

Agrupamiento de trayectorias vía clustering espectral incremental

Luciano Lorenti, Javier Giacomantone, Armando De Giusti

Instituto de Investigación en Informática (III-LIDI),
Facultad de Informática - Universidad Nacional de La Plata - Argentina.
La Plata, Buenos Aires, Argentina.
{llorenti,jog,degiusti}@lidi.info.unlp.edu.ar

Resumen. El agrupamiento de trayectorias permite entender los patrones de movimiento de los objetos capturados en vídeo. Debido a su naturaleza secuencial, los datos de trayectorias son adquiridos cuadro a cuadro de forma incremental. Este procedimiento implica actualizar las estructuras detectadas por el algoritmo de agrupamiento ante la llegada de nuevos datos. En este artículo se presenta un método de agrupamiento espectral incremental, que permite extraer las trayectorias representativas de los patrones de movimiento. El método propuesto explota un mecanismo orientado a reducir la demanda computacional en el cálculo de autovectores en cada iteración. Se presentan resultados experimentales sobre trayectorias de vídeos reales y trayectorias sintéticas, que permiten evaluar el método propuesto.

Palabras clave: Aprendizaje no supervisado, Clustering espectral incremental, Agrupamiento de trayectorias

1. Introducción

Las trayectorias de los objetos capturados en video son elementos importantes para el análisis de patrones de movimiento [1]. Las técnicas de agrupamiento, *clustering*, son un componente clave para el análisis de trayectorias. Su utilización resulta conveniente cuando en lugar de modelar y analizar el movimiento de un objeto individual, se procesan múltiples trayectorias para detectar las estructuras subyacentes de las actividades en video [2]. Las técnicas de clustering de trayectorias proveen beneficios para muchas tareas de visión automática como segmentación de movimientos [3], detección de objetos [4], reconocimiento de acciones [5] y modelado de escenas [6].

Una propiedad usual de las aplicaciones de seguimiento de objetos es que nuevas trayectorias son incorporados a la escena en cada instante de tiempo [7]. Frente a estos datos continuos, la información relacionada con las trayectorias debe ser procesada en forma incremental. Un punto importante a notar es que los agrupamientos que se encuentren distantes a los nuevos datos no se verán influidos. Por lo tanto, un enfoque que permite manejar datos que varían en

el tiempo consiste en ajustar los agrupamientos que se encuentren cercanos en algún sentido a los datos agregados o removidos para un instante de tiempo dado.

Por lo general las trayectorias se encuentran en un subespacio de baja dimensionalidad [8]. Es por esto que, usualmente, las trayectorias primero se representan en un espacio de baja dimensión antes de realizar el agrupamiento. Las técnicas de agrupamiento espectral obtienen una representación de los datos en un espacio de dimensión inferior obteniendo la descomposición en autovectores de la matriz laplaciana asociada al grafo de semejanza de los datos. Un enfoque posible para el problema del agrupamiento espectral incremental consiste en calcular los autovectores de la matriz laplaciana del grafo de proximidad en cada instante de tiempo, sin embargo, esto es computacionalmente costoso dado que resolver el sistema de autovalores es de orden cúbico con respecto a la cantidad de elementos en el grafo. Un modo de mejorar la eficiencia de los métodos de agrupamiento espectral es calcular una aproximación de la descomposición de autovectores en cada iteración utilizando el método de Nystrom [9] en el que se calculan las semejanzas entre un subconjunto de patrones y se aproximan las semejanzas restantes. Se ha abordado, recientemente, el problema de actualizar eficientemente el sistema de autovectores ante una modificación del grafo de semejanza [10] [11] [12].

En este trabajo se presenta un método de agrupamiento espectral incremental de trayectorias en el que en cada instante de tiempo se actualiza el grafo de semejanza entre los segmentos de trayectorias. Se aproximan los autovectores de la matriz laplaciana del grafo de semejanza utilizando lo calculado en el instante de tiempo anterior, se realiza un agrupamiento de los segmentos de trayectorias y, por último, se obtiene una trayectoria representativa para cada grupo. La evaluación del método propuesto se realizó utilizando trayectorias reales obtenidas en vídeos de vigilancia y trayectorias simuladas.

El artículo está organizado del siguiente modo, en la sección 2 se describe el problema del agrupamiento de trayectorias. En la sección 3 se presenta el mecanismo de agrupamiento espectral incremental. En la sección 4 se expone el método propuesto. En la sección 5 se muestran los resultados experimentales obtenidos. Finalmente, en la sección 6 se presentan las conclusiones.

2. Agrupamiento de trayectorias

En el contexto de este trabajo una trayectoria consiste en una secuencia de puntos. Cada punto p_i tiene asociada una información espacial $pos_{p_i} \in \mathbb{R}^n$ y una información temporal $t_{p_i} \in \mathbb{R}$. Un segmento de trayectoria es un segmento $s = p_i p_j (i < j)$ donde p_i y p_j son puntos elegidos de la misma trayectoria. Cada segmento de trayectoria tiene asociado, a su vez, información temporal: $t_s = t_{p_j}$. Se define $B(s)$ como el rectángulo delimitador mínimo de s . Un cluster es un conjunto de segmentos de trayectorias. Los segmentos de trayectorias que pertenecen al mismo cluster son similares de acuerdo a una función de semejanza. Segmentos distintos de una misma trayectoria puede pertenecer a agrupamientos

diferentes, dado que el agrupamiento se realiza sobre los segmentos y no sobre las trayectorias. Dado un conjunto de trayectorias $I = \{TR_1, \dots, TR_n\}$ obtenidas en un instante de tiempo t , la salida del método propuesto para ese instante consiste en un agrupamiento de los segmentos de las trayectorias $TrClu = \{C_1, \dots, C_k\}$ y una trayectoria representativa $\forall C_i \in TrClu$. Una trayectoria representativa de un cluster es una secuencia de puntos que indica el comportamiento mayoritario de los segmentos de línea pertenecientes al mismo.

2.1. Semejanza entre segmentos

La función de distancia entre segmentos de trayectorias propuesta por [13] está compuesta por tres componentes: la distancia perpendicular, la distancia paralela y la distancia angular. La figura 1 muestra las tres componentes de la distancia. Sean $L_i = s_i e_i$ y $L_j = s_j e_j$ segmentos de línea d -dimensionales. Donde $s_i, e_i, s_j, y e_j$ son las posiciones en el espacio de los puntos iniciales y finales de cada segmento. Asignemos, sin perder generalidad, al segmento más largo el nombre L_i y al segmento más corto, el nombre L_j .

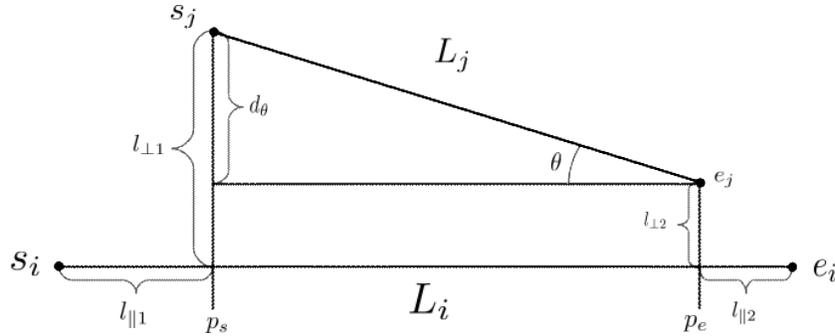


Figura 1: Componentes de la función de distancia para los segmentos de línea

La distancia perpendicular entre L_i y L_j se define como:

$$d_{\perp}(L_i, L_j) = \frac{l_{\perp 1}^2 + l_{\perp 2}^2}{l_{\perp 1} + l_{\perp 2}}$$

donde $l_{\perp 1}$ es la distancia euclídea entre el punto s_j y el punto p_s . El punto p_s es la proyección de s_j sobre L_i . La distancia $l_{\perp 2}$ se define de la misma manera para e_j y p_e . La distancia paralela entre L_i y L_j se define como:

$$d_{\parallel}(L_i, L_j) = \min(l_{\parallel 1}, l_{\parallel 2})$$

Supongamos que la proyección de los puntos s_j y e_j en L_i son p_s y p_e respectivamente. Se define $l_{\parallel 1}$ como el mínimo de la distancia euclídea de p_s a s_i y e_i . Del mismo modo, $l_{\parallel 2}$ es la distancia euclídea mínima de p_e a s_i y e_i . La distancia angular entre L_i y L_j se define como:

$$d_{\theta}(L_i, L_j) = \begin{cases} \|L_j\| \sin(\theta) & \text{si } 0^{\circ} \leq \theta \leq 90^{\circ} \\ \|L_j\| & \text{si } 90^{\circ} \leq \theta \leq 180^{\circ} \end{cases}$$

donde $\|L_j\|$ es la longitud de L_j y α el ángulo de intersección mas pequeño entre L_i y L_j . Finalmente, se define la distancia entre dos segmentos de trayectoria de la siguiente manera:

$$dist(L_i, L_j) = d_{\perp}(L_i, L_j) + d_{\theta}(L_i, L_j) + d_{\parallel}(L_i, L_j) \quad (1)$$

2.2. Trayectoria representativa

La trayectoria representativa de un cluster describe el movimiento general de los segmentos de trayectorias que pertenecen al mismo. En el método propuesto en [13], ilustrado en la figura 2, los puntos que componen la trayectoria representativa son extraídos analizando los puntos finales e iniciales de cada segmento de trayectoria. La idea central del algoritmo consiste en hacer pasar una línea vertical a lo largo de los segmentos de trayectorias en la dirección del eje mayor del cluster. Se contabiliza el número de segmentos que son atravesados por la línea de barrido. Este número solo cambia cuando la línea de barrido pasa por un punto de inicio o por un punto final de un segmento. Si este número es mayor o igual a un parámetro definido por el usuario, denominado *MinLns*, se calcula la coordenada promedio de los segmentos de línea con respecto al eje mayor y se inserta el punto promedio en la trayectoria representativa, caso contrario se saltea el punto que se está procesando. Además, con el objetivo de suavizar la trayectoria generada, si el punto que se está procesando se encuentra demasiado cerca de un punto ya insertado en la trayectoria representativa, ese punto no es tenido en cuenta.

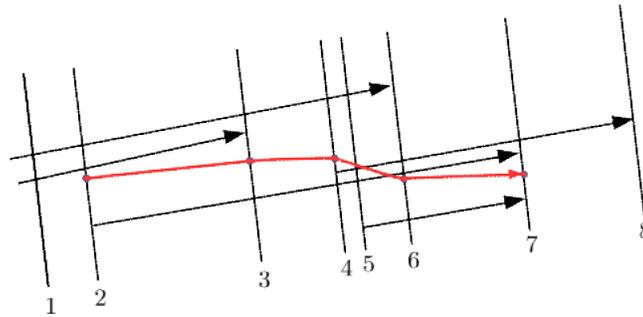


Figura 2: Trayectoria representativa de un agrupamiento. $MinLns = 3$

3. Clustering espectral incremental

Dado un conjunto de patrones $X = \{x_1, x_2, \dots, x_n\} \in \mathbb{R}^m$ y una función de semejanza $d(x_i, x_j) = \exp\left(\frac{-d^2(x_i, x_j)}{\alpha^2}\right)$, es posible construir un grafo de semejanza $G = (V, E)$ tal que $V = X$ y los pesos de las aristas estén dados por $d(x_i, x_j)$. Los algoritmos de agrupamiento espectral obtienen una representación de los datos en un espacio de dimensión inferior en la que resulta sencillo aplicar un algoritmo de clustering tradicional [14] [15]. Este espacio se obtiene calculando la matriz de afinidad W del grafo G y resolviendo el siguiente problema de optimización:

$$\begin{aligned} \max_{U \in \mathbb{R}^{n \times m}} \quad & Tr(U^T L U) \\ \text{s.t.} \quad & U^T U = I \end{aligned} \quad (2)$$

donde $L = D^{-\frac{1}{2}} W D^{-\frac{1}{2}}$ es la matriz laplaciana de W y D es una matriz diagonal con la suma de las filas de W ubicadas en su diagonal principal. Una vez obtenido U sus filas son consideradas como las nuevas coordenadas de los patrones. En este nuevo espacio los pares de puntos semejantes se encuentran cercanos. Esto se debe a que los autovectores de la matriz laplaciana minimizan la siguiente función objetivo [16]:

$$\sum_{i,j} W_{ij} \|u^{(i)} - u^{(j)}\|^2$$

donde $u^{(i)} = [u_1(i), \dots, u_m(i)]$ es la representación m -dimensional del i -ésimo patrón.

3.1. Escalado local del peso de las aristas

El parámetro de escalado α de la función de semejanza es una medida que determina cuando dos puntos son considerados semejantes. La selección del parámetro se realiza, por lo general, de forma manual. La presencia de un escalado fijo para todo el conjunto de datos ocasiona problemas cuando se encuentran agrupamientos de distintas densidades. Los métodos de clustering espectral pueden no encontrar un buen agrupamiento cuando los datos contienen múltiples escalas, incluso utilizando el α óptimo. En lugar de seleccionar un sólo parámetro de escala, en [17] se propuso un método para calcular la escala local de α_i para cada punto s_i . La distancia de s_i a s_j vista por s_i es $\frac{d(s_i, s_j)}{\alpha_i}$ mientras que la distancia s_i a s_j vista por s_j es $\frac{d(s_i, s_j)}{\alpha_j}$. Por lo tanto la semejanza entre un par de puntos puede ser escrita como:

$$W_{ij} = \exp\left(-\frac{d(s_i, s_j)^2}{\alpha_i \alpha_j}\right)$$

Un parámetro de escalado específico para cada punto permite configurar la escala de acuerdo a la densidad encontrada en la vecindad del punto i y del punto j . La selección de la escala local α_i puede ser establecida estudiando los estadísticos locales de la vecindad del punto s_i .

3.2. Número de Clusters

Una herramienta para determinar el número de agrupamientos utilizando las técnicas de clustering espectral consiste en encontrar una caída drástica en la magnitud de los autovalores. Sea $\lambda_1, \lambda_2, \dots, \lambda_m$ los autovalores ordenados de menor a mayor de la matriz laplaciana. El objetivo es elegir un número k tal que los valores de $\lambda_1, \lambda_2, \dots, \lambda_k$ sean pequeños y λ_{k+1} sea relativamente grande [18]. Se define, por lo tanto, $k_{\text{opt}} = \underset{i}{\operatorname{argmax}}(|\lambda_i - \lambda_{i+1}|)$.

3.3. Aproximación incremental de los autovectores

Debido a que nuevos segmentos de trayectorias son incorporados al algoritmo de agrupamiento en cada instante de tiempo, el objetivo es procesar dichos datos y producir nuevos agrupamientos utilizando un enfoque incremental.

En [11] se propuso un método incremental eficiente que a partir de los autovectores de la matriz laplaciana de un grafo G_t permite obtener los autovectores de la matriz laplaciana del grafo G_{t+1} que se obtuvo agregando o eliminando vértices o modificando los pesos de las aristas de G .

Para esto plantean a las modificaciones del grafo como la suma de matrices simétricas $L_{t+1} = L_t + U$, donde L_t es la matriz laplaciana del grafo G_t y U es una matriz que codifica el cambio realizado a G_t para transformarlo en G_{t+1} . La idea central consiste en calcular la descomposición de L_t y luego aproximar la descomposición de rango k de $L_{t_k} + U$, donde L_{t_k} es la aproximación de L_t usando los k autovectores mas grandes. El método aproxima los autovectores de $L_k + U$ a calculando $\tilde{Q}H_k$, donde \tilde{Q} es una matriz con columnas ortonormales que generan $L_k + U$ y H_k representan los k autovectores mas grandes de la matriz $\Delta = \tilde{Q}(L_k + U)\tilde{Q}^T$. En [11] se muestra que las dos matrices pueden ser obtenidas a partir de L_k y U en $O((k^2 + p^2)(p + k) + np(p + k))$, donde n es la cantidad total de patrones, p es la cantidad de datos modificados y k es la cantidad de autovectores obtenidos.

4. Descripción del método propuesto

Dado el tamaño de la ventana de tiempo a analizar tw , un conjunto de segmentos de trayectorias $C = \{l_1, l_2, \dots, l_n\}$, un segmento l_i , y un parámetro de vecindad β , se define el conjunto de segmentos vecinos de l_i como

$$N(l_i, \beta) = \{ l_j | \beta B(l_i) \cap \beta B(l_j) \neq \emptyset \forall_{i \neq j} \}.$$

donde $\beta B(l_j)$ es el rectángulo delimitador mínimo de l_j escalado por β .

En el método propuesto se utiliza como función de semejanza entre dos segmentos de trayectoria:

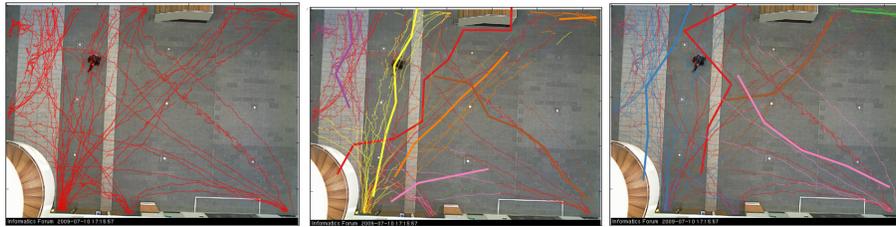
$$w(s_i, s_j) = \exp\left(-\frac{\operatorname{dist}(s_i, s_j)^2}{\alpha_i \alpha_j}\right) * \exp\left(-\frac{|t_{s_i} - t_{s_j}|}{0,5tw}\right) \quad (3)$$

donde $\alpha_i = \frac{\sum_{\forall l_j \in N(l_i, \beta)} \text{dist}(l_i, l_j)}{|N(l_i, \beta)|}$ y dist es la función de distancia entre dos segmentos definida en la ecuación 1.

Sean k el máximo número posible de clusters, β un parámetro de vecindad, T el número de iteraciones que transcurren sin volver a calcular el sistema de autovectores. Sea $G_{t-1} = (V_{t-1}, E_{t-1})$ el grafo de segmentos de trayectorias obtenido en la iteración $t - 1$, J_{t-1} los autovectores aproximados obtenidos en la iteración $t - 1$ y traj_i los segmentos de trayectorias extraídos en la iteración i . En cada iteración el método realiza los siguientes pasos:

1. $G_i = G_{i-1}$ donde $G_i = (V_i, E_i)$.
2. Se eliminan del grafo G_i los segmentos $\{s_j | s_j \in V_i, i - t_{s_j} > tw\}$.
3. $V_i = V_i \cup \text{traj}_i$. Se conecta cada $t \in \text{traj}_i$ con sus $N(t, \beta)$. El peso de la arista está dado por la ecuación 3.
4. Se calcula la escala local $\forall t \in \text{traj}_i$.
5. Se obtiene la matriz laplaciana L_i a partir de G_i de acuerdo a [11].
6. Si $i \bmod T = 0$.
 - a) Se recalculan las escalas locales de cada punto.
 - b) Se vuelven a calcular los k autovectores J_i a partir de L_i .
7. Si $i \bmod T \neq 0$.
 - a) Se aproximan los k autovectores J_i de acuerdo a la sección 3.3 a partir de J_{i-1} .
8. Se calcula el número de agrupamientos k_{opt} según lo expuesto en la sección 3.2.
9. Se aplica k -medias con centroides iniciales \mathbf{c}_{i-1} sobre J_i para obtener k_{opt} centroides. Se almacena \mathbf{c}_i .
10. Se extraen las trayectorias representativas de cada cluster.

5. Resultados Experimentales



(a) Trayectorias en una iteración (b) Trayectorias representativas obtenidas con $\beta = 5$ (c) Trayectorias representativas obtenidas con $\beta = 7$

Figura 3: Trayectorias extraídas en el conjunto de datos de utilizando una ventana de tiempo de 100 cuadros y $k = 15$

El rendimiento del algoritmo de clustering propuesto fue evaluado sobre un conjunto de trayectorias extraídas de videos de vigilancia reales. El conjunto de datos provisto por [19] consiste en un conjunto de trayectorias obtenidas mediante el seguimiento de personas en el edificio principal de la Escuela de Informática de la Universidad de Edimburgo. Los datos contienen varios meses de observaciones que resultan en más de mil trayectorias observadas por cada día. La figura 3 muestra la influencia del parámetro de escala en la extracción de trayectorias para un instante de tiempo dado. Se puede visualizar como el método permite extraer trayectorias con diferente nivel de detalle.

Uno de los conjuntos de datos provisto por [20] consiste en trayectorias simuladas en una intersección de calles. El algoritmo de agrupamiento fue evaluado utilizando el criterio propuesto en [13] que denominamos SSQ . El criterio consiste en minimizar el promedio de la suma de las distancias al cuadrado de los segmentos pertenecientes a cada cluster. La figura 4 muestra una comparación entre la suma de los errores al cuadrado de los agrupamientos obtenidos mediante el método propuesto y el algoritmo de agrupamiento de trayectorias TRACCLUS [13] para cuatro instantes de tiempo utilizando una ventana de tiempo de 500 iteraciones. En cada instante se muestra la cantidad de segmentos de trayectorias presentes. Comparando con TRACCLUS, el promedio de la suma de las distancias al cuadrado es ligeramente mayor, pero el tiempo de procesamiento del método propuesto es significativamente menor. Para procesar los 5000 puntos en promedio de cada cuadro el método propuesto toma en promedio 0.6 segundos, mientras que TRACCLUS toma alrededor de 8 segundos. Esto se debe a que solo se actualizan los segmentos de trayectorias incorporados o eliminados a la escena y los segmentos cercanos a ellos.

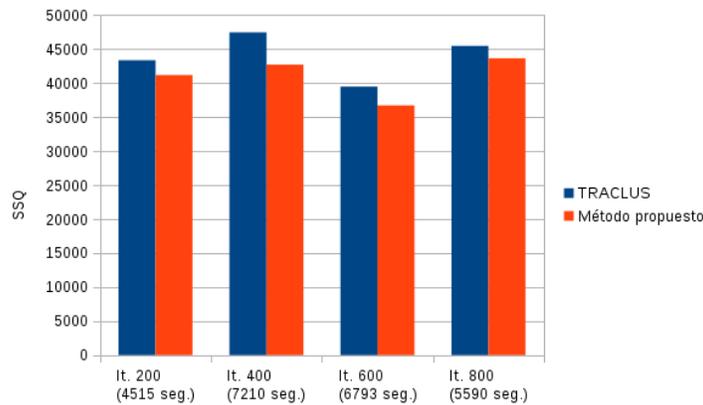


Figura 4: Comparación sobre la efectividad el algoritmo de agrupamiento

6. Conclusiones

En este artículo se presenta un método de agrupamiento incremental de trayectorias obtenidas mediante el seguimiento de objetos en vídeo. Los resultados obtenidos sobre vídeos de vigilancia reales presentan resultados preliminares satisfactorios. El algoritmo incorpora adecuadamente la información provista por los nuevos segmentos de trayectorias mediante la actualización incremental del sistema de autovalores. El método permite obtener las trayectorias representativas de cada cluster en una ventana tiempo. Una etapa futura de este trabajo prevé la incorporación de técnicas de discretización de autovectores con el fin de evitar el paso de aplicar un algoritmo de agrupamiento sobre el espacio generado por los autovectores. Otro aspecto importante sería evaluar la conveniencia de utilizar una medida de semejanza alternativa entre trayectorias.

Referencias

1. B. T. Morris and M. M. Trivedi, "A survey of vision-based trajectory learning and analysis for surveillance," *IEEE transactions on circuits and systems for video technology*, vol. 18, no. 8, pp. 1114–1127, 2008.
2. N. Anjum and A. Cavallaro, "Multifeature object trajectory clustering for video analysis," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 18, no. 11, pp. 1555–1564, 2008.
3. G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, and Y. Ma, "Robust recovery of subspace structures by low-rank representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 1, pp. 171–184, 2013.
4. T. Brox and J. Malik, "Object segmentation by long term analysis of point trajectories," in *European conference on computer vision*, pp. 282–295, Springer, 2010.
5. M. Vrigkas, V. Karavasilis, C. Nikou, and I. A. Kakadiaris, "Matching mixtures of trajectories for human action recognition," *Computer Vision and Image Understanding*, vol. 19, pp. 27–40, 2014.
6. X. Wang, K. T. Ma, G.-W. Ng, and W. E. L. Grimson, "Trajectory analysis and semantic region modeling using nonparametric hierarchical bayesian models," *International journal of computer vision*, vol. 95, no. 3, pp. 287–312, 2011.
7. Z. Li, J.-G. Lee, X. Li, and J. Han, "Incremental clustering for trajectories," in *International Conference on Database Systems for Advanced Applications*, pp. 32–46, Springer, 2010.
8. Z. Zhang, K. Huang, T. Tan, P. Yang, and J. Li, "Red-sfa: Relation discovery based slow feature analysis for trajectory clustering," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.
9. C. Fowlkes, S. Belongie, F. Chung, and J. Malik, "Spectral grouping using the nyström method," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, pp. 214–225, February 2004.
10. H. Ning, W. Xu, Y. Chi, Y. Gong, and T. S. Huang, "Incremental spectral clustering by efficiently updating the eigen-system," *Pattern Recognition*, vol. 43, no. 1, pp. 113–127, 2010.
11. C. Dhanjal, R. Gaudel, and S. Cléménçon, "Efficient eigen-updating for spectral graph clustering," *Neurocomputing*, vol. 131, pp. 440–452, 2014.

12. T. Kong, Y. Tian, and H. Shen, "A fast incremental spectral clustering for large data sets," in *2011 12th International Conference on Parallel and Distributed Computing, Applications and Technologies*, pp. 1–5, IEEE, 2011.
13. J.-G. Lee, J. Han, and K.-Y. Whang, "Trajectory clustering: a partition-and-group framework," in *Proceedings of the 2007 ACM SIGMOD international conference on Management of data*, pp. 593–604, ACM, 2007.
14. J. Shi and J. Malik, "Normalized cuts and image segmentation," in *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*, pp. 731–737, Jun 1997.
15. A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS*, pp. 849–856, MIT Press, 2001.
16. M. Belkin and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," *Neural computation*, vol. 15, no. 6, pp. 1373–1396, 2003.
17. L. Zelnik-manor and P. Perona, "Self-tuning spectral clustering," in *Advances in Neural Information Processing Systems*, pp. 1601–1608, 2005.
18. U. Von Luxburg, "A tutorial on spectral clustering," *Statistics and computing*, vol. 17, no. 4, pp. 395–416, 2007.
19. B. Majecka, "Statistical models of pedestrian behaviour in the forum,"
20. B. Morris and M. Trivedi, "Learning trajectory patterns by clustering: Experimental studies and comparative evaluation," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pp. 312–319, IEEE, 2009.