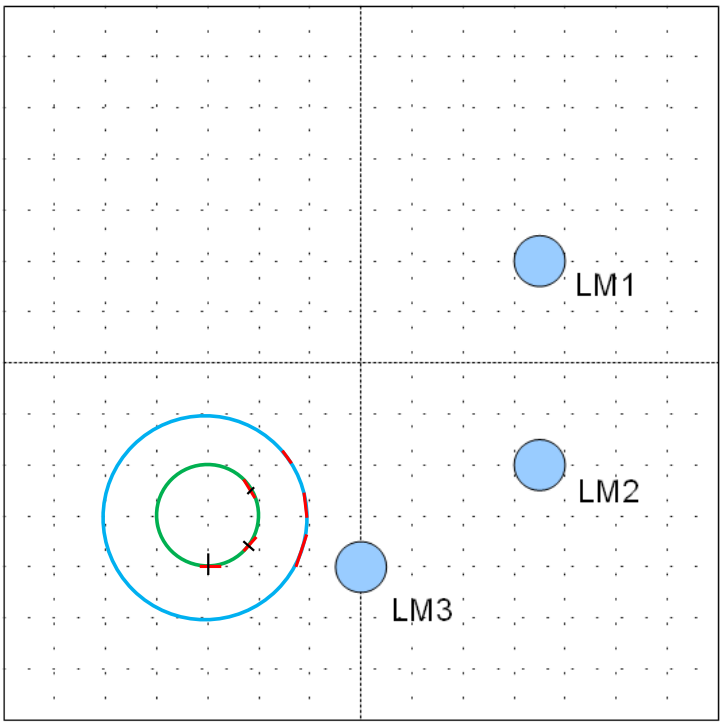
Calculate the model features



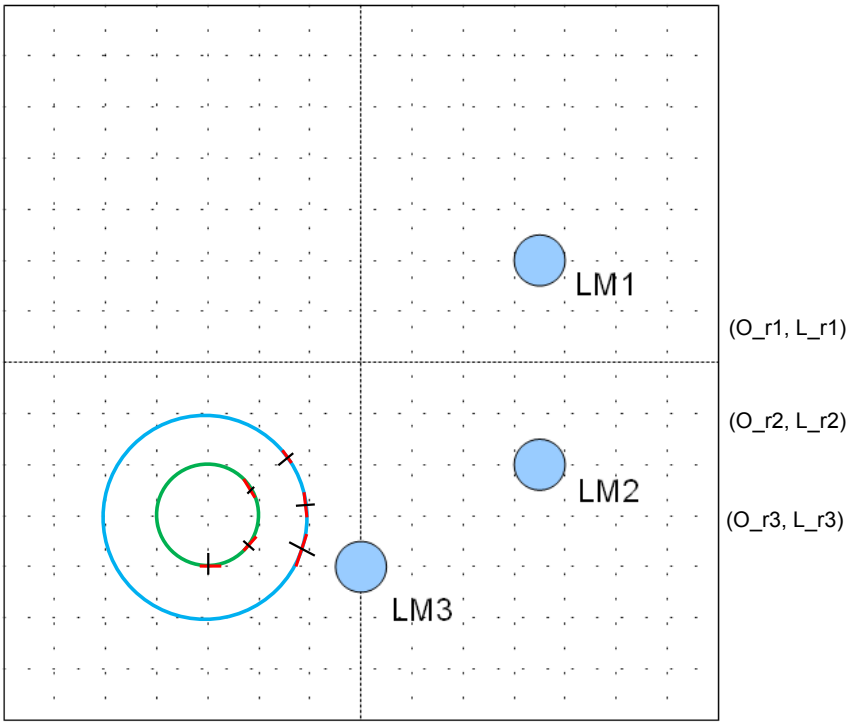
(O_m1, L_m1)

(O_m2, L_m2)

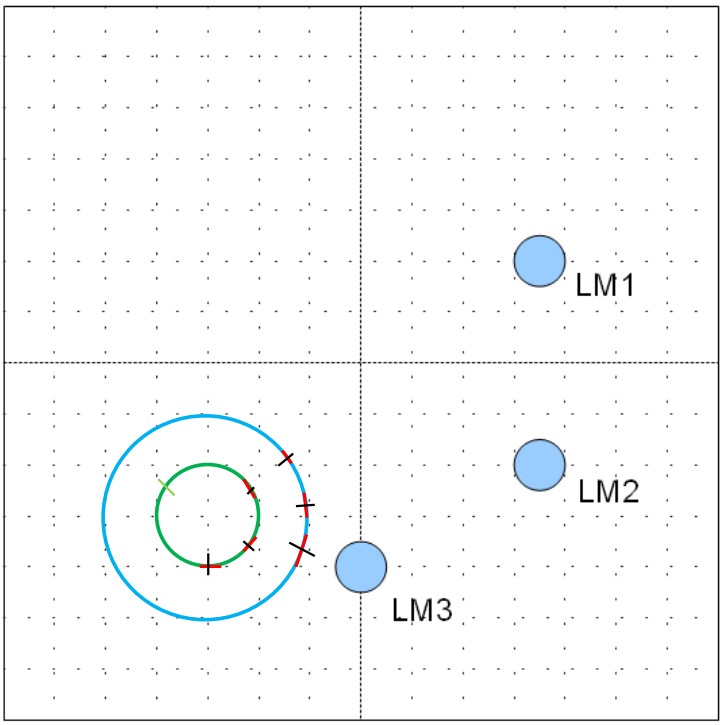
Calculate the model features



Calculate the retina features

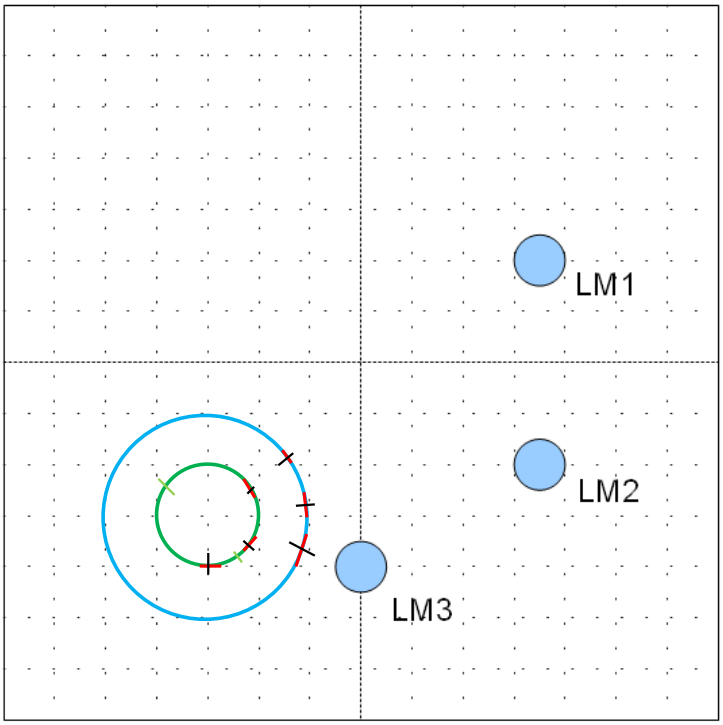


Calculate the model features



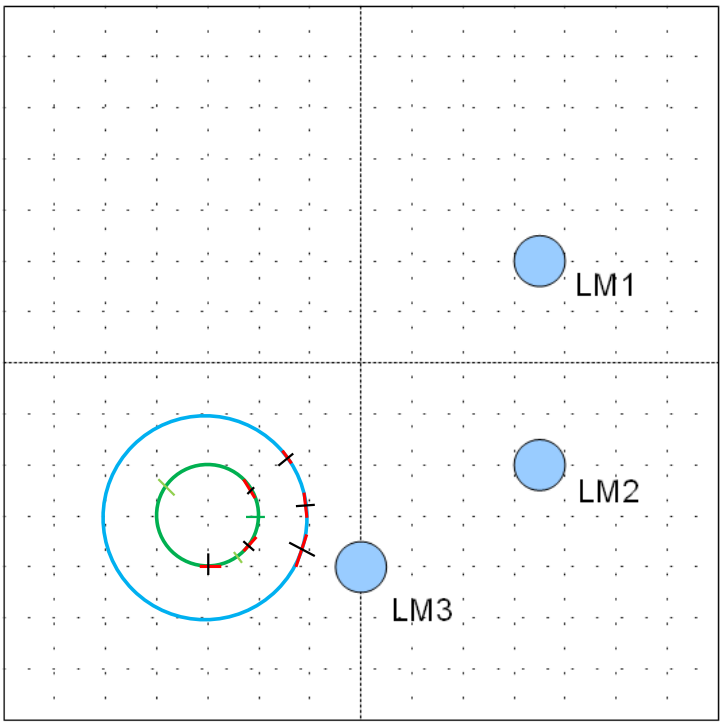
(O_m4, L_m4)

Calculate the model features



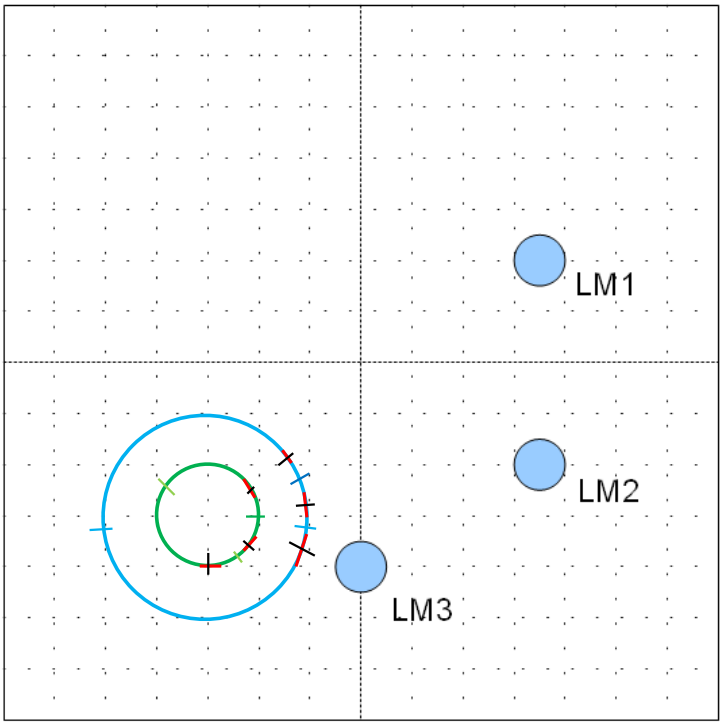
(O_m5, L_m5)

Calculate the model features



(O_m6, L_m6)

Calculate the retina features

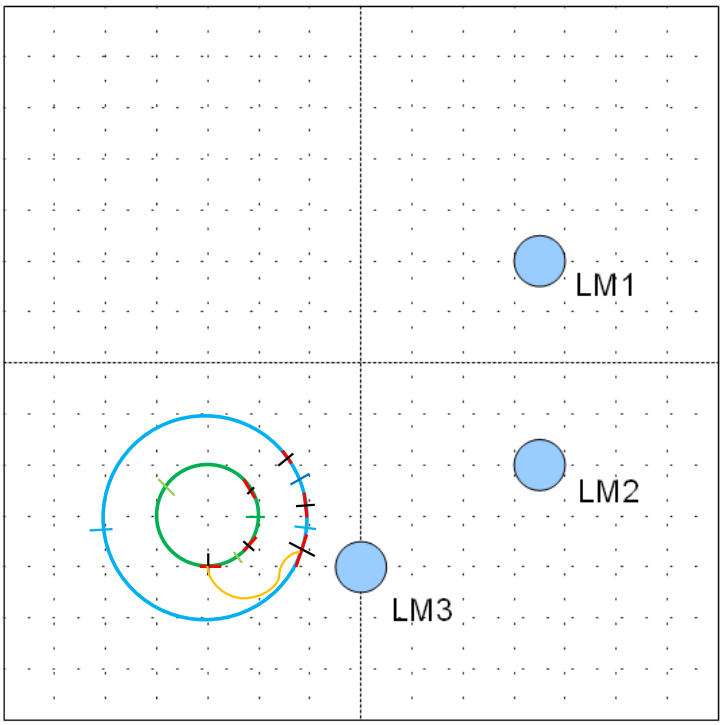


(O_r4, L_r4)

(O_r5, L_r5)

(O_r6, L_r6)

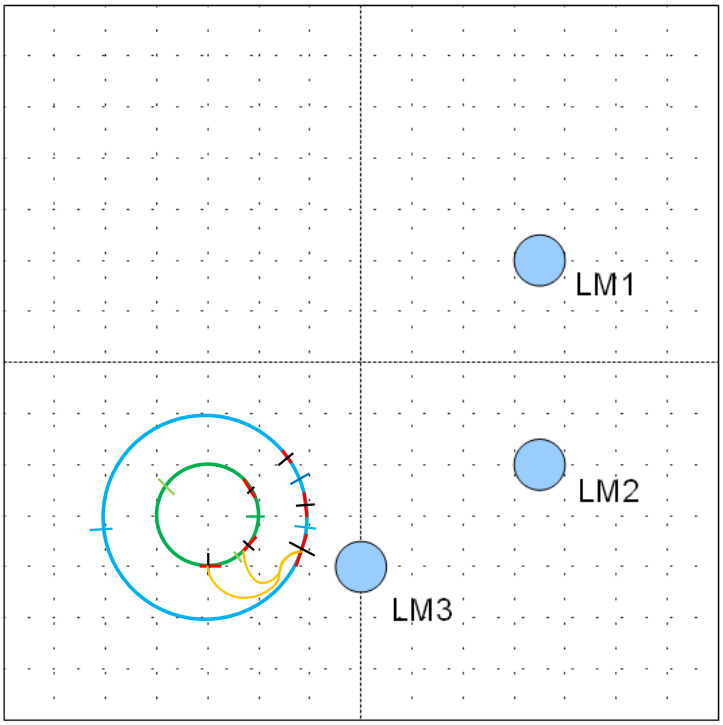
Matching model and retina features



Match O_m3 to
O_r1, O_r2, O_r3

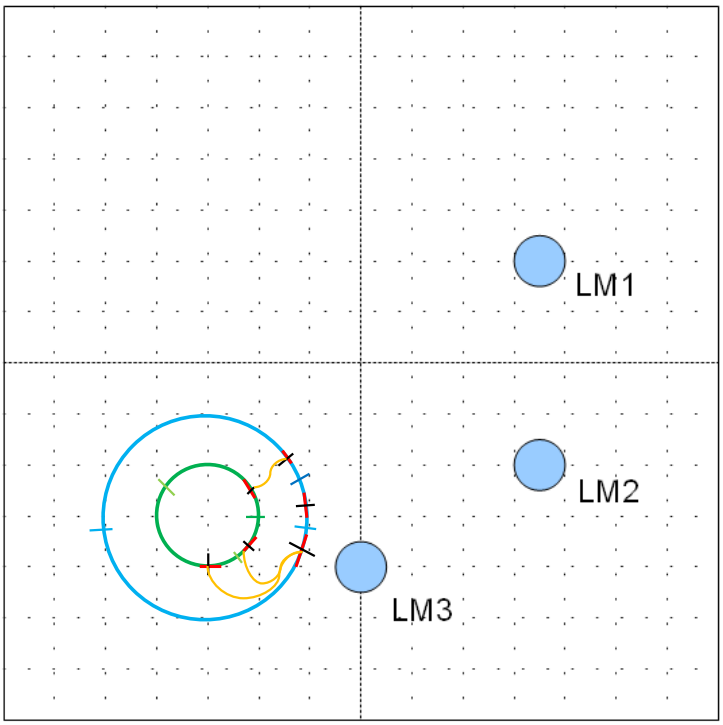
The matched one is O_r3, as
angleDistance(O_m3, O_r3)
Is the smallest one

Matching model and retina features



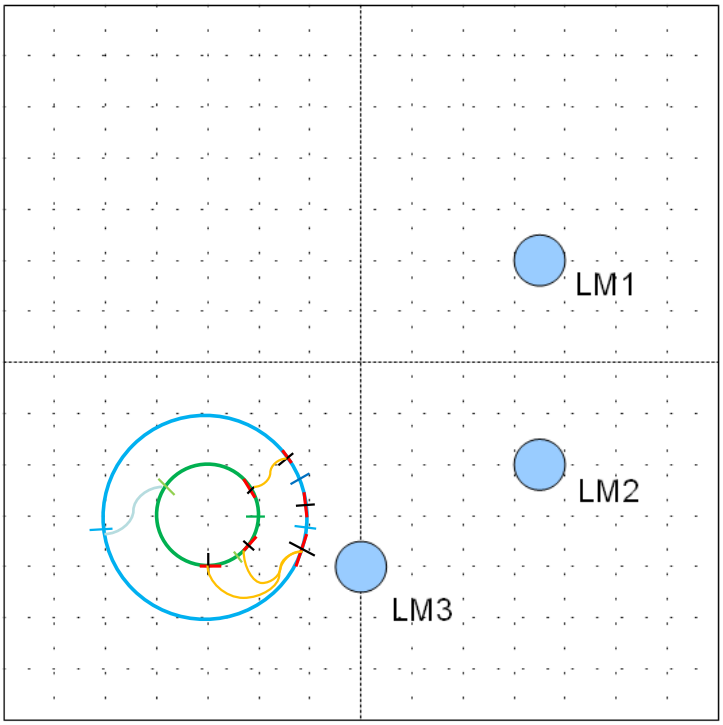
O_m2 is matched
also to O_r3

Matching model and retina features



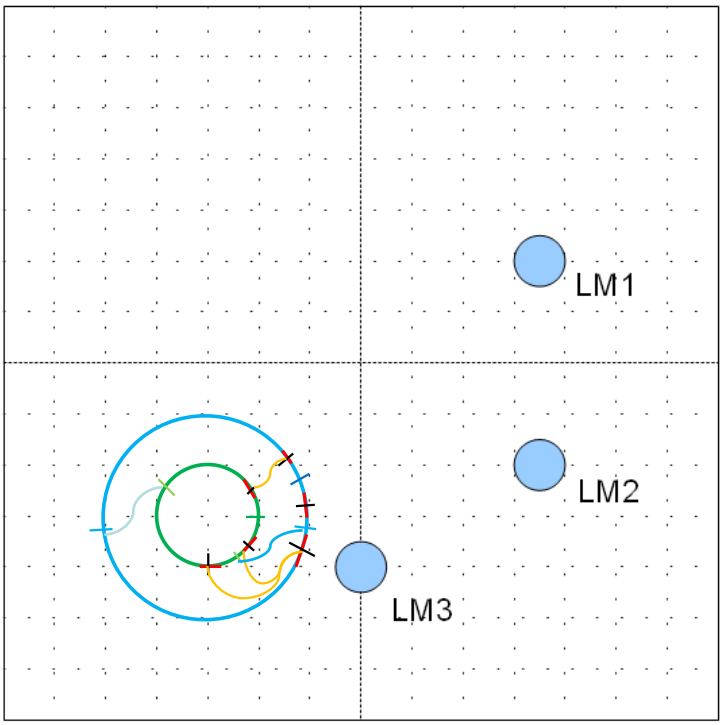
O_m1 is matched to O_r1

Matching model and retina features



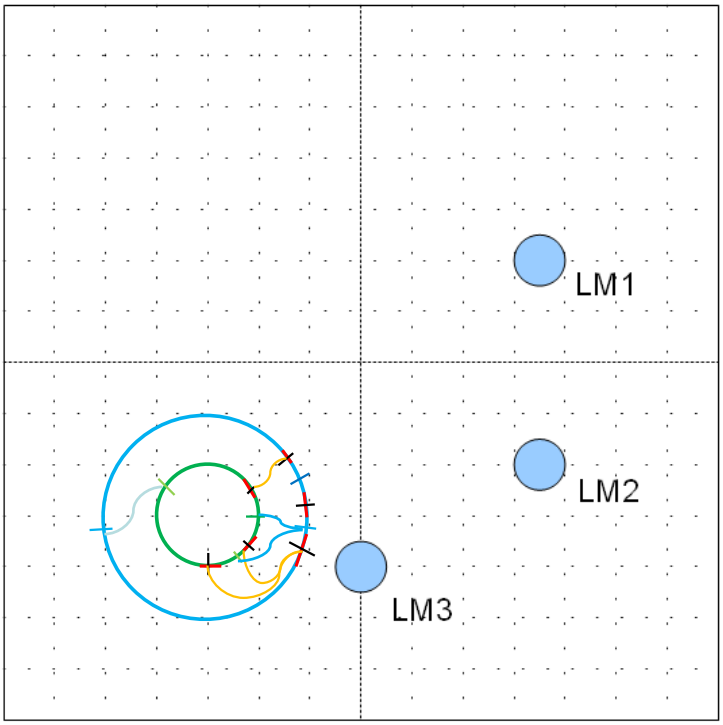
O_m4 is matched to O_r4

Matching model and retina features



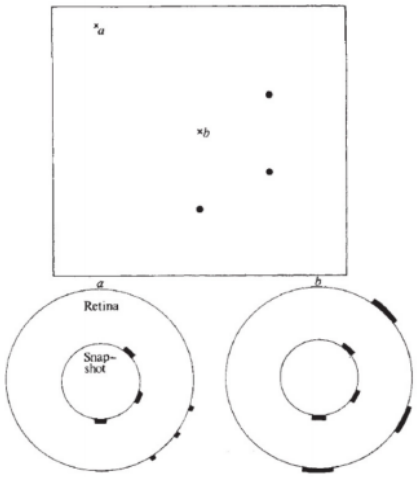
O_m5 is matched
To O_r5

Matching model and retina features



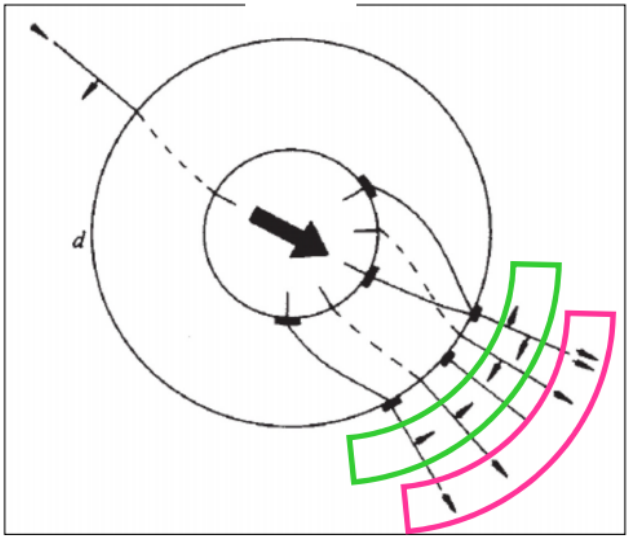
O_m6 is matched
also to O_r5

The snapshot model



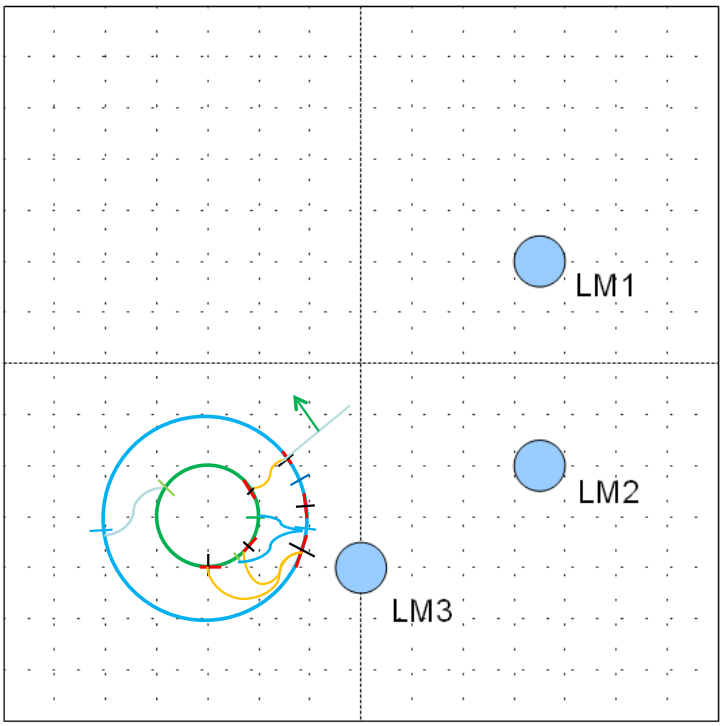
Top: configuration with three landmarks and two marked positions

Bottom: inner circle shows panoramic snapshot obtained at position *b*, outer circle shows current image visible at positions *a* and *b*, resp.

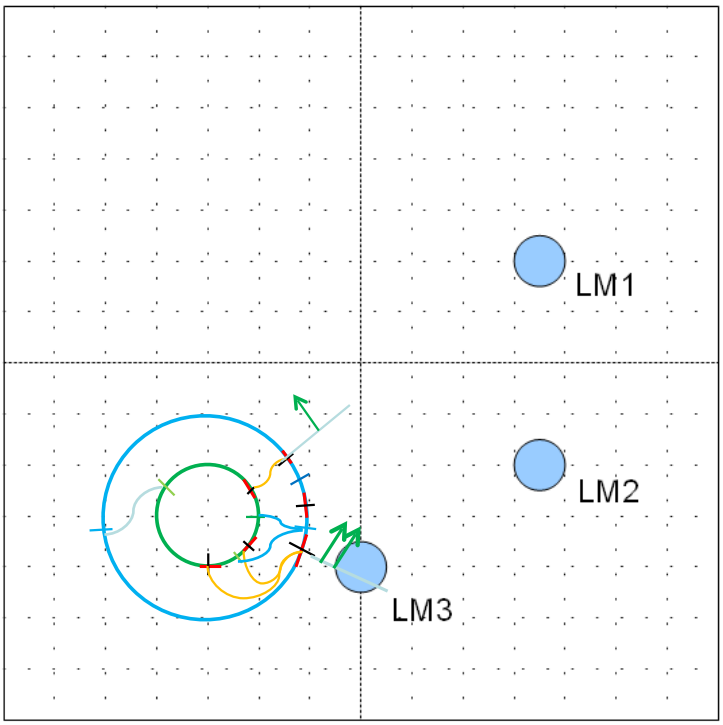


Snapshot algorithm. Features (black sectors) and gaps are matched. If, in current image, feature or gap is smaller than in snapshot, approach vector is activated. If position is wrong, turn vector is activated. Agent moves in direction of average vector.

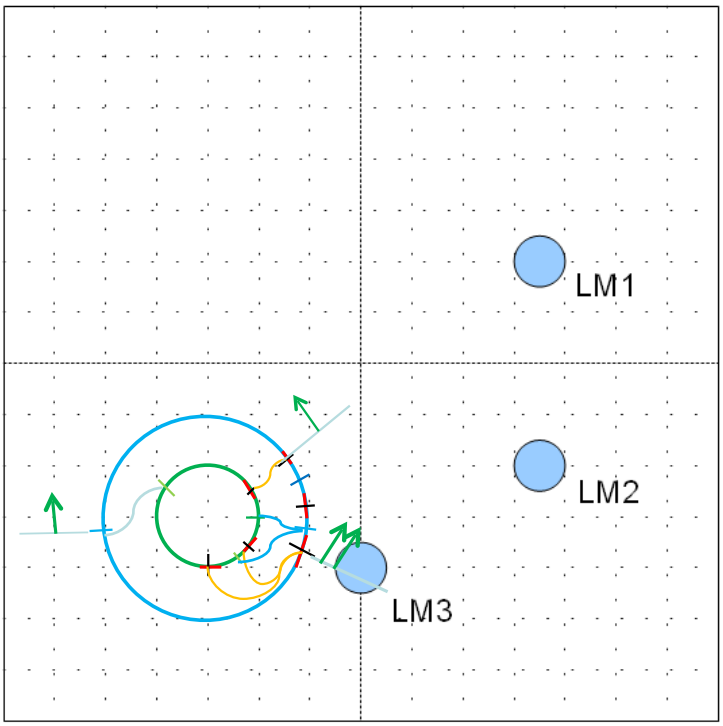
Calculating the turn vectors



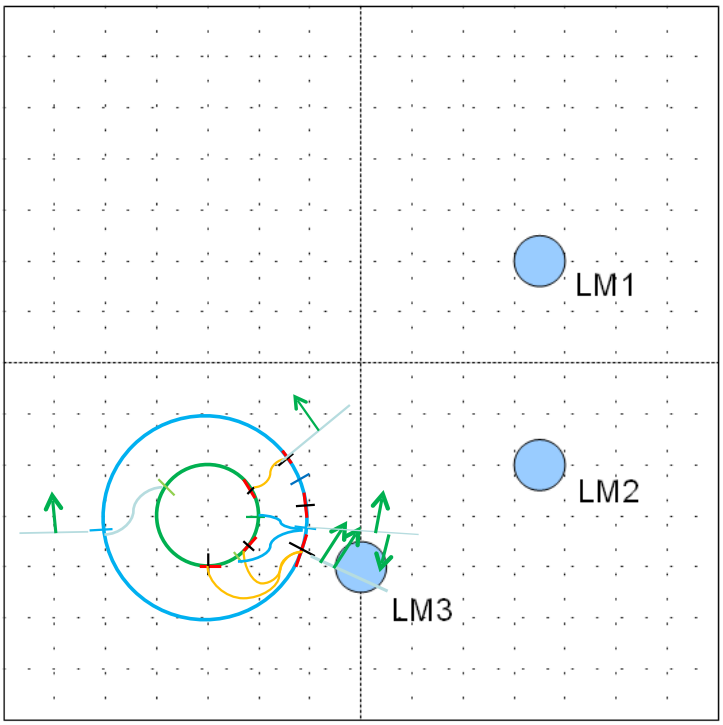
Calculating the turn vectors



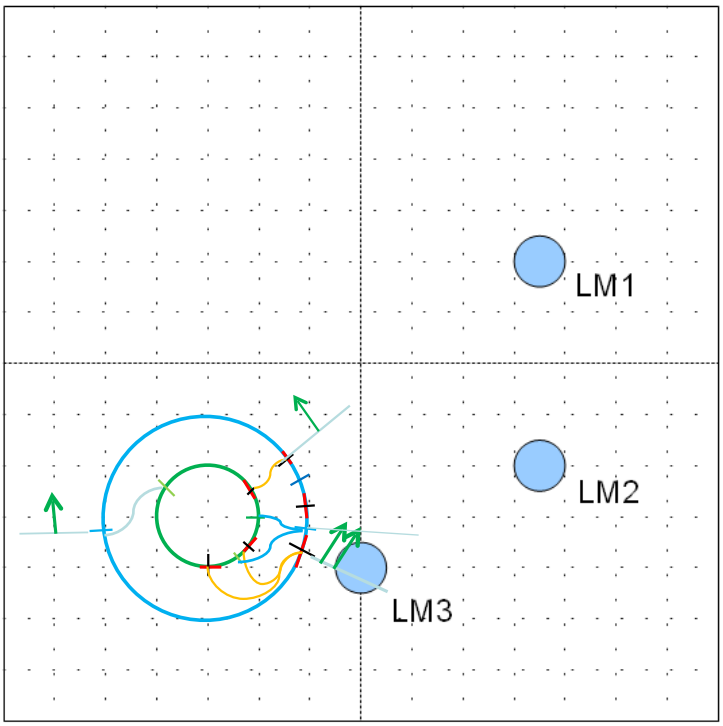
Calculating the turn vectors



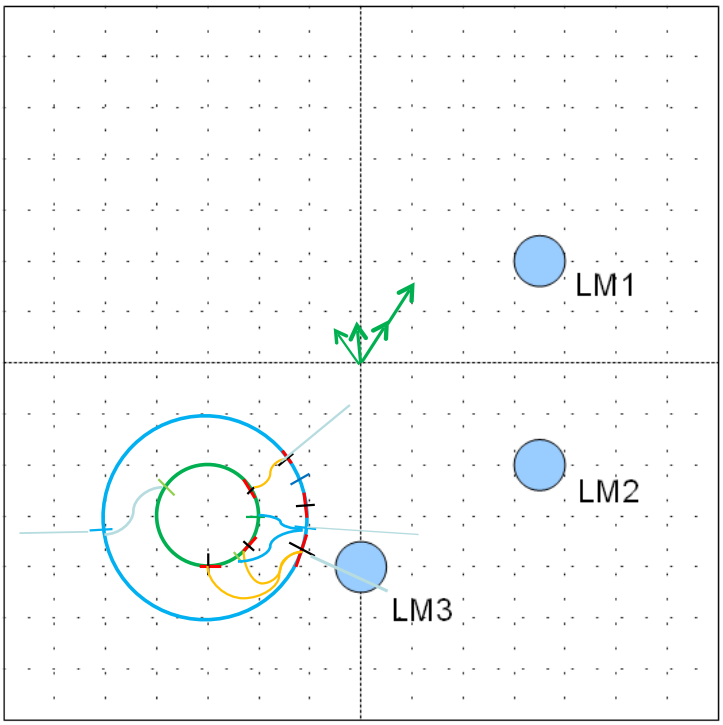
Calculating the turn vectors



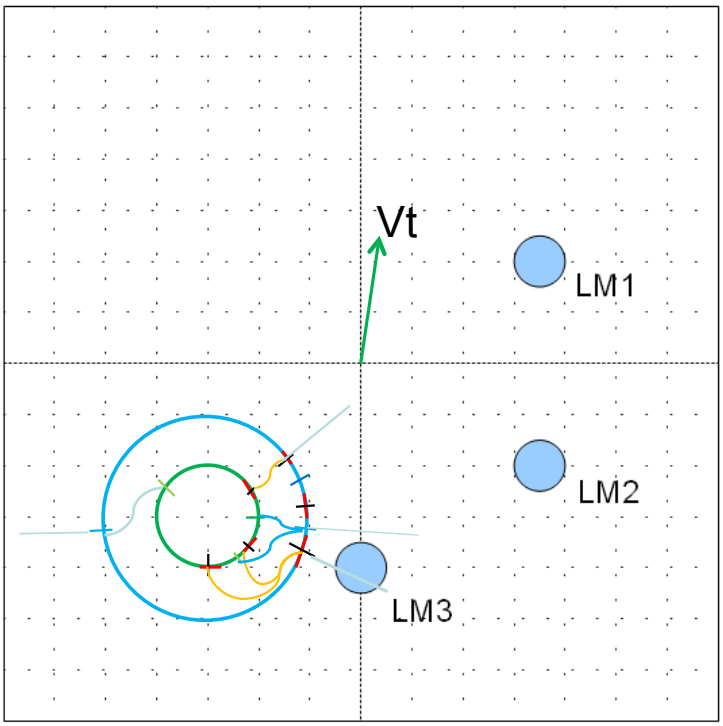
Calculating the turn vectors



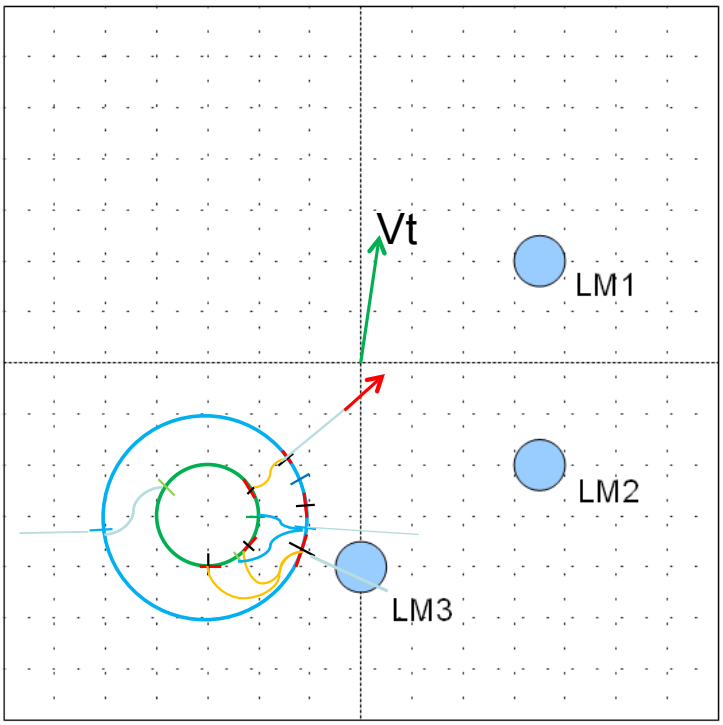
Calculating the turn vectors



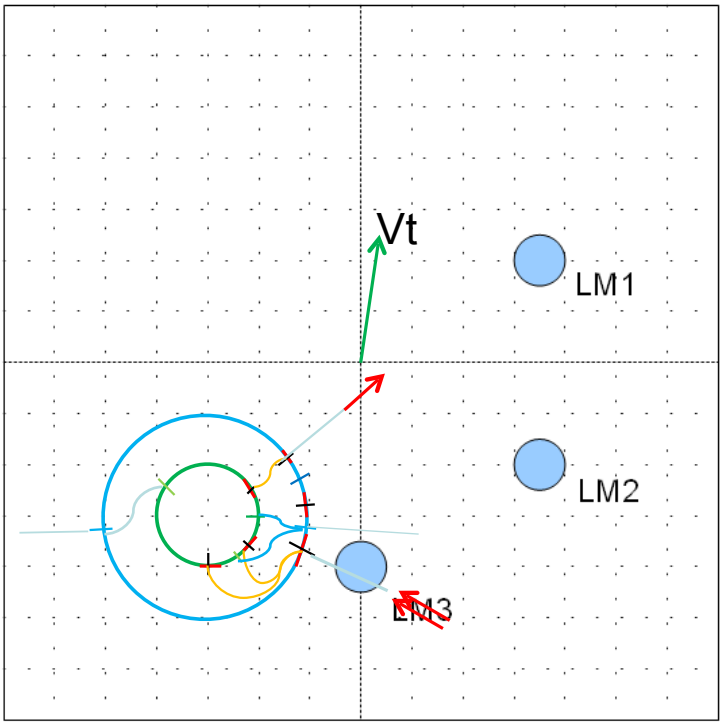
Calculating the turn vectors



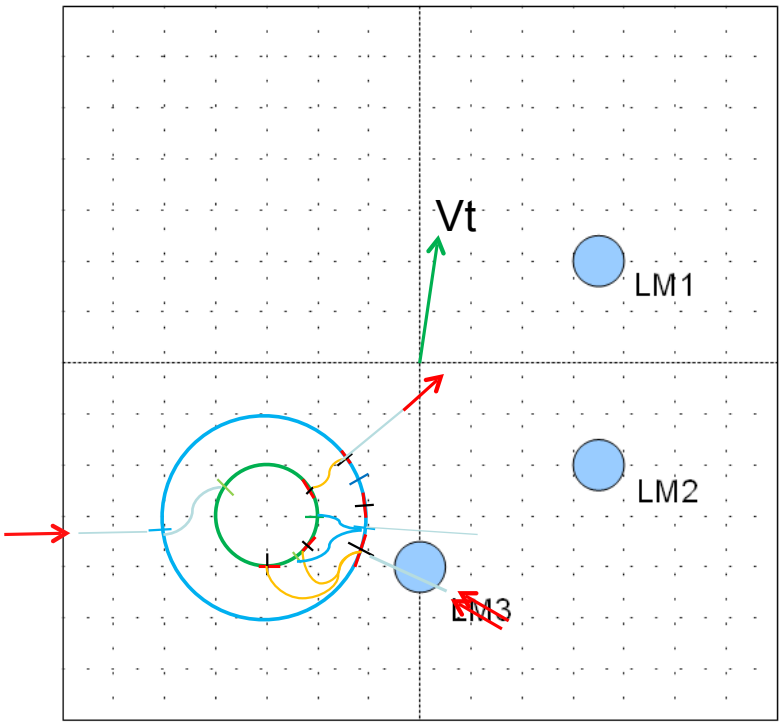
Calculating the position vectors



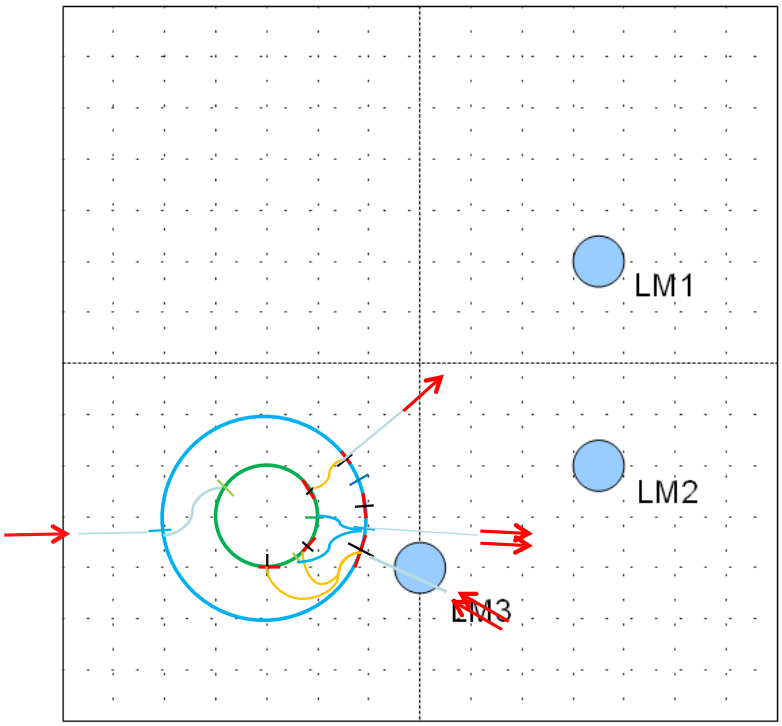
Calculating the position vectors



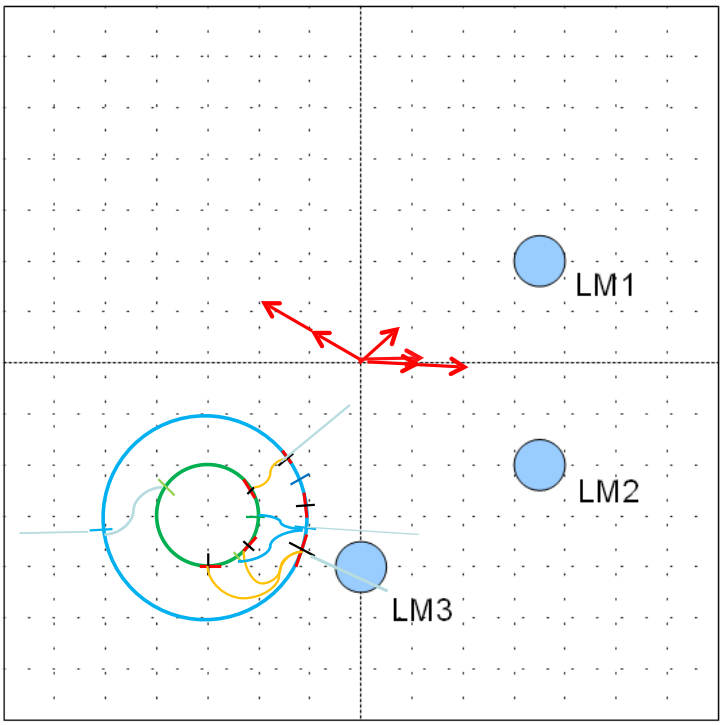
Calculating the position vectors



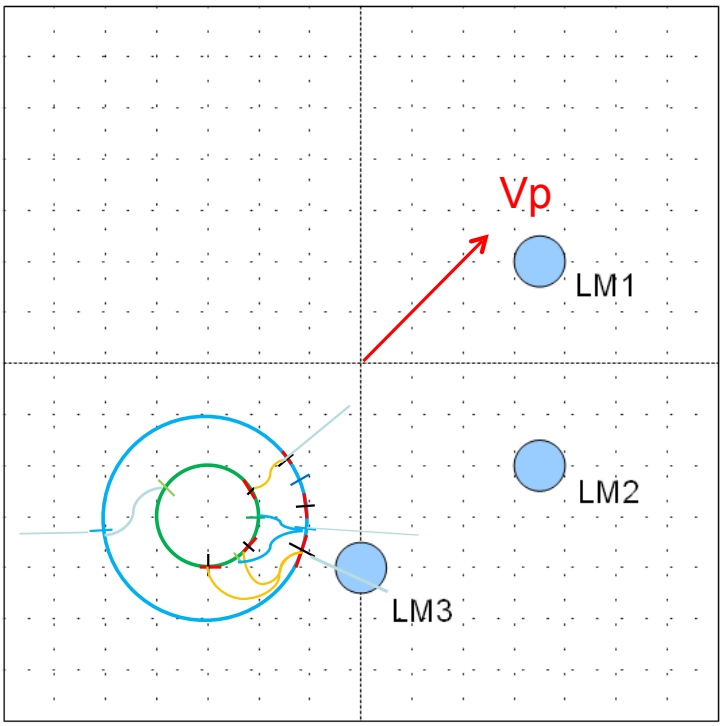
Calculating the position vectors



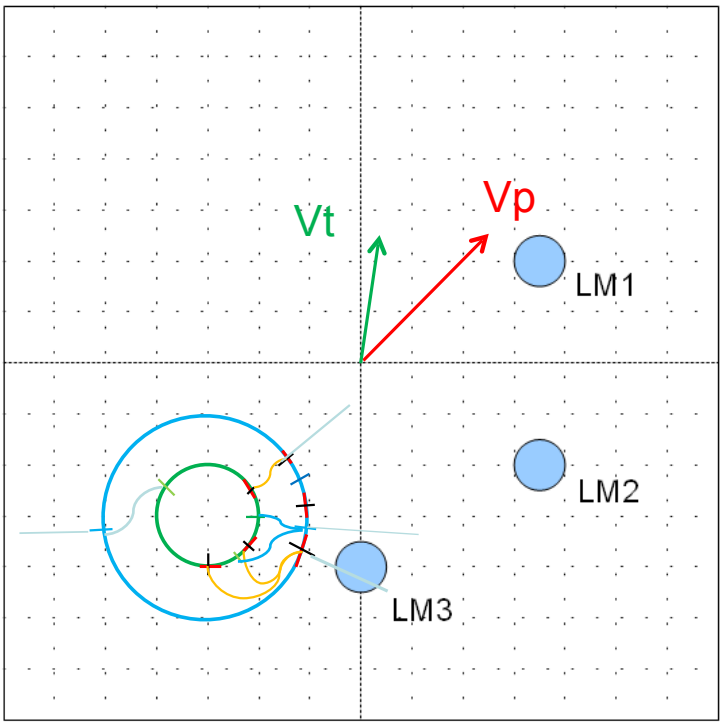
Calculating the position vectors



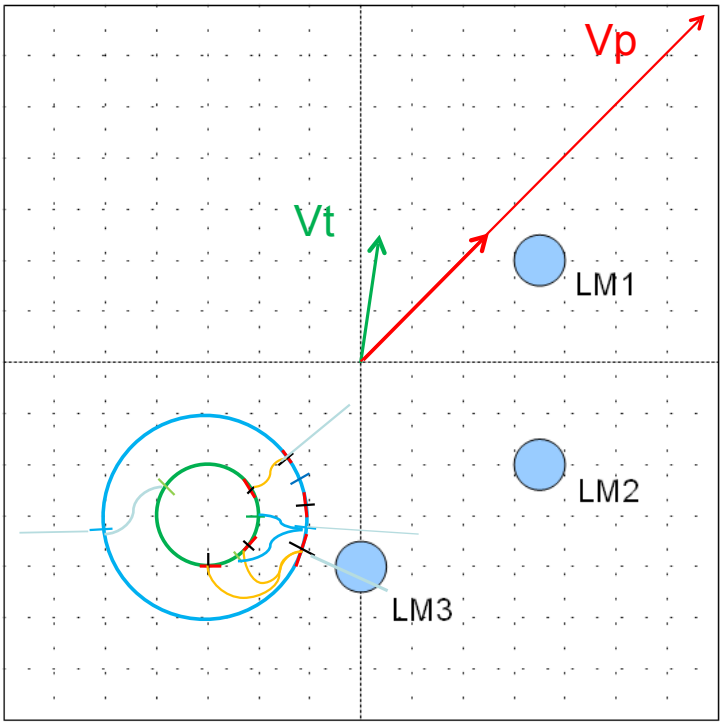
Calculating the position vector



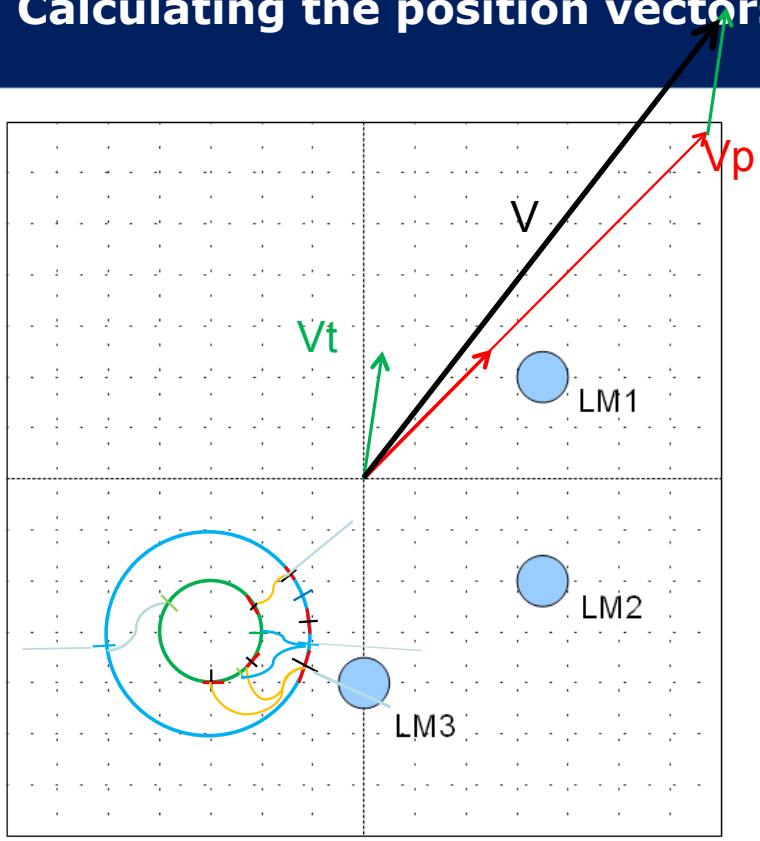
Calculating the final homing vector



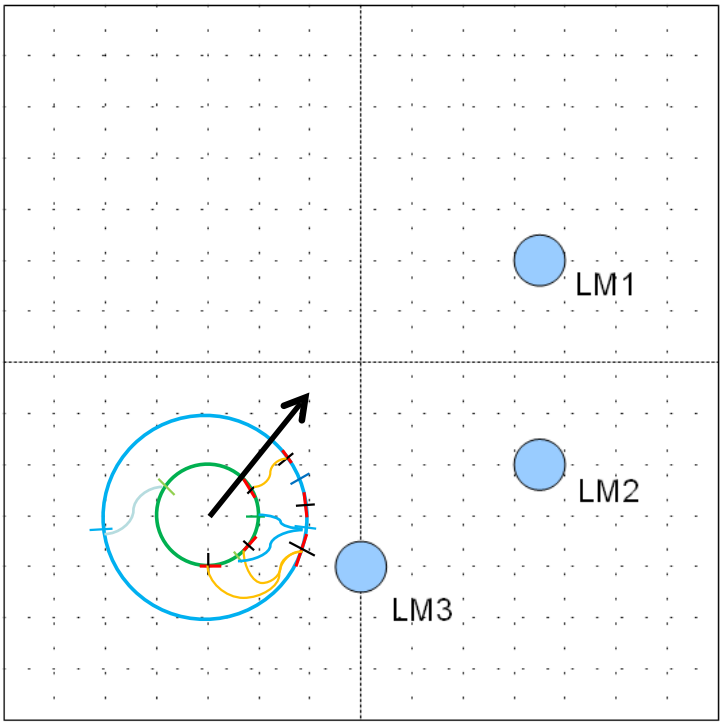
Calculating the final homing vector



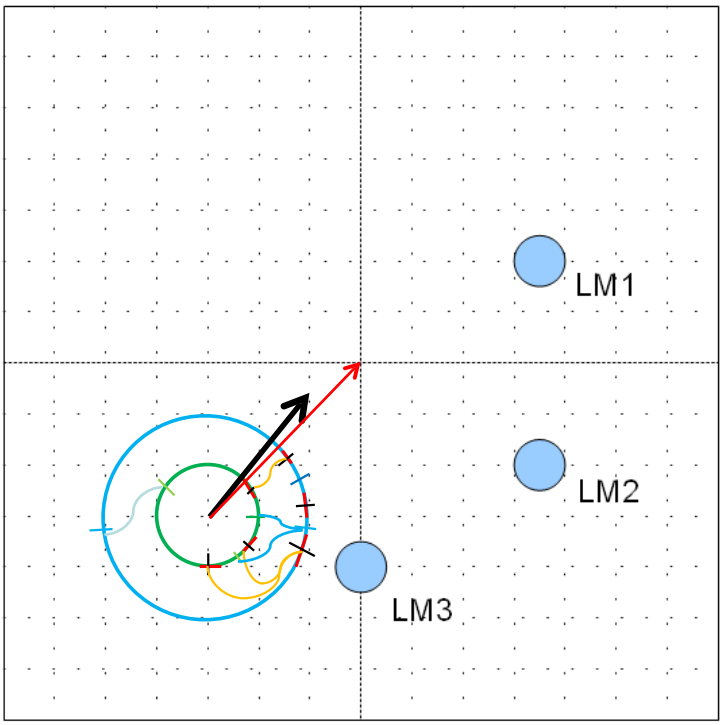
Calculating the position vectors



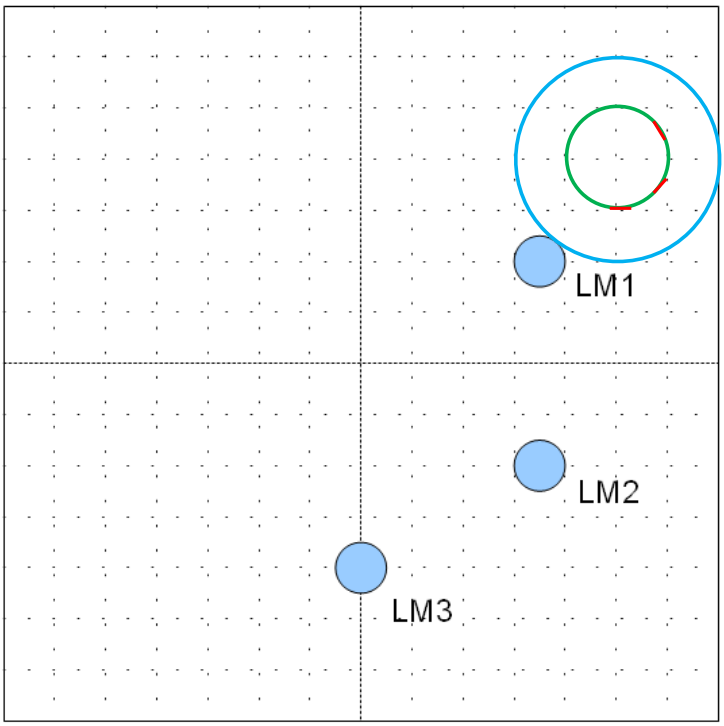
Calculating the position vectors



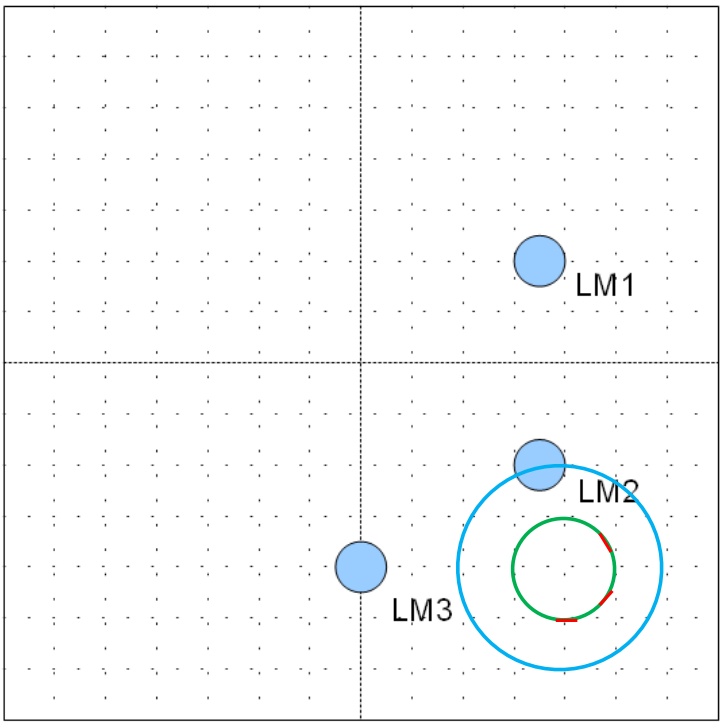
Calculating the position vectors



Map the landmarks onto retina



Map the landmarks onto retina



Homing vectors