

CB-SMoT+: UNA EXTENSIÓN AL ALGORITMO CB-SMoT

CB-SMoT+: AN EXTENSION TO THE CB-SMoT ALGORITHM

Ph.D. Francisco Moreno

*Universidad Nacional
Medellín, Colombia*

Est. Guillermo Orrego

*Universidad Nacional
Medellín, Colombia*

Est. Andrés Pineda

*Universidad Nacional
Medellín, Colombia*

(Recibido el 19/02/2012. Aprobado el 19/03/2012)

Resumen. Una trayectoria es un registro de la evolución de la posición de un objeto móvil. Por ejemplo, un vehículo que se mueve en el espacio durante un intervalo de tiempo. Una trayectoria se representa mediante una secuencia de observaciones que indican la posición y el tiempo en el que fue tomada cada observación. CB-SMoT es un algoritmo que identifica las partes de una trayectoria durante las cuales el objeto mantuvo una velocidad promedio por debajo de un límite dado. En este artículo se propone una extensión para dicho algoritmo que permite identificar las partes de una trayectoria durante las cuales el objeto mantuvo una velocidad promedio entre observaciones por debajo de un límite dado. Esto posibilita la identificación, por ejemplo, de violaciones a un límite de velocidad que no son advertidas por el algoritmo original. Para el estudio se usó el sistema de gestión de bases de datos PostgreSQL y los algoritmos se implementaron en su lenguaje de programación, llamado PL/pgSQL. Además, se hicieron experimentos con 100 trayectorias de vehículos con el propósito de mostrar la utilidad y la viabilidad de la propuesta.

Palabras clave: Algoritmo CB-SMoT; Movimiento; Objetos móviles; Trayectoria; Velocidad.

Abstract. A trajectory is the record of the evolution of the position of a moving object. For example, a vehicle which moves in a space during a period of time. A trajectory is represented by a sequence of observations that indicate the position and the time when an observation was taken. The CB-SMoT is an algorithm that identifies the parts of a trajectory during which the object had an average speed below a given threshold. In this paper we propose an extension to this algorithm in order to identify the parts of a trajectory during which the object had an average speed between observations below a given threshold. This allows the identification, for example, of violations of a speed limit that are not identified by the original algorithm. We used PostgreSQL as the database management system and the algorithms were developed in its programming language, called PL/pgSQL. In addition, we conducted a series of experiments with 100 vehicle trajectories in order to show the usefulness and feasibility of our proposal.

Keywords: CB-SMoT algorithm, movement, moving objects, trajectories, speed.

1. INTRODUCCIÓN

Gracias a los Sistemas de Posicionamiento Global (GPS, por sus siglas en inglés) [1] y a los dispositivos móviles, es posible recopilar datos sobre la posición de objetos móviles como personas, animales y vehículos, entre otros, en un instante y durante un intervalo de tiempo. Esto ofrece una amplia gama de oportunidades para el desarrollo de aplicaciones, enfocadas en la gestión, en el análisis y en la extracción de conocimiento a partir de estos datos y en múltiples dominios.

Una trayectoria es un registro de la evolución de la posición (percibida como un punto), de un objeto que se mueve en el espacio, durante un intervalo de tiempo, con el fin de alcanzar un objetivo [2]. Una trayectoria se representa mediante una secuencia de observaciones $\langle \text{Obs}_1, \text{Obs}_2, \dots, \text{Obs}_n \rangle$, donde $\text{Obs}_i = (x_i, y_i, t_i)$, $x_i, y_i \in \mathbb{R}$, $t_i \in \mathbb{R}^+$ para $i = 0, 1, 2, \dots, n$ y $t_1 < t_2 < \dots < t_n$, donde x_i, y_i representan la posición del objeto y t_i el tiempo de la observación.

En [3] se propone el algoritmo SMoT (*Stops and Moves Of Trajectories*), que identifica las paradas, es decir, los sitios en los que el objeto móvil permaneció un tiempo superior a un límite establecido. En [4] se presenta una alternativa para el algoritmo SMoT denominada CB-SMoT (*Clustering-Based SMoT*) donde la identificación de las paradas (denominadas *clusters*) se basa en la velocidad del objeto. La consideración esencial del algoritmo CB-SMoT es que las partes de la trayectoria en las que la velocidad promedio fue baja (el analista de la aplicación establece un límite), corresponden a las paradas hechas por el objeto. Por ejemplo, considérese un turista que visita una ciudad; posiblemente su velocidad es baja cuando visita un museo o un restaurante. Otro ejemplo, es el de un vehículo que se encuentra en una zona de congestión de tráfico.

En este artículo se presenta un algoritmo llamado CB-SMoT+ que también se basa en la velocidad para identificar las paradas de un objeto. La diferencia con el algoritmo CB-SMoT es la siguiente: supóngase que el analista establece un límite máximo de velocidad igual a 80 km/h y que el algoritmo CB-SMoT identificó una parada de un objeto entre las 3:00 pm y las 4:00 pm y la velocidad promedio fue de 50 km/h. Nótese que por ser una velocidad promedio, el objeto pudo haber superado dicho límite en algún momento de ese intervalo. Sin embargo, el algoritmo CB-SMoT no detecta estos casos.

El nuevo algoritmo CB-SMoT+ procura garantizar que el límite no haya sido violado en ningún momento dentro del intervalo de tiempo de la parada. Esto permite analizar con mayor detalle el comportamiento de la velocidad del objeto. Por ejemplo, identificar violaciones a los límites de velocidad que pasan desapercibidas por el algoritmo CB-SMoT. Además, el algoritmo propues-

to también considera rangos de velocidad, aspecto no contemplado por el algoritmo CB-SMoT.

Esta propuesta puede ser útil en áreas como transporte, análisis y prevención de accidentes, deportes (carreras de autos, atletismo), y otras que estén relacionadas con el monitoreo de la velocidad con la que se mueven los objetos.

En cuanto a trabajos similares, en [5] se estudia la relación entre la velocidad, la densidad de vehículos en una vía y el flujo vehicular y la forma en la que dicha relación está influenciada por los controles de velocidad. Se analizan los límites de velocidad variables (VSL), es decir, diferentes límites de velocidad establecidos en cada tramo de una vía (los VSL son una estrategia usada comúnmente para mitigar la congestión vehicular). El objetivo es encontrar la relación que permita analizar y modelar de manera efectiva el flujo vehicular en ciertas circunstancias.

En [6] se evalúa la implementación de un sistema inteligente de transporte mediante la estrategia de VSL, para mejorar la seguridad en autopistas en las que se ha detectado que es alta la probabilidad de ocurrencia de accidentes de tránsito.

En [7] se investiga la relación entre los límites de velocidad, la velocidad a la que efectivamente se desplazan los vehículos y los accidentes de tránsito. Por ejemplo, el flujo vehicular a una velocidad alta posibilita la reducción de los tiempos de viaje; sin embargo, puede ocasionar un aumento en el número de accidentes de tránsito. El objetivo es establecer los límites de velocidad más adecuados de acuerdo con las circunstancias.

El artículo está organizado así: en la sección 2 se presenta un ejemplo motivador, en la sección 3 se propone el algoritmo CB-SMoT+, en la sección 4 se presentan algunos resultados experimentales con datos reales y en la sección 5 se concluye el artículo y se plantean trabajos futuros.

2. EJEMPLO MOTIVADOR

Supóngase que las autoridades de tránsito están interesadas en monitorear la velocidad de circulación de los vehículos en una ciudad. Se necesita identificar cuáles son los tramos de las vías en los que los vehículos transitan rápido o lento (i.e., superan o no un límite de velocidad establecido por las autoridades). Un tramo de una vía donde los vehículos transitan a una velocidad baja puede ser indicio de problemas como: automóviles que se estacionan en lugares no autorizados y obstruyen el flujo vehicular, daños o averías en la vía, exceso de controles (semáforos, ressaltos), entre otros. Por otro lado, un tramo donde los

vehículos transitan a una velocidad alta podría ayudar a identificar zonas de alta accidentalidad y que, por lo tanto, requieren controles adicionales.

El algoritmo CB-SMoT [4] identifica, a partir de la trayectoria de un objeto, e.g., un vehículo, las zonas (*clusters*) en las que el objeto mantuvo una Velocidad Promedio Total (VPT) por debajo de un límite establecido y el *cluster* abarcó un tiempo mínimo dado.

Considérese, por ejemplo, la trayectoria de un vehículo, como se muestra en la Fig. 1. La trayectoria se compone de 36 observaciones y en cada observación se registra la posición (x, y) y el tiempo en el que ocurrió. Por simplicidad, se supone que la velocidad del vehículo se mantuvo constante entre observación y observación (suposición de velocidad constante [8]). Al aplicar el algoritmo CB-SMoT, se identifican los *clusters* G1 (ver Tabla 2) y G2 (ver Tabla 3). Estos *clusters* corresponden a tramos donde la VPT del vehículo no

superó un límite dado (35 km/h) y el *cluster* abarcó un tiempo mínimo dado (40 minutos).

Sin embargo, nótese que aunque la VPT del vehículo durante el *cluster* G1 no superó el límite de velocidad, hubo ocasiones dentro del *cluster* en las que el vehículo *sí* superó dicho límite. Por ejemplo, la Velocidad Promedio Total entre las Observaciones (VPO) 6 y 7 fue 80,5 km/h y entre la 10 y la 11 fue 53,7 km/h. Por otro lado, en el *cluster* G2 la VPO no superó el límite de velocidad en ninguna ocasión.

Aunque el algoritmo CB-SMoT identifica *ambos clusters*, porque en ambos la VPT no superó el límite dado, interesa determinar solamente aquellos *clusters* donde la VPO no superó el límite (35 km/h), es decir, *clusters* como G2. Además, también puede ser de interés para las autoridades de tránsito determinar los *clusters* donde la VPO se mantuvo por encima de un límite o en un rango dado. Estos casos no son soportados por el algoritmo CB-SMoT.

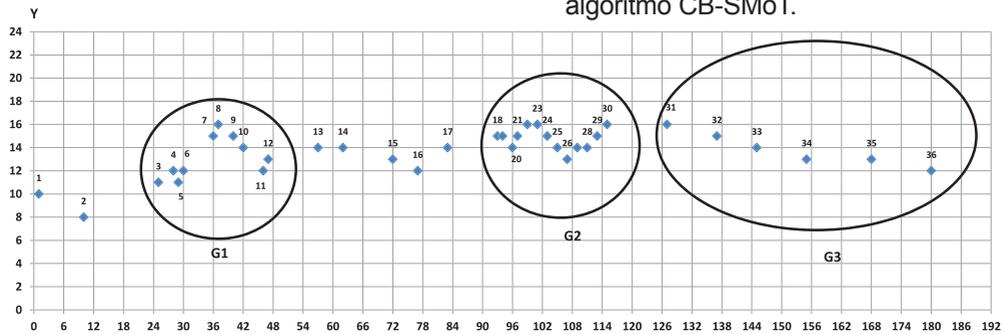


Fig. 1. Trayectoria

Tabla 1. Observaciones de G1

Obs.	x	y	t HH:MM	d(km) entre Obs. _{i-1} y Obs. _i	v(km/h) entre Obs. _{i-1} y Obs. _i
3	25	11	08:10		
4	28	12	08:15	3,2	37,9
5	29	11	08:20	1,4	17,0
6	30	12	08:25	1,4	17,0
7	36	15	08:30	6,7	80,5
8	37	16	08:35	1,4	17,0
9	40	15	08:40	3,2	37,9
10	42	14	08:45	2,2	26,8
11	46	12	08:50	4,5	53,7
12	47	13	08:55	1,4	17,0

Tabla 2. Observaciones de G2

Obs.	x	y	t HH:MM	d(km) entre Obs. _{i-1} y Obs. _i	v(km/h) entre Obs. _{i-1} y Obs. _i
18	93	15	09:25		
19	94	15	09:30	1,0	12,0
20	96	14	09:35	2,2	26,8
21	97	15	09:40	1,4	17,0
22	99	16	09:45	2,2	26,8
23	101	16	09:50	2,0	24,0
24	103	15	09:55	2,2	26,8
25	105	14	10:00	2,2	26,8
26	107	13	10:05	2,2	26,8
27	109	14	10:10	2,2	26,8
28	111	14	10:15	2,0	24,0
29	113	15	10:20	2,2	26,8
30	115	16	10:25	2,2	26,8

Tabla 3. Observaciones de G3

Obs.	x	y	t HH:MM	d(km) entre Obs _{i-1} y Obs _i	v(km/h) entre Obs _{i-1} y Obs _i
31	127	16	10:30		
32	137	15	10:35	10,0	120,6
33	145	14	10:40	8,1	96,7
34	155	13	10:45	10,0	120,6
35	168	13	10:50	13,0	156,0
36	180	12	10:55	12,0	144,5

3. ALGORITMO CB-SMoT+

3.1 Definiciones

Condición de velocidad entre observaciones (CVO): es una condición que especifica la velocidad entre las observaciones consecutivas de una trayectoria. Por ejemplo, $CVO = (\leq 80)$, indica que la velocidad máxima entre observaciones es 80 (e.g., km/h). $CVO = (\leq 30 \wedge \leq 80)$, indica que la velocidad entre observaciones debe ser mínimo 30 y máximo 80.

Neighbour_Obs: dada una trayectoria $t = \langle Obs_1, Obs_2, \dots, Obs_n \rangle$, el conjunto de observaciones vecinas de una observación Obs_i , denotado $Neighbour_Obs(i)$, está conformado por el mayor número

posible de observaciones de t cuya velocidad entre Obs_i y Obs_{i+1} para $j = i, i+1, \dots, k$ y $k < n$, satisface la CVO (ver Figura 2). Este conjunto de observaciones conforma un *cluster*.

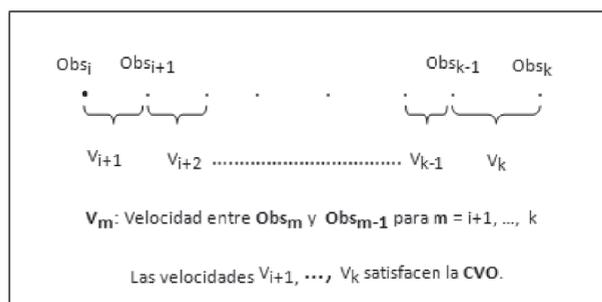


Fig. 2. Neighbour_Obs(i)

3.2 Algoritmo CB-SMoT+

A continuación se presenta el pseudo-código del algoritmo CB-SMoT+.

ALGORITMO CB-SMoT+

```

INPUT: T //Conjunto de trayectorias
           CVO //Condición de velocidad entre observaciones
OUTPUT: clusters //Conjunto de clusters que satisfacen la CVO
BEGIN
clusters = new clusters(); //Conjunto vacío de clusters
FOR each trajectory T ∈ T DO
  n = size(T); //Número de observaciones de T
  i = 1; //Variable para recorrer las observaciones de T
  WHILE (i < n) DO //Se recorre T desde su primera hasta su penúltima observación
    Neighbour_Obs = ∅; //Se inicializa el conjunto de observaciones del cluster
    Neighbour_Obs.add(T[i]); //Se agrega la observación actual (primera observación
del cluster)
    j = i; //Se guarda el número de la observación actual
    WHILE (CVO(T[i], T[i+1]) AND i < n) DO //Se verifica la CVO
      //Se agrega la observación i+1 al cluster
      Neighbour_Obs.add(T[i+1]);
      i = i + 1; //Número de la siguiente observación de T
    END WHILE
    IF (i <> j) THEN //Se valida que el cluster contenga al menos dos observaciones
      clusters.add(Neighbour_Obs, id(T)); //Se agrega el cluster junto
con el identificador de T
    END IF
    i = i+1; // Número de la siguiente observación de T
  END WHILE
END FOR
END
  
```

Por ejemplo, al aplicar el algoritmo a la trayectoria de la Figura 1, con $CVO = (> 95 \text{ km/h})$ se obtiene el *cluster* $G3 = \{Obs_{31}, Obs_{32}, Obs_{33}, Obs_{34}, Obs_{35}, Obs_{36}\}$.

4. EXPERIMENTOS

Para validar la propuesta se usaron datos de 100 trayectorias hechas por vehículos en Rio de Janeiro. Se usó el sistema de gestión de bases de datos PostgreSQL y los algoritmos se implementaron en su lenguaje de programación PL/pgSQL. En cada trayectoria las observaciones fueron tomadas en intervalos de un segundo. Las pruebas se hicieron con diferentes límites superiores de velocidad. En la Tabla 4 se presentan los resultados obtenidos al aplicar el algoritmo CB-SMoT+.

Tabla 4. Resultados de los experimentos para el algoritmo CB-SMoT+

CVO Límite superior de velocidad (km/h)	Número de <i>clusters</i> identificados	Número de trayectorias	Tiempo promedio por <i>cluster</i> (h)	Número promedio de observaciones por <i>cluster</i>	Velocidad promedio por <i>cluster</i>
40	135	75	0,21	521	11,7
30	74	53	0,23	457	7,62
20	34	26	0,32	359	4,45
14	17	15	0,45	246	2,58
8	8	8	0,75	187	0,90

Nótese, por ejemplo, que a medida que el límite superior de velocidad disminuye, el número de *clusters* identificados también disminuye. Esto sugiere que la mayoría de los vehículos tuvieron periodos durante los cuales su velocidad promedio se mantuvo por encima de 30 km/h. De hecho, se detectaron 209 (135 + 74) *clusters*. Estos datos sugieren que cada vehículo tuvo en promedio dos *clusters* en los que su velocidad se mantuvo por encima de 30 km/h. Además, cada uno de estos *clusters* tuvo una duración promedio de 0,22 h, es decir, alrededor de 12 minutos.

Por otro lado, con un límite superior de velocidad de 8 km/h, se identificaron 8 *clusters* con un tiempo promedio de 40 minutos por *cluster*. Esto significa que hubo (a lo sumo) 8 vehículos que experimentaron periodos de tránsito lento (menor a 8 km/h) y cuya duración estuvo alrededor de 45 minutos (0,75 h). Esto sugiere posibles embotellamientos, zonas en las que por algún motivo no se debe superar dicho límite (una zona escolar o militar) o el conductor transitó lentamente. En la Tabla 5 se muestran los resultados obtenidos cuando se aplicó el algoritmo CB-SMoT.

Tabla 5. Resultados de los experimentos para el algoritmo CB-SMoT

CVO Límite superior de velocidad (km/h)	Número de <i>clusters</i> identificados CB-SMoT	Tiempo promedio por <i>cluster</i> (h)
40	90	0,31
30	53	0,26
20	21	0,24
14	7	0,22
8	2	0,19

Nótese que el algoritmo CB-SMoT detectó menos *clusters* que el CB-SMoT+ pero el tiempo promedio de cada *cluster* fue mayor. Esto es razonable porque el algoritmo CB-SMoT permite que haya violaciones al límite superior de velocidad, siempre y cuando la velocidad promedio del *cluster* no supere dicho límite. Por ello, se generan menos *clusters* pero de mayor duración y en los que no se alcanza a percibir si hubo violaciones al límite de velocidad establecido.

5. CONCLUSIONES

En este artículo se propuso una extensión para el algoritmo CB-SMoT. El nuevo algoritmo, llamado CB-SMoT+, permite analizar con mayor detalle (identificación de violaciones de límites) el comportamiento de un objeto móvil con respecto a su velocidad. La diferencia esencial entre los dos algoritmos es que el algoritmo CB-SMoT considera la velocidad promedio total (VPT) para la identificación de los *clusters* mientras que el algoritmo CB-SMoT+ considera la velocidad promedio entre observaciones (VPO).

Aunque se requieren experimentos más exhaustivos, los resultados mostraron la viabilidad de la propuesta y evidenciaron la diferencia entre los dos algoritmos y su utilidad ya que permiten identificar, por ejemplo, las zonas donde se presenta tráfico lento o rápido (con respecto a un límite de velocidad establecido).

Como trabajo futuro se espera extender el algoritmo para la identificación de *clusters* con más condiciones además de la CVO. Por ejemplo, identificar solamente los *clusters* que tienen una duración mínima o que recorrieron una distancia mínima o que pasaron por o cerca de determinados sitios o vías. También se pretende comparar el algoritmo propuesto con otros algoritmos de generación de *clusters*, por ejemplo, DBSCAN [9].

REFERENCIAS

- [1] P. A. Longley, M Goodchild, D. J. Maguire & D. W. Rhind. "Geographic Information Systems and Science". 560, p. 2010.
- [2] S. Spaccapietra, C. Parent, M. Damiani, J. Macedo, F. Porto & C. Vangenot. "A conceptual view on trajectories". *Data & Knowledge Engineering*, Vol. 65, No. 1, pp. 126-146, Abr. 2008.
- [3] L. O. Alvares, V Bogorny, B Kuijpers, J. Macedo, B. Moelans & A. Vaisman. "A model for enriching trajectories with semantic geographical information". In *Proc. 2007 15th Annual ACM International Symposium on Advances in Geographic Information Systems*, Nueva York, pp. 1-22.
- [4] A. Palma, V. Bogorny, B. Kuijpers & L. Alvares. "A Clustering-based Approach for Discovering interesting Places in Trajectories". In *Proc. 2008 ACM symposium on Applied computing*, Nueva York, USA, pp. 863-868.
- [5] B. G. Heydecker & J. D Addison. "Analysis and modelling of traffic flow under variable speed limits". *Transportation Research Part C: Emerging Technologies*, Vol. 19, No. 2, pp. 206-217, Abr. 2011.
- [6] M. Abdel-Aty, J. Dilmore & A. Dhindsa. "Evaluation of variable speed limits for real-time freeway safety improvement". *Accident Analysis & Prevention*, Vol. 38, No. 22, pp. 335-345, Mar. 2006.
- [7] A. A. M. Aljanahi, A. H. Rhodes. A. V. Metcalfe. "Speed limits and road traffic accidents under free flow conditions". *Accident Analysis & Prevention*, Vol. 31, No. 1, pp. 161-168, Ene. 1999.
- [8] R. H. Güting & M. Schneider. "Moving objects databases". Morgan Kaufmann: San Francisco. 2005.
- [9] M. Ester, H. -P. Kriegel, J. Sander & X. Xu. "A density-based algorithm for discovering clusters in large spatial databases with noise". In *Proc. 1996 Second Conference on Knowledge Discovery and Data Mining*, Portland, Oregon, USA, pp. 226-231.