

Stay Point Detection in un museo

Jaspreet Kaur

Università degli Studi di Milano

Email: jaspreet.kaur@studenti.unimi.it

Numero di matricola: 980691

Marco Incerti

Università degli Studi di Milano

Email: marco.incerti@studenti.unimi.it

Numero di matricola: 989428

1. Introduzione

Uno stay point identifica una regione geografica dove l'utente è rimasto per un certo periodo di tempo. Da questa definizione, si può dedurre che uno stay point può essere caratterizzato in due modi diversi:

- uno stay point è un punto dove l'utente rimane fermo per un certo periodo di tempo, che supera una threshold fissata.
- uno stay point è il centroide di un insieme di punti, che identificano delle posizioni dove l'utente girovaga per un certo periodo di tempo, che supera una threshold fissata.

L'algoritmo di Stay Point Detection permette di individuare automaticamente le regioni spaziali dove l'utente ha passato un periodo di tempo che supera una certa threshold. Il riconoscimento di stay point permette di aggiungere semantica ai dati delle traiettorie e aumentare la nostra conoscenza sui comportamenti degli utenti.

In questo progetto, l'algoritmo di Stay Point Detection viene applicato sulle traiettorie dei visitatori di un museo. L'informazione ottenuta consente di capire quali sono i percorsi seguiti da un visitatore e quali sono le esposizioni che hanno suscitato più interesse. Questa conoscenza è fondamentale per strutturare le esposizioni all'interno di un museo con l'obiettivo di aumentare il coinvolgimento dei visitatori e migliorare la loro esperienza.

2. Dataset

L'analisi è stata svolta su un dataset che contiene informazioni riguardo a un museo e ad alcuni suoi visitatori. In particolare, i dati sono formati da:

- traiettorie di 3 visitatori: contengono diversi punti, ognuno dei quali identifica la posizione di un visitatore in un dato momento, specificato da un timestamp. Tutti i dati si concentrano tra le 12:08 e le 12:20 del 13/12/2021. In particolare, si hanno a disposizione 7388 punti per la persona con identificatore 57, 7008 punti per la persona con identificatore 67 e 7211 punti per la persona con identificatore 68. Tali dati sono stati raccolti con

una frequenza di campionamento molto alta grazie a un sistema di localizzazione basato sulla tecnologia UWB.

- tavoli del museo: contengono gli identificatori e le posizioni dei tavoli collocati nel museo. In particolare, si considerano 7 tavoli.
- esposizioni sui tavoli: contengono gli identificatori e le posizioni delle esposizioni collocate sui tavoli del museo. In particolare, si considerano 35 esposizioni.

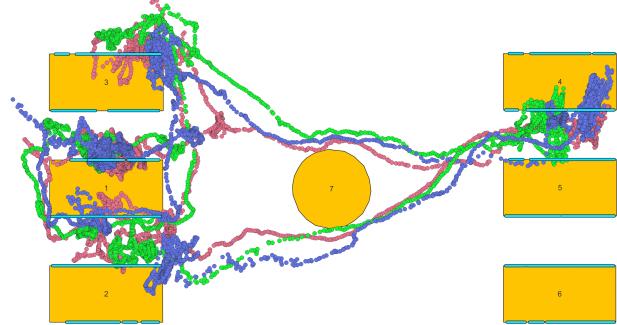


Figura 1: Dataset Iniziale

Nella figura 1 si possono osservare i dati di partenza. I tavoli sono rappresentati in colore arancio ed etichettati con i rispettivi identificatori. Le esposizioni sui tavoli sono collocate sui bordi dei tavoli e rappresentate in azzurro neon. Le traiettorie delle varie persone sono rappresentate con colori diversi.

3. Tecnologie e strumenti usati

Il progetto è formato da uno script in linguaggio Python, che include query in linguaggio SQL. Tali query vengono eseguite su un database PostGIS, che mantiene i dati e i risultati intermedi e finali. I risultati vengono rappresentati usando QGIS, collegato allo stesso database PostGIS.

4. Stay Point Detection

4.1. Pre-processing

Focalizziamo soltanto sulla traiettoria di un visitatore. La figura 2 rappresenta la traiettoria della persona con identificatore 57.

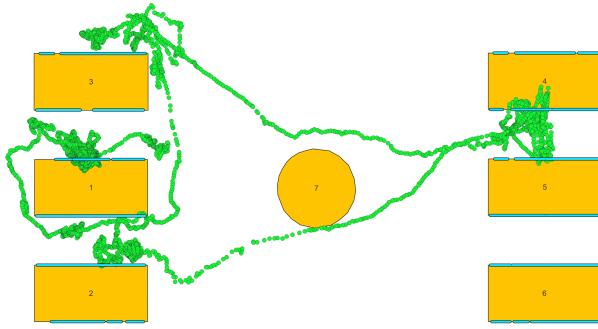


Figura 2: Traiettoria visitatore 57

Si può notare come i dati siano soggetti a rumore: infatti, il 19.8% dei punti della traiettoria sono collocati al di sopra dei tavoli, chiaramente irraggiungibili dal visitatore.

A causa della presenza di dati rumorosi, è stato deciso di applicare delle operazioni di pre-processing.

L'idea su cui si basano tali operazioni è la seguente: il sistema di localizzazione ha erroneamente identificato come posizioni dell'utente dei punti sopra i tavoli; tuttavia, è molto probabile che il sistema non si sia sbagliato di molto, quindi nei momenti in cui la posizione del visitatore viene indicata sopra un tavolo, probabilmente l'utente si trova vicino al tavolo, ma fuori da esso. Seguendo questo ragionamento, tutti i punti rumorosi sono stati trasformati nei punti più vicini ad essi che si trovano sul bordo del tavolo.

La fase di pre-processing appena illustrata è stata implementata attraverso delle query SQL. Supponendo di concentrarci solo sulla traiettoria della persona con identificatore 57, i passaggi sono stati i seguenti:

- 1) Creare una nuova tabella, contenente gli stessi dati della traiettoria della persona con identificatore 57.

```
CREATE TABLE IF NOT EXISTS
    person_57_refined as
        (SELECT * FROM person_57)
```

- 2) Creare una nuova tabella che contiene solamente i punti della traiettoria che intersecano un qualsiasi tavolo.

```
CREATE TABLE IF NOT EXISTS
    intersection_tables_57 as
        (SELECT DISTINCT p1.id, p1.
            geom as geom
        FROM person_57 as p1
        WHERE p1.geom in
            (SELECT DISTINCT
                st_intersection(p1.geom,
                    t.geom) as geom
            FROM person_57 as p1,
            tables as t))
```

- 3) Creare una nuova tabella che associa ad ogni punto della tabella precedente il punto più vicino ad esso sul bordo del tavolo che interseca.

```
CREATE TABLE IF NOT EXISTS
    closest_points_57 as
        (SELECT DISTINCT p.id,
            st_closestpoint(st_boundary
                (t.geom), p.geom) as geom
        FROM intersection_tables_57 as
            p, tables as t
        WHERE st_intersects(p.geom, t.
            geom))
```

- 4) Aggiornare la tabella creata al punto 1, sostituendo le geometrie dei punti nella tabella creata al punto 3 con i rispettivi punti più vicini sul bordo del tavolo.

```
UPDATE person_57_refined
SET geom = c.geom
FROM closest_points_57 as c
WHERE c.id = person_57_refined.id
```

Dopo aver applicato tali operazioni, la traiettoria della persona con identificatore 57 viene modificata come rappresentato nella figura 3 in viola. Si può notare come tutti i punti precedentemente collocati sui tavoli, visibili in figura in verde, siano stati spostati sul bordo del tavolo che intersecavano.

Queste operazioni sono state applicate anche sulle traiettorie delle persone con identificatore 67, dove i dati rumorosi erano il 19.5%, e con identificatore 68, dove i dati rumorosi erano il 35.3%. La successiva analisi è stata svolta sui dati ottenuti dalla fase di pre-processing appena illustrata.

4.2. Algoritmo

L'algoritmo di Stay Point Detection può essere descritto dallo pseudocodice in Algorithm 1.

Nella nostra analisi, se un individuo passa più di 30 secondi in una distanza di 1 metro, la regione rilevata viene considerata uno stay point. La scelta della threshold temporale è stata dettata dal dataset: infatti, le traiettorie si riferiscono ai movimenti dei visitatori in un lasso temporale di soli 10 minuti, quindi, scegliere una threshold temporale più ampia avrebbe tralasciato molti possibili stay point; d'altro canto, una threhsold temporale più breve non

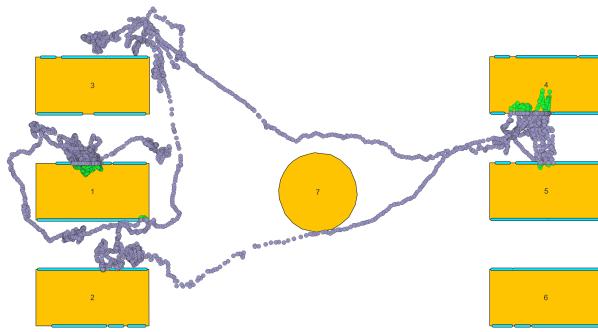


Figura 3: Traiettoria visitatore 57 refined

Algorithm 1 Algoritmo di Stay Point Detection

```

Input: un log GPS  $P$ , una threshold di distanza  $distThresh$ ,  
una threshold di tempo  $timeThresh$   

Output: un insieme di punti  $SP = \{S\}$ 
 $i \leftarrow 0$ 
 $pointNum \leftarrow |P|$  //numero di punti GPS
while  $i < pointNum$  do
     $j \leftarrow i + 1$ 
     $token \leftarrow 0$ 
    while  $j < pointNum$  do
         $d \leftarrow distance(pi, pj)$ 
        if  $d > distThresh$  then
             $\Delta T \leftarrow pj.T - pi.T$ 
            if  $\Delta T > timeThresh$  then
                 $S.coord \leftarrow computeMean -$ 
                 $Coord(pk|i \leq k \leq j)$ 
                 $S.arrT \leftarrow pi.T$ 
                 $S.levT \leftarrow pj.T$ 
                 $SP.insert(S)$ 
                 $i \leftarrow j$ 
                 $token \leftarrow 1$ 
            end if
            break
        end if
         $j \leftarrow j + 1$ 
    end while
    if  $token \neq 1$  then
         $i \leftarrow i + 1$ 
    end if
end while
return  $SP$ 

```

sarebbe significativa, considerando che i visitatori si stanno spostando a piedi. La scelta della threshold spaziale è stata dettata dall'ambiente museale: il boundary delle geometrie che rappresentano i tavoli ha una lunghezza pari a 12 metri, quindi i tavoli hanno probabilmente 4 metri di lunghezza e 2 metri di larghezza. Su ogni tavolo, sono presenti più esposizioni, quindi l'utente può fermarsi in più punti significativi

del tavolo. Una threshold spaziale più grande di 1 metro non permetterebbe di cogliere le informazioni riguardo alle esposizioni a cui l'utente ha mostrato interesse. Inoltre, la distanza tra un tavolo e l'altro è di 1.74 metri, quindi una threshold di distanza più ampia potrebbe unire posizioni in cui l'utente sta guardando le esposizioni di due tavoli vicini.

La nostra implementazione può essere descritta dai seguenti punti:

- 1) pre-processing del dataset: i dati memorizzati su PostGIS vengono pre-processati come descritto nella sezione 4.1.
- 2) acquisizione del dataset: i dati pre-processati vengono recuperati da PostGIS.
- 3) applicazione algoritmo: i dati recuperati vengono analizzati con l'algoritmo di Stay Point Detection.
- 4) salvataggio dei risultati: i risultati ottenuti vengono memorizzati su PostGIS.

4.3. Risultati

Nella figura 4 sono rappresentati in rosso gli stay point del visitatore con identificatore 57. In particolare, sono stati trovati 11 stay point.

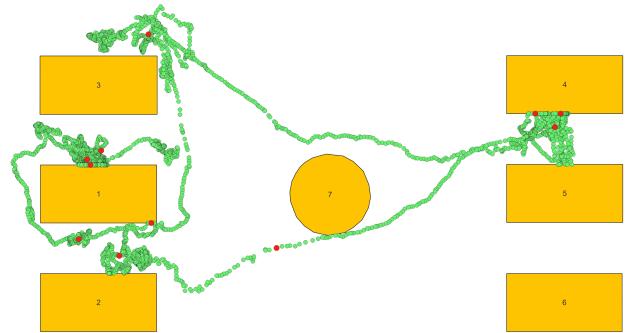


Figura 4: Stay point del visitatore 57

Nella figura 5 sono rappresentati in rosso gli stay point del visitatore con identificatore 67. In particolare, sono stati trovati 13 stay point.

Nella figura 6 sono rappresentati in rosso gli stay point del visitatore con identificatore 68. In particolare, sono stati trovati 12 stay point.

4.4. Ulteriori esperimenti

I risultati presentati nella sezione 4.3 sono stati ottenuti dall'esecuzione dell'algoritmo di Stay Point Detection, preceduta da una fase di pre-processing dei dati con l'obiettivo di eliminare le posizioni rumorose. Per dimostrare la necessità della fase di pre-processing, si è applicato l'algoritmo direttamente sui dati originali, caratterizzati da rumore. Si è mantenuta la threshold spaziale di 1 metro e la threshold

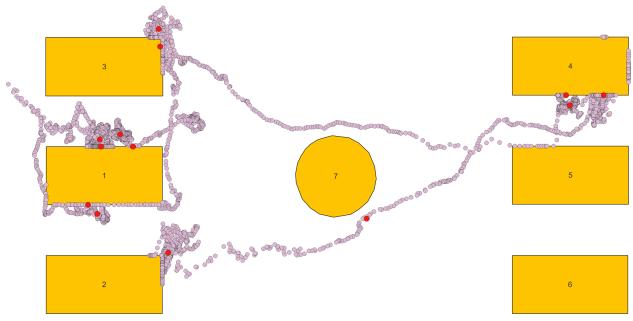


Figura 5: Stay point del visitatore 67

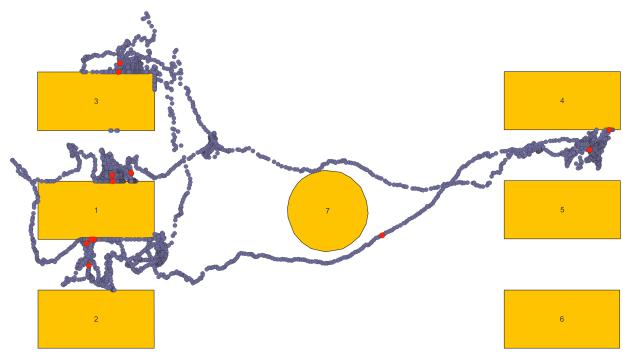


Figura 6: Stay point del visitatore 68

temporale di 30 secondi. Nella figura 7 sono rappresentati gli stay point del visitatore con identificatore 57, se l'algoritmo viene applicato dopo la fase di pre-processing (in verde) e se viene applicato senza la fase di pre-processing (in rosso). In particolare, sono stati trovati 7 stay point nel secondo caso.

Nella figura 8 sono rappresentati gli stay point del visitatore con identificatore 67, se l'algoritmo viene applicato dopo la fase di pre-processing (in verde) e se viene applicato senza la fase di pre-processing (in rosso). In particolare, sono stati trovati 7 stay point nel secondo caso.

Nella figura 9 sono rappresentati gli stay point del visitatore con identificatore 68, se l'algoritmo viene applicato dopo la fase di pre-processing (in verde) e se viene applicato senza la fase di pre-processing (in rosso). In particolare, sono stati trovati 12 stay point nel secondo caso.

Si può notare come in alcuni casi gli stay point identificati senza la fase di pre-processing e gli stay point identificati con la fase di pre-processing coincidano. Negli altri casi, invece, i due tipi di stay point sono molto simili. Questo vuol dire che la fase di pre-processing non ha corrotto i dati originali, ma ha mantenuto le informazioni in essi contenute.

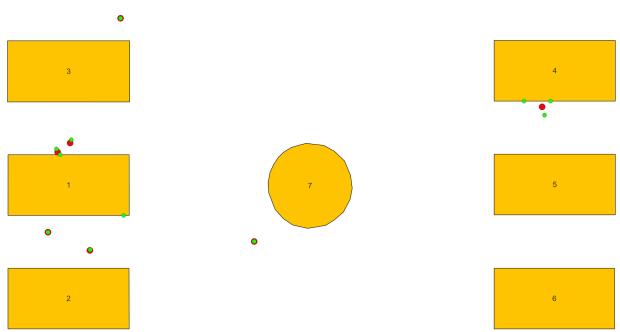


Figura 7: Stay point del visitatore 57 con e senza pre-processing

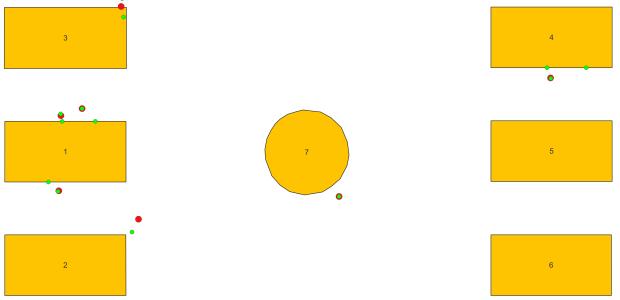


Figura 8: Stay point del visitatore 67 con e senza pre-processing

Il preprocessing permette inoltre di dare più credibilità alla soluzione ottenuta, perché i dati su cui è stato applicato l'algoritmo sono più veritieri.

5. Identificazione delle esposizioni visitate

Partendo dai risultati ottenuti dall'algoritmo di Stay Point Detection, rappresentati graficamente nella sezione 4.3, si può calcolare quali esposizioni hanno attirato l'attenzione di ogni visitatore. In particolare, se lo stay point di un visitatore si trova a una distanza minore di 1 metro da una certa esposizione, l'utente ha visitato l'esposizione in questione. Quest'informazione è stata ricavata attraverso una query SQL. Di seguito, viene riportata la query per il visitatore con identificatore 57. Per i rimanenti visitatori, viene applicata la query analoga.

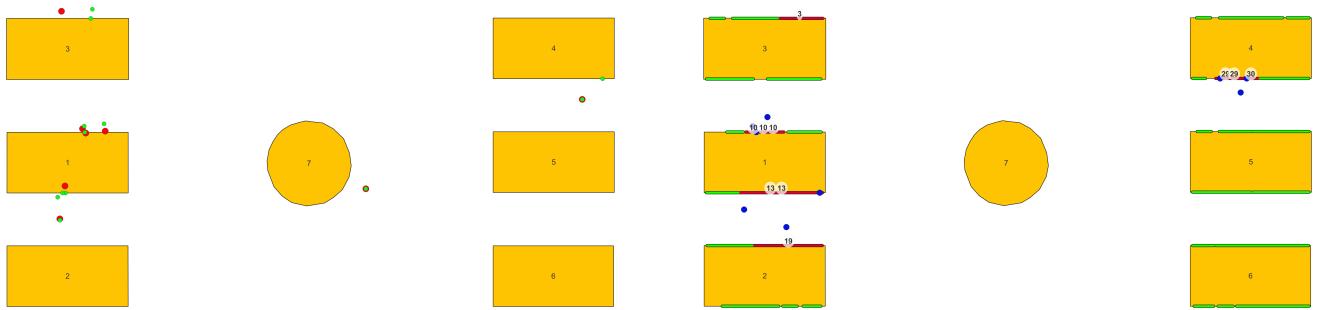


Figura 9: Stay point del visitatore 68 con e senza pre-processing

Figura 10: Esposizioni visitate dal visitatore 57

```

SELECT p.id, p.geom AS p, p.arrivetime,
       p.leavetime, e.geom AS geom
FROM exhibits_on_tables AS e,
     (
      SELECT p.geom, MIN(ST_DISTANCE(p.
          geom, e.geom)) AS minDist
      FROM stay_points_57 AS p,
           exhibits_on_tables AS e
      GROUP BY p.geom) AS i
JOIN stay_points_57 AS p
ON p.geom = i.geom
WHERE ST_DISTANCE(p.geom, e.geom) = i.
minDist AND ST_DISTANCE(p.geom, e.
geom) < 1
    
```

visitatore	esposizioni
visitatore 57	3, 10, 13, 19, 29, 30
visitatore 67	3, 10, 11, 13, 19, 30, 31
visitatore 68	3, 10, 11, 13, 19, 31

Tabella 1: Esposizioni visitate dai visitatori

La tabella 1 riassume i risultati, specificando per ogni visitatore le esposizioni visitate, mentre le figure 10, 11, 12 rappresentano mostrando tali risultati.

Dai risultati della query SQL precedente applicata ai tre visitatori si può notare che più stay point dello stesso visitatore riguardano le stesse esposizioni. Per risolvere queste ripetizioni, si è deciso di considerare per ogni esposizione visitata il centroide degli stay point che vi fanno riferimento. L'intervallo di tempo di tale centroide è dato dal tempo di arrivo più remoto tra i tempi di arrivo degli stay point considerati e dal tempo di partenza più recente tra i loro tempi di partenza. Tale calcolo è stato svolto attraverso la seguente query SQL.

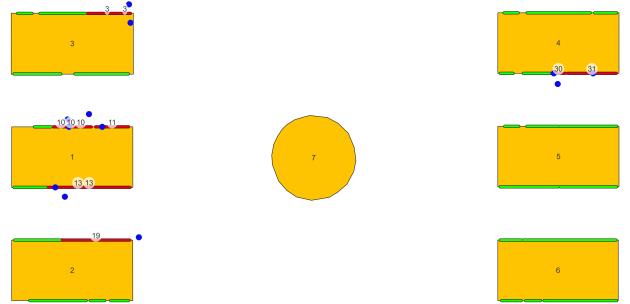


Figura 10: Esposizioni visitate dal visitatore 57

```

SELECT DISTINCT p.exhibitid, c.
centroid AS geom, c.arrivalTime, c
.leaveTime
FROM
  (
    SELECT exhibitid, st_centroid(
        st_collect(point)) AS centroid
        , min(arrivetime) AS
arrivalTime, max(leavetime) AS
leaveTime
    FROM exhibits_person_57
    GROUP BY exhibitid) AS c
JOIN exhibits_person_57 AS p
ON p.exhibitid = c.exhibitid)
    
```

Tale query è stata ripetuta per i visitatori con identificatore 67 e 68. Le figure 13, 14, 15 mostrano i centroidi degli stay point (in rosso) e gli stay point (in viola).

Da questi dati si possono calcolare se vi sono individui che hanno osservato delle esposizioni nello stesso momento. In particolare, se l'intervallo dello stay point più vicino o l'intervallo del centroide degli stay point più vicini a un'esposizione si sovrappone all'intervallo dello stay point

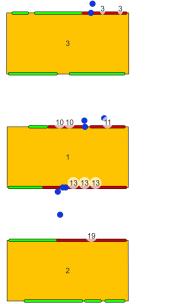


Figura 12: Esposizioni visitate dal visitatore 68

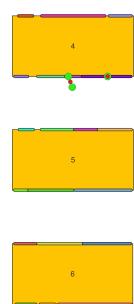
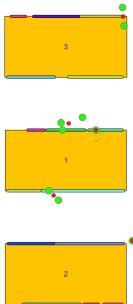
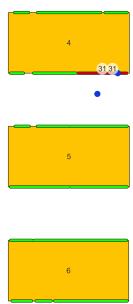


Figura 14: Centroidi degli stay point del visitatore 67

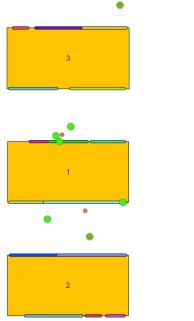


Figura 13: Centroidi degli stay point del visitatore 57

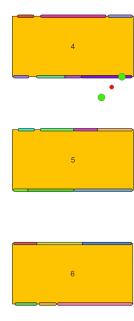
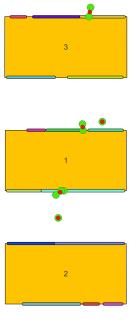
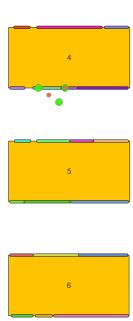


Figura 15: Centroidi degli stay point del visitatore 68

più vicino o all'intervallo del centroide degli stay point più vicini alla stessa esposizione di un altro visitatore, allora i due visitatori hanno osservato l'esposizione in questione contemporaneamente. Quest'informazione è stata ricavata attraverso una query SQL. Di seguito, viene riportata la query per i visitatori con identificatore 57 e con identificatore 67. Per le rimanenti coppie di visitatori, viene applicata la query analoga.

```
SELECT p1.exhibitid, p2.exhibitid
FROM centroids_exhibits_person_67 as p1,
      centroids_exhibits_person_68 as p2
WHERE (p2.leaveTime, p2.arrivalTime)
      OVERLAPS (p1.leaveTime, p1.
                  arrivalTime)
      AND p1.exhibitid = p2.exhibitid
```

La tabella 2 riassume i risultati, specificando per ogni esposizione visitata da almeno un visitatore, quali sono i visitatori e quant'è il numero di visitatori che l'hanno visitata nello stesso momento.

6. Conclusioni

In questo paper abbiamo presentato la nostra soluzione al progetto del corso di Geospatial Data Management. Il lavoro svolto ha permesso di identificare gli stay point di tre visitatori partendo dalle loro traiettorie in un museo e di svolgere analisi sulla loro visita. In particolare, sono state identificate le esposizioni visitate da ognuno di essi, le esposizioni visitate contemporaneamente da almeno due visitatori e, infine, quelle più visitate. Tale analisi può aiutare a studiare il comportamento dei visitatori di un museo e, quindi, può permettere di progettare al meglio lo spazio museale, dando più enfasi alle esposizioni più interessanti e arricchendo e migliorando le esposizioni meno visitate. Alcuni aspetti della soluzione proposta possono essere migliorati. In particolare, la riduzione del rumore dei dati originali può essere svolta in maniera differente. Infatti, non risponendo di una ground truth sui dati raccolti, il pre-processing è stato applicato solamente ai dati evidentemente rumorosi. Tuttavia, probabilmente anche gli altri dati delle traiettorie presentano del rumore. Per questo motivo, si possono utilizzare filtri e tecniche di riduzione del rumore

esposizione	visitatori	n visitatori
esposizione 3	57, 67, 68	3
esposizione 10	57, 67, 68	3
esposizione 11	67, 68	2
esposizione 13	57, 67, 68	3
esposizione 19	57, 67, 68	3
esposizione 29	57	1
esposizione 30	57, 67	2
esposizione 31	67, 68	2

Tabella 2: Esposizioni visitate nello stesso momento

differenti, che possano essere applicate su tutti i dati delle traiettorie e non solo sui dati chiaramente sbagliati. Inoltre, la ricerca degli stay point può essere condotta utilizzando algoritmi differenti, come ad esempio l'algoritmo SeqScan, che è in grado di gestire dati rumorosi e movimenti più complessi. Infine, è possibile raccogliere ulteriori dati che possano aiutare nell'identificazione delle esposizioni visitate. Nella nostra soluzione, infatti, sono state considerate le esposizioni più vicine agli stay point trovati. Tuttavia, altre informazioni riguardo l'ambiente, le traiettorie e il comportamento dei visitatori possono permettere di applicare ragionamenti più sofisticati e completi. Nonostante ciò, la soluzione proposta afferma che è possibile studiare il comportamento dei visitatori di un museo analizzando i loro stay point, ottenendo in questo modo informazioni preziose che possono essere usate per migliorare l'esperienza degli utenti e per lasciare un'impronta più significativa su di essi.

Riferimenti bibliografici

- [1] QUANNAN LI, YU ZHENG, XING XIE, YUKUN CHEN, WENYU LIU, WEI-YING MA, 2008, *Mining User Similarity Based on Location History in GIS '08: Proceedings of the 16th ACM SIGSPATIAL international conference on Advances in geographic information systems*, Association for Computing Machinery, New York, NY, United States, Article No.: 34, 1–10.
- [2] MADGY NEHAL, SAKR A. MAHMOUD, MOSTAFA TAMER, EL-BAHNASY KHALED, December 2018, *Review on trajectory similarity measures in Conference: 2015 IEEE Seventh International Conference on Intelligent Computing and Information Systems (ICICIS)*, Faculty of Commuter & Information Sciences, Ain Shams University, Cairo, Egypt, 613-619