
Algorithm 1 MergeLast - Finding flock patterns looking δ timestamps ahead.

Input: Set of spatio-temporal points P , maximum distance ϵ , minimum size μ , minimum duration δ , maximum traveled distance τ

Output: Set of flock patterns F

```
 $T \leftarrow$  sorted set of timestamps in  $P$ 
for each timestamp  $t_i$  in  $T - \delta$  do
   $P_i \leftarrow$  Set of points at timestamp  $t_i$ 
   $C_i \leftarrow \text{getMaximalDisks}(P_i, \epsilon, \mu)$ 
   $P_{i+\delta} \leftarrow$  Set of points at timestamp  $t_{i+\delta}$ 
   $C_{i+\delta} \leftarrow \text{getMaximalDisks}(P_{i+\delta}, \epsilon, \mu)$ 
  if  $C_i \neq \emptyset$  and  $C_{i+\delta} \neq \emptyset$  then
     $F' \leftarrow C_i \bowtie_{\tau} C_{i+\delta}$  {Distance join...}
     $P' \leftarrow$  Set of points at timestamp  $t_j$  involved in  $F'$ 
    for each timestamp  $t_j$  between  $t_i$  and  $t_{i+\delta}$  do
       $F \leftarrow \emptyset$ 
      for each candidate flock  $f$  in  $F'$  do
         $P'_j \leftarrow$  Set of points at timestamp  $t_j$  from  $P'$  involved in  $f$ 
         $d \leftarrow$  get maximum distance between points in  $P'_j$ 
        if  $d < \epsilon$  then
          add  $f$  to  $F$ 
        else
           $f' \leftarrow \text{getMaximalDisks}(P'_j, \epsilon, \mu)$ 
          add  $f'$  to  $F$ 
        end if
      end for
       $F' \leftarrow F$ 
    end for
  end if
end for
return  $F$ 
```

Algorithm 2 getMaximalDisks(T, ϵ, μ) - Finding maximal disks following a parallel approach.

Input: Set of points T , maximum distance ϵ and minimum size μ

Output: Set of maximal disks M

```

find the set of pairs of points  $P$  in  $T$  which are  $\epsilon$  distance each other
 $C \leftarrow \emptyset$ 
for each  $p_i$  in  $P$  do
    compute disks  $c_i^1$  and  $c_i^2$  of  $p_i$  using  $\epsilon$ 
    add  $c_i^1$  and  $c_i^2$  to  $C$ 
end for
 $D \leftarrow \emptyset$ 
for each  $c_i$  in  $C$  do
    find the set of points  $\rho_i$  which lie  $\epsilon$  distance around  $c_i$ 
    if  $|\rho_i| \geq \mu$  then
        compute centroid  $\varsigma_i$  of the MBR of  $\rho_i$ 
        set  $d_i.center$  as  $\varsigma_i$ 
        set  $d_i.points$  as  $\rho_i$ 
        if  $d_i$  not in  $D$  then
            add  $d_i$  to  $D$  {Pruning duplicate candidates...}
        end if
    end if
end for
build an R-Tree  $disksRT$  using centers in  $D$ 
 $E \leftarrow \emptyset$ 
for each MBR in  $disksRT$  do
    expand MBR to create an expanded MBR  $\varepsilon_i$  using a buffer of  $\epsilon$  distance
    add  $\varepsilon_i$  to  $E$ 
end for
for each  $d_i$  in  $D$  do
    for each  $\varepsilon_j$  in  $E$  do
        if  $d_i.center \cap \varepsilon_j$  then
            add  $d_i$  to  $\varepsilon_j$ 
        end if
    end for
end for
 $M \leftarrow \emptyset$ 
for each  $\varepsilon_i$  in  $E$  do
     $\chi \leftarrow \emptyset$ 
    for each  $d_i$  in  $\varepsilon_i$  do
        add  $d_i.points$  to  $\chi$ 
    end for
    find the set of maximal patterns  $F$  in  $\chi$ 
    for each  $f_i$  in  $F$  do
        compute centroid  $\varsigma_i$  of the items in  $f_i$ 
        if  $\varsigma_i$  is not in the expansion area of  $\varepsilon_i$  then
            set  $m_i.center$  as  $\varsigma_i$ 
            set  $m_i.points$  as  $f_i$ 
            add  $m_i$  to  $M$ 
        end if
    end for
end for
return  $M$ 

```
