



UNIVERSIDAD ANDRÉS BELLO

FACULTAD DE INGENIERIA

ESCUELA DE INGENIERIA CIVIL INFORMATICA

# **Mejoramiento en el rendimiento de la problemática Map-Matching**

Alumno Memorista: Ignacio P Zamorano.

Profesora Guía: Carola Blázquez L.

## Índice

<b>Resumen</b> .....	3
<b>Introducción</b> .....	4
<b>Problemática</b> .....	6
<b>Objetivos General</b> .....	7
<b>Objetivos específicos</b> .....	7
<b>Hipótesis</b> .....	7
<b>Alcance</b> .....	7
<b>Marco Teórico</b> .....	8
Map-Matching .....	8
Funcionamiento del algoritmo Map-Matching: .....	9
<b>Revisión bibliográfica</b> .....	11
<b>Metodología</b> .....	16
Identificación de las mejoras: .....	16
Implementación del código: .....	20
Preparación y selección de datos y testing: .....	25
Mejora de la estructura del algoritmo: .....	28
Análisis de resultados: .....	28
Comparación entre la versión actual y versión mejorada del MMA: .....	28
<b>Resultados</b> .....	29
Radio dinámico (Portage, EE. UU.): .....	29
Forzado de puntos sin ruteo (Renca, Chile.): .....	32
Forzado de puntos con ruteo (Renca, Chile.): .....	36
Heading (Renca, Chile.): .....	38
Tabla de resultados de mejoras: .....	41
Los resultados finales obtenidos .....	53
<b>Conclusión</b> .....	67
<b>Bibliografía</b> .....	68

## Resumen

El actual informe de título tiene como función dar a conocer una propuesta de mejora de sistema a un algoritmo de reglas de decisión resolviendo la problemática a ambigüedad espacial o también conocido en el inglés Map-matching (MMA) diseñado por la Doctora Carola Blázquez para su tesis doctoral (Blázquez, 2005). Este sistema Map-matching que será el tema central en este informe tiene como propósito la determinación del recorrido real de un vehículo por medio de los valores de datos GPS almacenados con una estructura ya sea como latitud, longitud y velocidad del vehículo, entre otras. Las aplicaciones que utilizan esta solución son por ejemplo los camiones recolectores de residuos sólidos domiciliarios o camiones para remover la nieve y hielo de las calles. De esta manera, se conocen las rutas por donde transitan y tener un control sobre estos.

Por último, se desea obtener como resultado una mejora de rendimiento y robustez del algoritmo utilizando distintas metodologías que se mencionan en este documento, para luego efectuar un análisis comparativo entre la versión original y la mejorada del sistema map-matching.

## Introducción

Para comenzar el sistema de posicionamiento global, más conocido globalmente como GPS por sus siglas en inglés, es un sistema que permite determinar en todo el planeta tierra la posición de cualquier objeto en este caso un vehículo con una precisión muy cercana a la calle. En la práctica son necesarios 4 o más satélites para determinar la posición con cierta precisión. Un buen ejemplo del uso del GPS es en la telefonía móvil son las aplicaciones que permiten conocer la posición de uno o la posición de amigos cercanos en tiempo real. Y, a partir de los puntos GPS se pueden obtener derivados como los parámetros por ejemplo la velocidad, distancia, dirección y altura entre otras, tal como se describió en el artículo por Karimí (2006).

La ambigüedad espacial consiste en base a las posiciones reales establecidas de un objeto en el espacio y la posición esperada o calculada del mismo objeto, en pocas palabras, esto sería no poder definir la ubicación exacta del objeto en el espacio en un tiempo determinado.

Para ello, se toman los datos de las ubicaciones por GPS o también con DGPS (Differential GPS) o GPS diferencial. En este proyecto, se utiliza GPS y la diferencia entre el DGPS para obtener los datos GPS siendo recepcionados por una estación ubicado en Tierra efectuando la corrección diferencial y proporcionando una mayor precisión en la posición. Agregando en el GPS se debe el gran margen de error por ejemplo a las alteraciones atmosféricas sobre todo lo referente a lluvias y mucho más a tormentas distorsionando las señales y los reflejos de edificios que producen fenómenos ópticos que causan grandes problemas en la precisión de las señales (GISGeography, 2018), como se ejemplifica en la Figura 1.

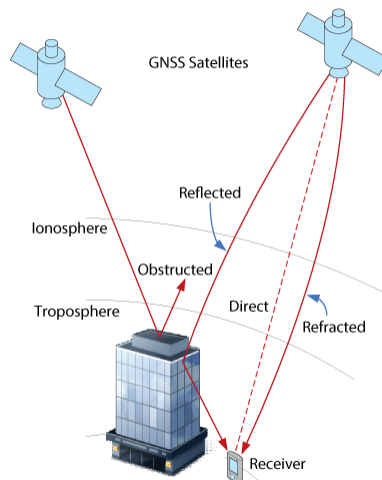


Figure 15 GNSS Signal Propagation

**Figura 1: Diagrama de error de ubicación GPS: Hexagon (2018)**

Además, existe el error de las redes viales que pueden estar mal posicionadas o desactualizadas. Hasta incluso el método que se usó para hacer la red vial no sea la más eficiente como por ejemplo si se hace a manual en una computadora o lo genere en forma automática con algún algoritmo. Pero uno de los métodos más eficientes es de la empresa Google ya que representa las redes viales con automóviles en ruta, es decir, en que transite un automóvil real por la red vial que represente con todo un sistema de equipamiento para ser mayormente preciso en el trazado de la red vial.

Los mayores usos de GPS que se pueden divisar cercanamente de esta tecnología en la vida real son en los computadores o en la más usada en la telefonía móvil conociendo la ubicación y conocer la ruta recorrida como por ejemplo en aplicaciones como GoogleMaps, Waze, Google Earth, que a través de la ubicación GPS de uno te puede dirigir por calles hacia el destino que uno desee.

Para determinar la ruta de un vehículo a través de los puntos obtenidos por GPS, estos vienen con un margen de error por distintos factores como fueron mencionados anteriormente, lo que se hace para contrarrestar este error es relacionar los puntos obtenidos con el eje de calzada más cercano, para en parte solucionar el problema.

Por último, la motivación para realizar dicho proyecto es el conocimiento cercano adquirido a previos proyectos realizados con GPS en Smartphone Android, encontrando puntos cercanos por distancia o categoría, o simplemente generando la ruta más factible entre dos puntos. Todo esto en algunos proyectos siendo integrado en otras tecnologías como en relojes inteligentes o radios con sistema Android. Siempre buscando y perfeccionándose en este ámbito que aún falta pulir en sus funciones. Pero siendo tan importante incluso formándose a través del tiempo una necesidad para el ser humano y viéndose aplicado en distintas aplicaciones del día a día.

## Problemática

Map-matching (su término en inglés) como propósito es determinar el recorrido real de un vehículo a partir de los valores capturados con un aparato GPS. Pero siempre no todo es tan perfecto al momento de resolver un asunto. Por ejemplo, para divisar el problema existe un ejemplo muy peculiar imaginando en la Figura 2 que la red vial de un vehículo correctamente es por “Road 1” y la incorrecta es “Road 2”. El punto GPS es representado por el punto amarillo en la Figura 2 donde claramente tiene un margen de error ya que no está inserto en la red vial y puede generar una idea errónea de ruta. Por lo tanto, viendo la diferencia de distancia entre las dos redes viales “Road 1” y “Road 2” al punto GPS la red vial más cercana al punto GPS es claramente “Road 2” dando como decisión incorrecta la relación del punto GPS y la red vial. Por esta razón se necesita un sistema robusto, que identifique bien las soluciones y no se vea marcado por los falsos positivos.

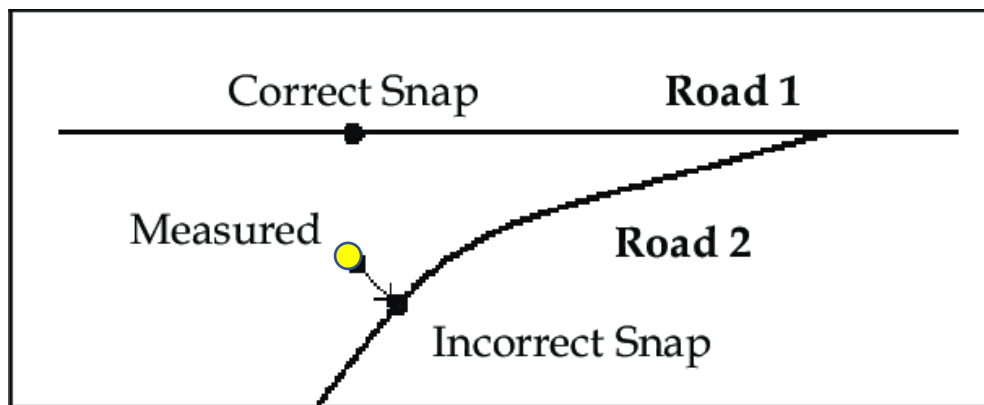


Figura 2: Measured GPS Data Point with Error Snapped to the Wrong Roadway Centerline: Blazquez (2012)

Frente a este caso común generado, se creó un sistema map-matching (MMA) topológico que resuelve la contrariedad de la ambigüedad espacial por la mediación del GPS facilitado por satélites. Pudiendo dar mayormente respuestas del recorrido real a pesar de la recepción de datos inexactos. Pero, aunque posea un buen rendimiento existen situaciones que no tienen buen desempeño y, por ende, requiere mejoras el sistema.

## Objetivos General

Mejorar el rendimiento en el algoritmo de Map-matching (ambigüedad espacial) inicial<sup>1</sup>.

## Objetivos específicos

- Identificar las mejoras al sistema Map-Matching (MMA).
- Implementar el código al MMA original
- Preparar y seleccionar un conjunto de datos y testing.
- Comparar resultados de la versión original y la versión mejorada del MMA.
- Estudiar la comparación entre el MMA existente y la versión mejorada del MMA.

## Hipótesis

En base a los estudios y el problema planteado donde el principal contratiempo es la precisión, el hallazgo nulo de respuesta y el tiempo de respuesta, siendo agregado los puntos de mejora que son:

1. Mejorar la calidad de la solución y tiempo computacional (minutos) del algoritmo en un 20%.

## Alcance

El alcance de esta etapa del proyecto es lograr un avance en integrar mejoras al algoritmo topológico de ambigüedad espacial y entregue soluciones más acercadas. Con DGPS y GPS en la obtención de los datos, coordenadas UTM, velocidades en el sistema mph y Km/h, tiempo en formato 24 horas y en una recolección de tiempo de estos datos de segundos de diferencia

---

<sup>1</sup> Se hace referencia al algoritmo final mejorado de cada hito anterior.

## Marco Teórico

### Map-Matching:

Por medio de esta inconsistencia de precisión y exactitud del GPS y además de la red vial, nace el sistema de Map-Matching. Como lógica por medio de las coordenadas geográficas registradas con un modelo lógico del mundo real GPS, es relacionarlos con el eje de la calzada para luego representar la ruta de un usuario o vehículo, tal como se puede ver reflejado en el artículo (A.Quddus, 2007) y en la Figura 3.

Los parámetros de Map-matching que mayormente son los valores de entrada al algoritmo son:

- **Buffer o radio:** Cualquier área espacial que se define a partir de un punto GPS en un mapa con una medida, por ejemplo, en pies o metros. Con un fin de encontrar referente a ese punto las calles más cercanas para insertarse en una de ellas.
- **Tolerancia de velocidad:** Rango de velocidad mínima y máxima que logra condicionar si está dentro del rango la velocidad resultante que existe entre dos puntos. Donde si logra estar dentro del rango se acepta a la condición, pero si no, se descarta buscando otra posibilidad.
- **Máxima cantidad de puntos consecutivos (N):** Como dice el nombre es la condición de la máxima cantidad de puntos que pueda tener una secuencia en busca de la solución. Si se llegase a pasar de la cantidad de puntos consecutivos se toma como si no hubiese solución para esa sección.



### Funcionamiento del algoritmo Map-Matching:

Ya teniendo como base lo anterior, se puede incorporar el funcionamiento del algoritmo Map-matching.

Primeramente, todo se inicia con los puntos GPS incorporados en el mapa con su respectiva coordenada y la red vial establecida igualmente. Cada punto GPS contiene un margen de error, por lo tanto, por medio de un radio establecido se analizan las vías cercanas a este en sentido de asociación al eje de la calzada por medio de restricciones para transformarse en solución de posicionamiento del punto en la ruta (Ver Figura 2), verificándose de a pares de puntos que coincidan en velocidad y distancia en un rango de error. Este sistema tiene una lógica que tiende a la cantidad de puntos consecutivos para encontrar un resultado o incluso no encontrarlo, es decir, tener una relación diferencia numérica no más de 5 puntos entre los dos puntos a analizar. Pero tiene una particularidad que, al momento de no encontrar los puntos alternativos referente a ejes de calzada de las calles empieza a avanzar hacia adelante y hacia atrás con la idea de incluso encontrar la solución entre los más lejanos forzando a los que queden en medio a formar parte de la solución de la ruta. De forma más técnica, se puede divisar en la Figura 3 en el diagrama de flujo. El algoritmo MMA se inicia con el fw con un valor inicial que es falso (0 en la informática), siguiendo pasa a verificar con los puntos snapeados de  $k_i$  y  $k_j$  verificando si existe una ruta posible de resultado. Si es así, entonces termina, en caso contrario, si no vuelve a buscar más alternativos a  $k_j$ , pero si encuentra verifica con los alternativos en  $k_j$  de una ruta como posible solución. Si es no, el MMA vuelve a buscar más alternativos en  $k_j$ . Luego como ya se mencionó, el caso positivo de encontrar alternativos  $k_j$  nos dirigimos al caso que no encuentre más, lo que sucede es que ahora busca alternativos  $k_i$ , si es sí, verifica base a  $(d_{i-1}, i, j)$ , es decir, el MMA verifica el punto anterior a  $k_i$ , con el actual  $k_i$  y el  $k_j$  actual, si existe una ruta posible de resultado termina, pero si vuelve a buscar otras alternativas a  $k_i$ . El otro caso en  $k_i$  que es el de no encontrar ninguna alternativa a  $k_i$ . Verifica el valor de N obteniendo con la diferencia entre  $j$  e  $i$ , porque si la diferencia entre ellos es mayor que N entonces "No existe solución". Sin embargo, viendo el caso que sea menor se comprueba que si fw es falso. En esta primera situación, esta variable sería false, entonces avanzamos uno en  $k_j$  y fw se vuelve a true, pero si hubiese sido el caso en que fw no hubiese sido falso, entonces se retrocede en uno con respecto a  $k_i$  y fw pasa a ser falso. No obstante, estos comparten la misma salida que sería de nuevo la comprobación inicio de verificar los puntos snapeados entre  $k_i$  y  $k_j$  siguiendo este ciclo hasta que las distancias de los puntos sean mayor N como fue mencionado anteriormente.

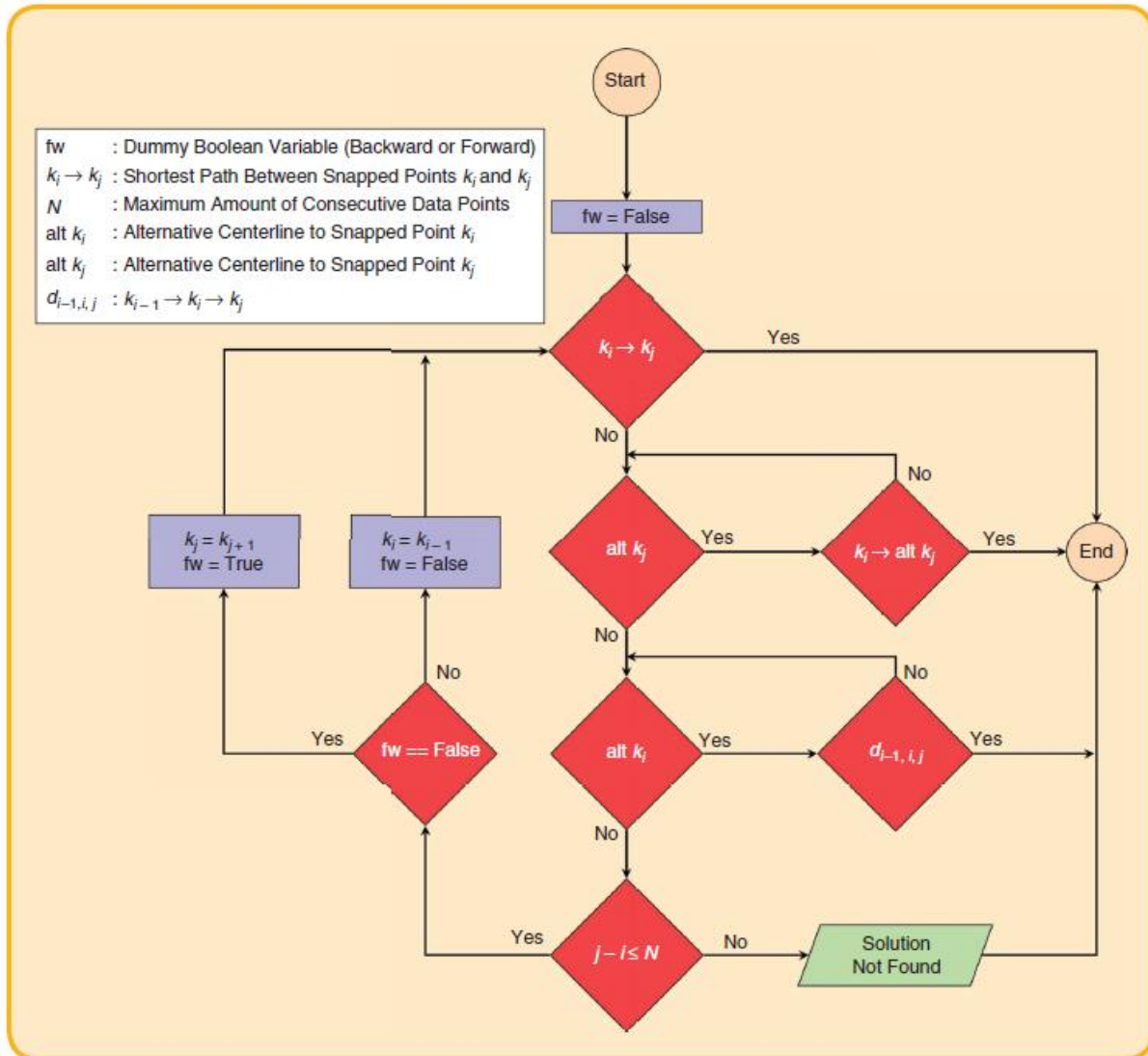


Figura 3: Diagrama de flujo del postproceso MMA. Fuente: Blazquez et al, (2018)

## Revisión bibliográfica

Se analizarán variadas bibliografías existentes de otros autores, el cual incluye artículos de proyectos similares de investigación, teniendo presente el algoritmo y las herramientas existentes.

Las bibliográficas estudiadas primero fueron las de (Brunsdon, 2004), que plantea una gran similitud con el sistema original a mejorar que comparte el objetivo que es el dar como resultado camino más probable de un automóvil referente a puntos GPS. Además, consta para comprobar la aceptación de los puntos a la calle es por medio de una tolerancia de velocidad con respecto a la diferencia de la velocidad. Discriminando a los puntos que no compartan la condición, si se comparte se asocia el punto a la calle más cercana con respecto a un radio estático.

Ahora bien, los puntos mencionados anteriormente pueden lograr asignar los puntos a las calles con un buen resultado de asignación, pero no quiere decir que estén asignados del todo bien con respecto al sentido de las calles porque se puede dar el caso que en un segmento de puntos de la ruta tenga un sentido, pero la calle por la que sea asociada tiene la dirección contraria a este, por lo tanto no perteneciendo la ruta a la calle y dando un falso positivo como resultado. Dándome paso a estudiar a (Tinkering, 2018), que plantea la comparación de la dirección angular de los puntos con respecto a la calle que se quiere asociar. Otorgando un resultado con una menor probabilidad de falso positivo.

Se dio con un artículo que justo comparte la idea de los puntos asociados a una calle que es (Bang, Kim, & Yu, 2016) que en simples palabras plantea que es el radio estático pero que ese mismo vaya acrecentándose gradualmente hasta que encuentre una calle cercana para asociarlo a este, por lo visto está más ambientado en curvas. Pero aun así se puede llevar a una idea general e integrarlo.

En los últimos años, el GPS se ha establecido como una importante tecnología de posicionamiento para proporcionar datos de ubicación para aplicaciones de ITS (Zito et al. 1995). Investigadores de todo el mundo han desarrollado una serie de algoritmos de coincidencia de puntos y rutas establecidas en un mapa como el análisis topológico de datos de redes de carreteras espaciales, teoría probabilística, filtro de Kalman, lógica difusa y teoría de creencias. Todas estas técnicas mencionadas han sido modificadas y ajustadas a una mejor exactitud y respuesta al igual que en la situación de los datos que han mejorado su calidad.

El propósito general de un algoritmo de correlación de mapas es identificar el segmento de carretera correcto en el que se desplaza el vehículo y determinar la ubicación del vehículo en ese segmento (Greenfeld, 2002; Quddus et al., 2003). Agregando a lo planteado por Quddus es mejorar la precisión ya que todo el Map-matching para que funcione y sea verídico es a través de buenos datos e instrumentos, ya que una ruta errónea podría llevar a un gran error.

Por otra raíz de un trabajo muy conocido es el algoritmo de coincidencia denominado Odometer Map Matched GPS con sus siglas (OMMGPS), que se puede ver reflejado en el artículo (Brunsdon, 2004) que plantea en palabras resumidas el lograr cumplir como objetivo el conocimiento de antemano el camino más probable de un viaje. En un inicio con el software existente de correlación de mapas del Sistema de Posicionamiento Global (GPS), llamado MMGPS. Posteriormente, se llamó OMMGPS con observaciones de odómetro. En la OMMGPS, la información de altura obtenida de los modelos de terreno digitales (DTM) se usa para obtener posiciones de puntos de GPS 3D, cuando solo tres satélites GPS son visibles para el receptor. Lo más importante es que la ayuda de altura mejora la precisión de las posiciones de puntos de GPS con geometría. OMMGPS usa distintos puntos a medir el error de distancia que va según el valor absoluto de la diferencia entre la distancia bruta y la distancia de referencia, valor absoluto de la diferencia entre el posicionamiento sin procesar y el posicionamiento de referencia y otro muy significativo es el heading según la dirección que se vaya avanzando se toma como referente de posible solución bajo a los ciertos criterios de aceptación del algoritmo.

Pero el que mayor marca la diferencia es la altura para lograr una mejor precisión y predicción. A continuación, como muestra en la Figura 4, se ve una referencia al trabajo de este algoritmo según varios puntos van descartando y aceptando la ruta más probable.

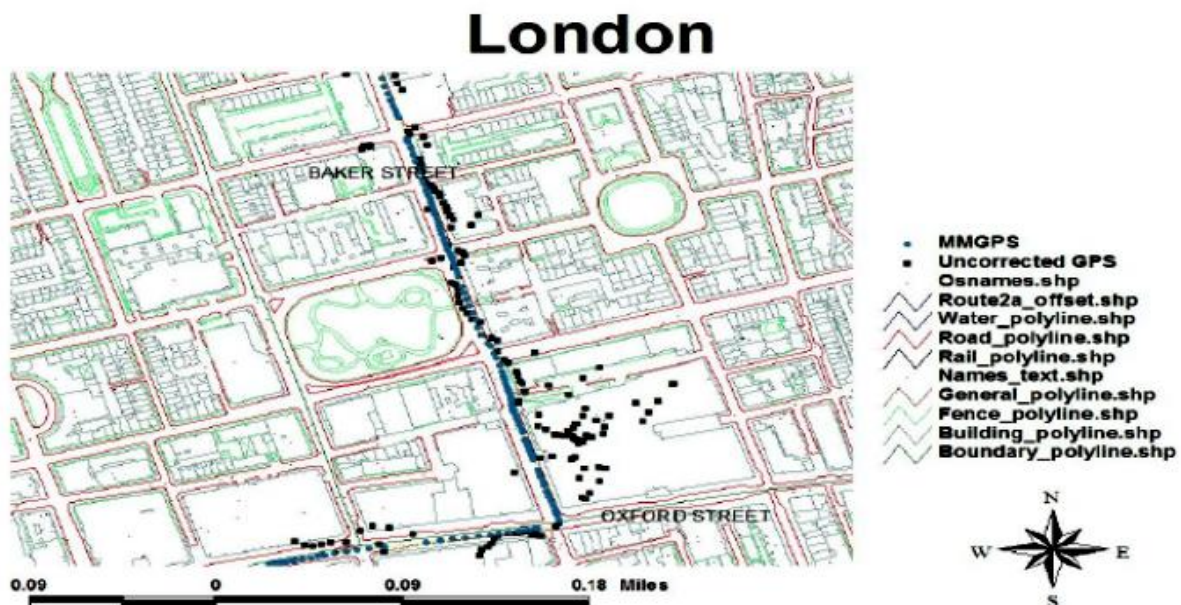


Figura 4: Map Matched GPS positions (MMGPS): Brunsdon, (2004)

Lo que existe como sistema de trabajo actual no está muy alejado a la realidad de OMMGPS donde se puede comparar la forma del rango de distancia y velocidad bajo un margen de error, tomando o discriminando puntos lógicos frente al problema, referenciando los puntos a una probabilidad de cercanía a la ruta usando un determinado

radio, la comparación de soluciones frente a una guía y por último visualizar como resultado la ruta más probable.

Con respecto a un punto muy importante que es el buffer, a lo existente en el proyecto está fijado, limitado a la búsqueda de más opciones puntos alternativos, dando una mayor probabilidad de no encontrar los puntos necesarios para un resultado.

Además, un punto muy favorable en estos sistemas es la dirección, donde por definición es la brújula en la que apunta el arco o la nariz del vehículo, es decir, es la dirección de la brújula en la que un dispositivo está viajando o debería estar viajando, medido en grados referente al Norte magnético y / o el Norte verdadero, tal como lo muestra en la Figura 4.1 (Tinkering, 2018). La tecnología y los productos (Global Navigation Satellite System, GNSS) de vanguardia de hoy permiten a los usuarios determinar con precisión hacia dónde apunta un objeto en particular en el plano horizontal. El rumbo del GNSS se calculan a partir de los informes de las dos últimas posiciones en las que se encontró el objeto; en otras palabras, dónde estaba el objeto antes y dónde se encuentra en el momento presente (Brunsdon, 2004). Por esta razón, una brújula GNSS requerirá que el objeto se mueva para determinar el rumbo. La mayoría de los dispositivos GNSS actualizarán los datos de posicionamiento en un intervalo de tiempo, por ejemplo, cada uno o dos segundos, proporcionando al usuario los cálculos más precisos posibles.

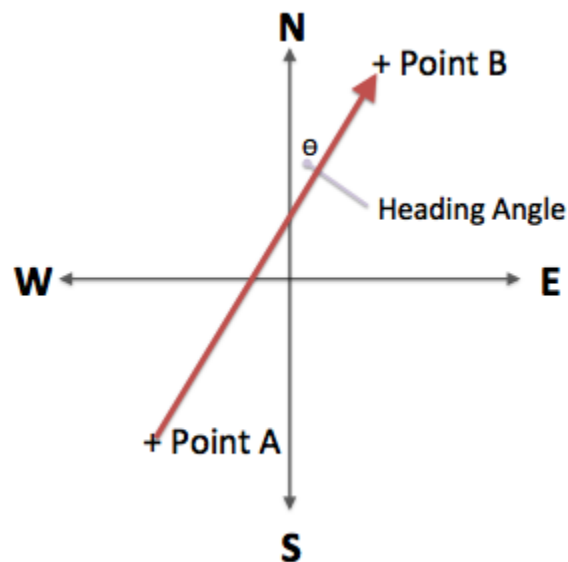


Figura 4.1: Getting from Point A to Point B: Wayne`s (2012)

### Asistencia de navegación personal y Map matching

White et al. (2000) discute una solución para el problema de Map-matching para asistencia de navegación personal (PNA). Cuatro diferentes algoritmos Map-matching fueron implementados y probados:

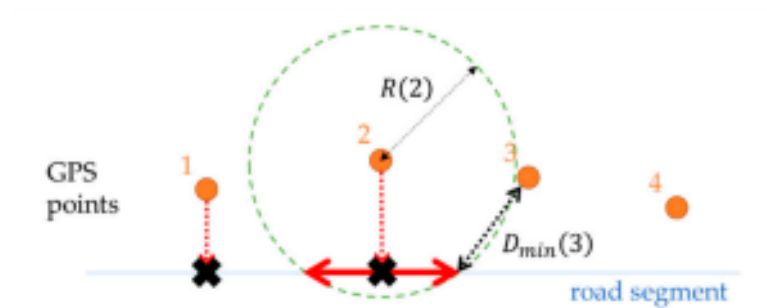
- Usa la mínima distancia (punto-a-arco).
- Comparación de la información de la dirección del vehículo con el arco y la trayectoria.
- Uso de topología para seleccionar caminos que son alcanzable desde el camino actual.
- Construcción de un conjunto de variables de las curvas de diferentes caminos, seguido por comparación de éstas hacia la línea central de las curvas usando puntos (curvas-a-curvas). Los autores concluyen que esos algoritmos trabajarían mejor cuando las distancias entre los puntos GPS y el camino más cercano sean pequeñas y que el correcto acierto tienda a ocurrir con mejor velocidad sobre calles rectas.

Najjar y Bonnifait (2007) propusieron un algoritmo Map-matching que funciona en tiempo real utilizando un sistema de asistencia a la conducción avanzada (ADAS), la estrategia de selección de segmentos de caminos se basa en múltiples criterios usando teoría de creencia (Belief Theory). Los autores pusieron atención particularmente en el desarrollo de las funciones de creencia basándose en las mediciones y estimaciones de distancias relativas, dirección del vehículo (heading) y velocidades.

Bang et al (2016) desarrollaron un método novedoso de correlación de mapas utilizando el GPS. Este método reduce el error en la posición, que es una suma de las varias fuentes, incluido el retraso de la señal debido a la ionosfera y a la atmosfera. Esta estimación es una corrección de mapa coincidente, que proporciona una alternativa autónoma a DGPS para la navegación en el automóvil. Este método es basado en la distancia de Fréchet aplicándose a una trayectoria completa. Por ejemplo, para explicar cómo funciona, una ilustración muy popular de la métrica de Fréchet muestra a un hombre paseando a su perro. Él está caminando en una curva y el perro camina en la otra curva. Ambos sujetos pueden controlar su velocidad, pero no se les permite ir hacia atrás. La distancia de Fréchet entre las curvas es la longitud mínima necesaria para una correa.

Dado el ejemplo, la distancia de Fréchet es el valor mínimo del radio de búsqueda (buffer) y se deja un umbral de limite llamado error posicional máximo, donde va agrandándose gradualmente el valor mínimo del radio de búsqueda hasta que logre cada uno de los puntos coincidir con la red de la carretera integrándose inmediatamente a ésta. Ésta es la idea del radio dinámico, como se ve reflejada la Figura 4.2.

El radio dinámico se aplica al sistema original cuando los puntos GPS  $k_i$  y  $k_j$  se tratan de buscar los puntos alternativos a las calles, accionando el aumento gradual del radio cuando no se encuentren puntos alternativos. El aumento es hasta un cierto límite que es ingresado manualmente al comienzo del código.



**Figura 4.2: An Improved Map-Matching Technique Based on the Fréchet Distance Approach for Pedestrian Navigation Services: Bang et al (2016)**



## Metodología

Identificación de las mejoras:

Las versiones base de los algoritmos a trabajar se encuentran en GitHub:

- [https://github.com/nachozamorano/Map-matching/tree/Version-base-\(Portage\)](https://github.com/nachozamorano/Map-matching/tree/Version-base-(Portage))
- [https://github.com/nachozamorano/Map-matching/tree/Version-base-\(Renca\)](https://github.com/nachozamorano/Map-matching/tree/Version-base-(Renca))

Se busca mejorar el rendimiento del algoritmo a través del estudio, identificación y planificación de mejoras al sistema MMA. Las mejoras se identificarían en puntos que puedan mejorar el rendimiento del MMA que se está trabajando. En este estudio, se tomaron como base los artículos estudiados anteriormente en la revisión bibliográfica.

La primera mejora identificada es el radio dinámico que nació de la idea del paper de (Bang, Kim, & Yu, 2016), de cómo los puntos GPS establecidos en el mapa traen cierto margen de error y el algoritmo de su sistema cuenta con un radio es estático y con la implementación de un radio dinámico. Por consiguiente, existen casos en que los puntos no alcanzan a asociarse a una calle más allá de ese rango establecido, y no se asociaba a una calle, por lo tanto, la forma de alcanzar una calle que está más lejana a ese tamaño de buffer es que éste se aumente de forma constante hasta que dicho radio intercepte la vía o llegue a un límite máximo. Por ejemplo, como muestra la Figura 5, en base al punto tiene un radio base de “color celeste”, que no intercepta ninguna vía, pero al incrementar el radio del buffer se puede asociar el punto a alguna calle.

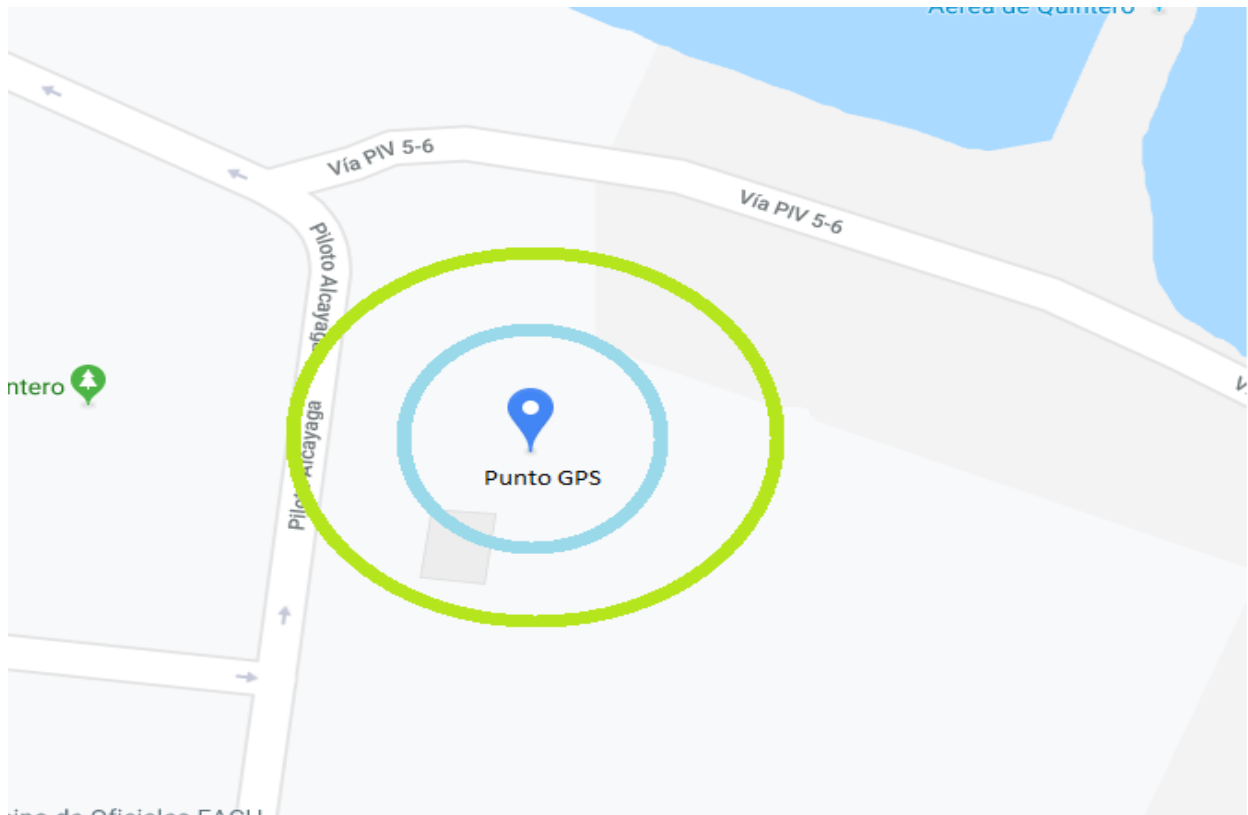


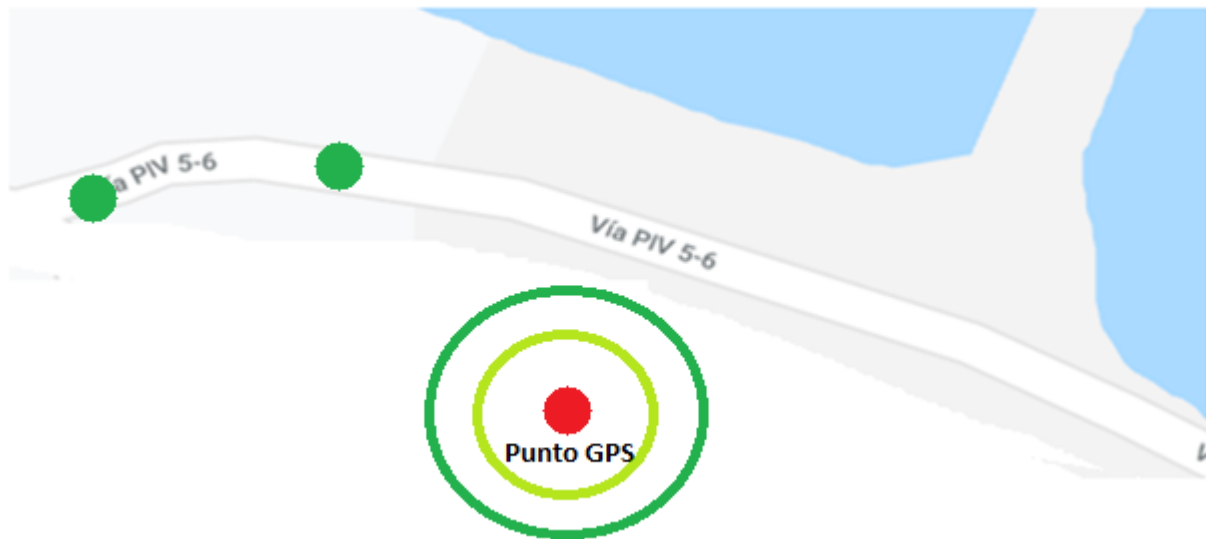
Figura 5: Imagen editada de Google Maps: Elaboración propia



Al momento de no encontrar ningún segmento de vía a través del radio dinámico ya que llegó al límite máximo (que el usuario escoge e ingresa como parámetro del algoritmo), el punto GPS quedaría desasociado, y no forma parte de la solución de la ruta. Entonces ahí sería cuando entraría el forzado de puntos GPS para encontrar esos puntos que no están siendo asociados a ningún segmento de vía y así asignarlos al punto cercano a lo largo de una vía. Por ejemplo, en la Figura 5.1, el punto rojo representa el punto GPS que aún no está asociado a la calle y los puntos verdes son los que ya han sido asociados a la vía. Además, el punto cuenta con dos círculos a su alrededor donde el círculo de color verde claro es el radio estático, que luego este radio es aumentado hasta un cierto límite inmediatamente sin pasar por otros valores entre medio y este es representado por el círculo de color verde oscuro. Entonces la mejora del radio dinámico no es suficientemente grande para alcanzar el snapeo del punto GPS a alguna vía. Por lo tanto, el forzado se ejecuta y el punto GPS que iba a quedar desasociado se asociará al punto alineado perpendicularmente a la vía más cercana.

Esta idea nació de varios estudios anteriores de cada reunión con mi profesora guía Carola Blázquez L. ya que la solución del radio dinámico dejaba puntos desasociados y se tomó esta medida de forzar estos puntos a la vía más cercana que componen la ruta final como solución.

En resumen, la idea es agregarlos a la fuerza a la calle con respecto al último punto anterior que haya sido asignado a alguna calle y ese punto debiese haber sido evaluado por todas las condiciones, asegurando que los puntos que estén hacia atrás estén correctos.

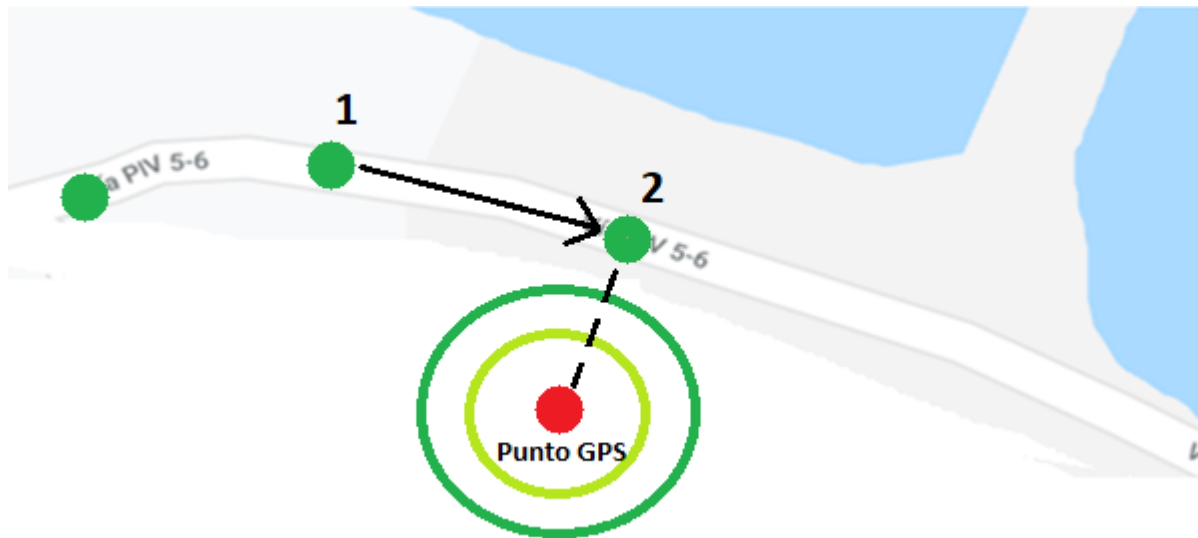


**Figura 5.1: Imagen editada de Google Maps forzado: Elaboración propia**

Una tercera mejora implementada es el forzado de puntos GPS después de asociarse al segmento de vía que debe ser ruteado<sup>2</sup> y determinar si el punto forzado es una

<sup>2</sup> Función con base algorítmica de Dijkstra que retorna la distancia y velocidad promedio entre dos puntos y ve si cumple los criterios de tolerancia de velocidad y heading.

solución viable a la ruta. Por ejemplo (Ver Figura 5.2), siguiendo la idea de la mejora anterior, una vez que el punto es asociado a la vía, se debe verificar que ésta sea la calle correspondiente o resultado de la ruta. Esto se verifica a través del ruteo entre el punto número 1 que está asociado y el punto forzado, numero 2, donde si logra cumplir con el criterio de tolerancia de velocidad, se acepta la ruta, en caso contrario, se rechaza la ruta y se buscan otros candidatos de segmentos de vías al cual se puede asociar el punto GPS.



**Figura 5.2: Ejemplo de un punto forzado al segmento de vía más cercano y posterior ruteo: Elaboración propia**

La cuarta mejora al algoritmo map-matching es la incorporación del heading ya que se vieron resultados positivos, pero no del todo ya que se obtuvieron casos particulares en que puntos forzados con ruteo lograban ser aceptados por el criterio de la tolerancia de velocidad, quedando asociados a calles incorrectas. Por eso, en una reunión con mi profesora guía Carola Blázquez se discutió que para llegar a una mejor solución (aumentando la calidad de la solución), se agregó la condición del heading en aceptación a los puntos una vez que se haga el ruteo entre pares de puntos. Esta condición sería por medio del sentido y dirección de cada punto GPS versus las dos direcciones de la calle, como se muestra en la Figura 5.3. En esta figura, el punto GPS cuenta con un valor en grados referente al norte magnético que indica en qué sentido viaja el vehículo y la vía que se va a asociar cuenta con dos ángulos referente al norte magnético. Por lo tanto, el ángulo del punto GPS y los dos de la vía se comparan con el parámetro de una tolerancia del ángulo, con el fin de evaluar si el punto es aceptado. Finalmente, un punto GPS es aceptado tanto por los criterios tolerancia de velocidad y heading del punto.

La tesis de Ponce (2010) plantea, que a través de un punto GPS que se asocie a la calle, sumándole un metro en la dirección de su eje "y" hacia al norte y posteriormente

proyectar este mismo punto a la calle nuevamente para formar con ello un triángulo rectángulo, como se aprecia en la “imagen 2” de la Figura 5.3 que puede ser aplicado esta técnica en distintas direcciones de las calles como es en el caso “A”, “B” y “C”.

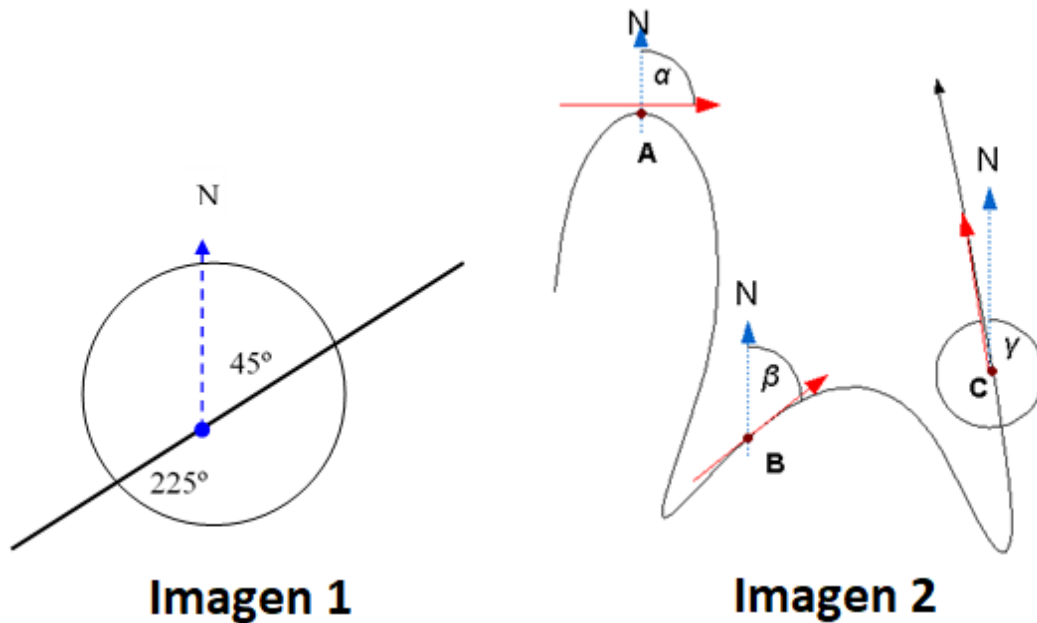


Figura 5.3: El ángulo acimut de un arco para distintas direcciones Fuente: Ponce (2010)

Siguiendo con la revisión bibliográfica, se investigó una tesis anterior de un alumno egresado de la Universidad Andrés Bello, Álvaro Ponce, que menciona que a través de un triángulo rectángulo como el de la Figura 4.3 y la fórmula  $\omega = \arccos\left(\frac{B^2 + C^2 - A^2}{2 * B * C}\right)$

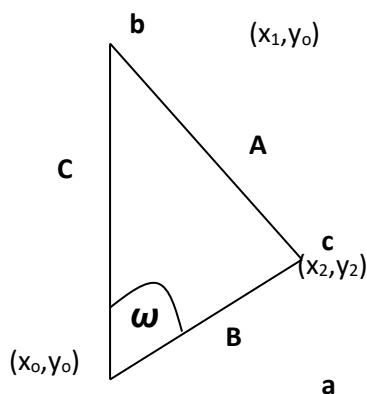


Figura 4.3: Triángulo que contiene el ángulo auxiliar de un arco y coordenadas de los vértices.

Se puede obtener a través de esto, el ángulo en los dos sentidos de una calle y con eso poder compararlo con el sentido del punto GPS.

### Implementación del código:

Todos los códigos implementados en este estudio se encuentran respaldados en GitHub: <https://github.com/nachozamorano/Map-matching>.

Luego de efectuar el estudio y la presentación de la identificación de mejoras, se continuará a efectuar la programación en el algoritmo base creando distintas versiones generadas, utilizando como lenguaje principal Python 2.7 y siendo visualizado en ArcGIS 10.01.

Cuando se buscan vías alternativas para K<sub>j</sub> o K<sub>i</sub> y no se llegase a encontrar ninguna, entonces se aumenta el radio del buffer para intentar encontrar alguna calle al cual asociar el punto GPS. Esta mejora del buffer dinámico es implementada dentro de la función mapMatch ()<sup>3</sup> en la condición “if not alt\_j” para K<sub>j</sub> y “if not alt\_i and not alt\_j” para K<sub>i</sub>.

A continuación, se presenta en la Figura 5.4, el funcionamiento del radio dinámico del buffer en el algoritmo. Primero, se inicia normalmente buscando una ruta factible entre los dos puntos k<sub>i</sub>, k<sub>j</sub> (k<sub>i</sub> -> k<sub>j</sub>). Si no existe una ruta factible, entonces se buscan nuevas ubicaciones alternativas de estos puntos con respecto a K<sub>j</sub>. Si no se encuentran alternativas, entonces se verifica si el radio del buffer es menor al límite máximo del radio dinámico. sí es afirmativo, entonces se aumenta el radio en un determinado valor (por ejemplo 1 o 2 o 3 pies o metros). Luego, se vuelven a buscar alternativas con respecto a k<sub>j</sub>. Se aplica la misma lógica para el punto k<sub>i</sub> cuando no se encuentren alternativos, es decir, se aumentará el radio, si éste es menor al límite máximo, entonces se aumentará el radio buscando nuevamente ubicaciones alternativas para el punto k<sub>i</sub>.

---

<sup>3</sup> Función principal del algoritmo que recoge la mayoría de los parámetros y toma la decisión de validar los puntos GPS a condiciones de cual calle debe ser asignado formando la ruta resultante.

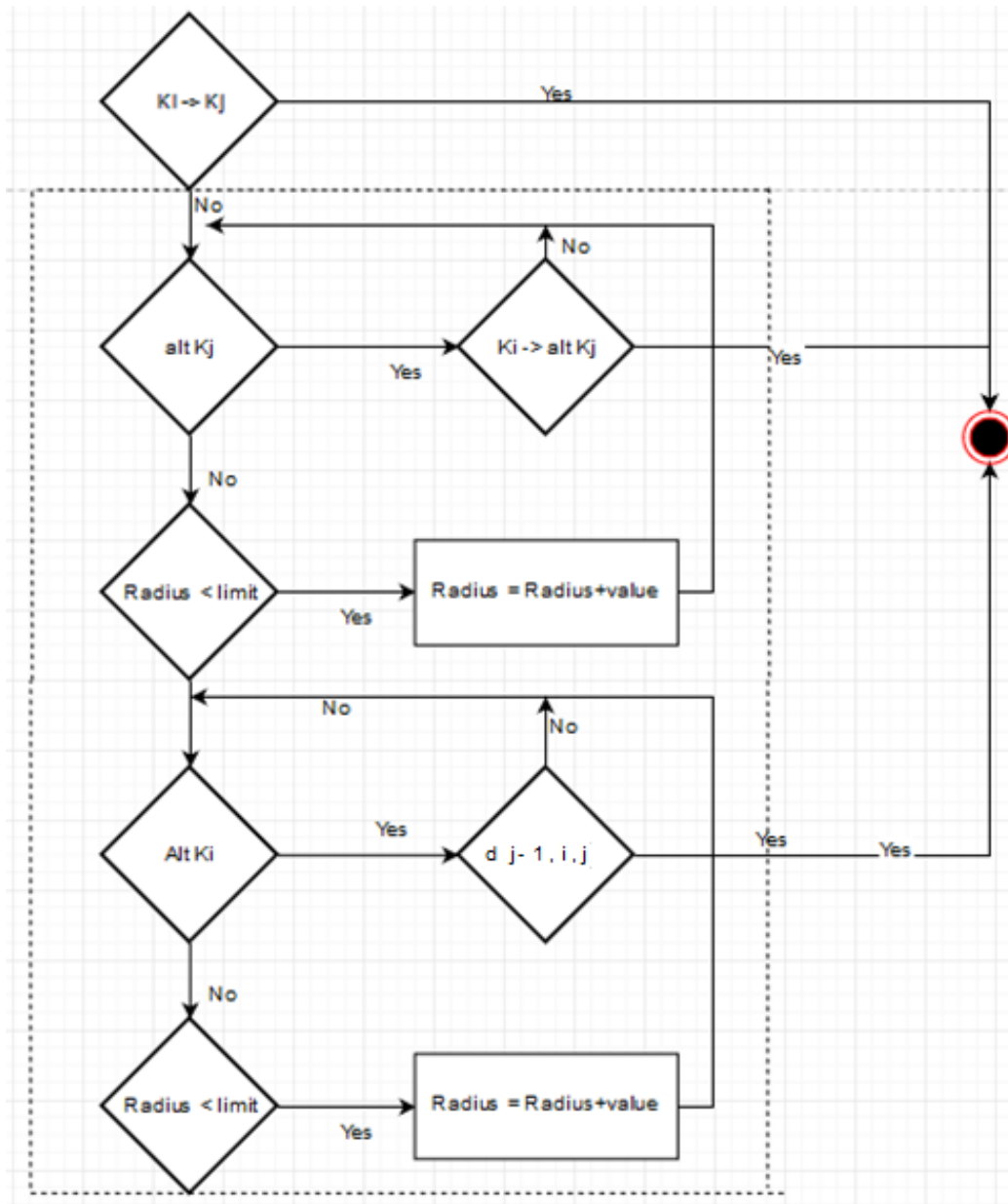


Figura 5.4: Diagrama algoritmo radio dinámico. Fuente: Elaboración propia

En principio, se implementa el forzado de los puntos GPS a sus respectivos segmentos de vías más cercanos cuando se ejecute la función `mapMatch()`, donde se verifica a través de una condición “`gps_j in SnapDict`”, que quiere decir sí, con el radio base no se encuentran puntos alternativos en `snapDict`<sup>4</sup> para ese `gps_j`<sup>5</sup> en calles cercanas, entonces entra a la condición “`gps_j in SnapDict`”, busca ubicaciones alternativas para los puntos en calles cercanas al punto GPS con el límite máximo del radio dinámico pasando a la siguiente condición “`SnapDict=[]`”. Que verifica nuevamente si con el límite

<sup>4</sup> Es un arreglo que contiene el id de las calles posibles al que puede asociarse los punto `ki` o `kj`.

<sup>5</sup> Punto GPS que esta sin ser asociado a una calle

máximo del radio dinámico existen alternativas. Si no llegase a encontrar, entonces entra a la condición “SnapDict= []” y se fuerza el punto GPS a la vía que estaría quedando fuera de la solución con respecto al punto anterior que ya fue asociado a la calle y es parte de la solución, como muestra el diagrama del algoritmo que fuerza los puntos a segmentos de vías en la Figura 5.5.

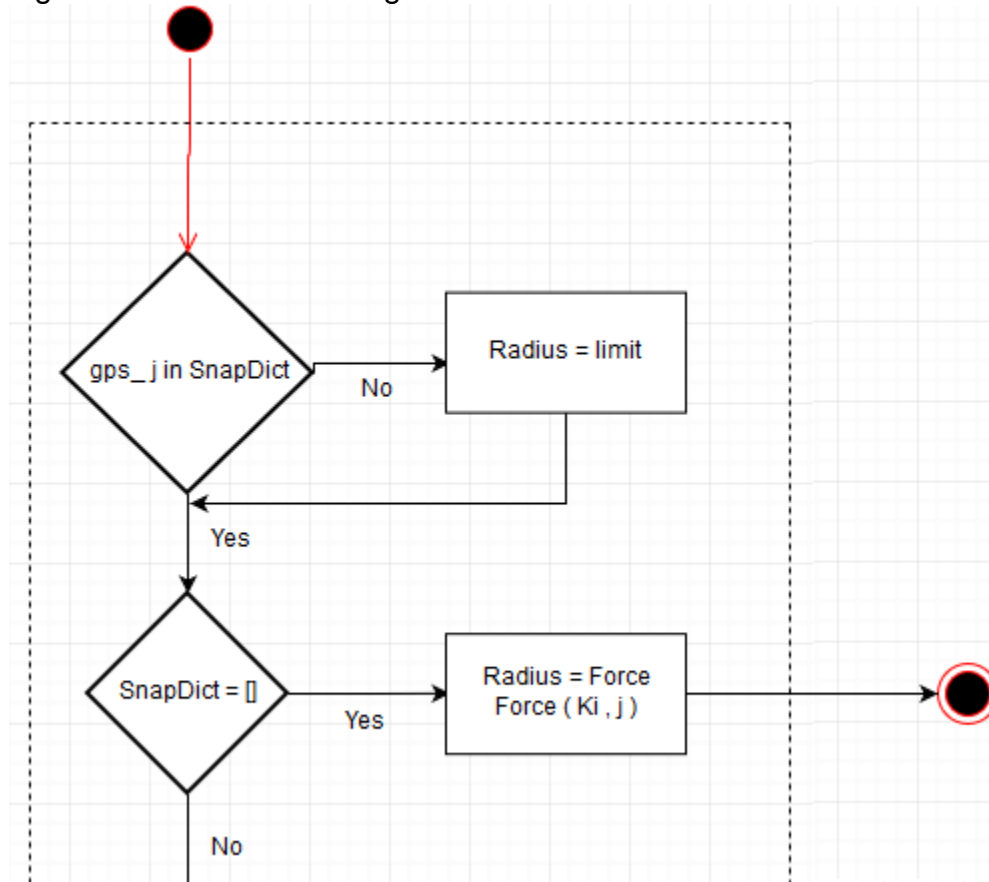


Figura 5.5: Diagrama algoritmo forzado de puntos GPS. Fuente: Elaboración propia

Con respecto al forzado de puntos con ruteo que es una nueva mejora a algo ya planteado que es lo mismo que el forzado anterior, pero esta vez con la mejora que verifica si la velocidad del punto anterior asociado a la calle y el punto forzado se puede aceptar bajo una tolerancia de velocidad y con esto verificar si entre estos dos puntos da una velocidad del vehículo. Por ejemplo, 30 km/h entre punto y punto y no 1000 km/h y pueda ser aceptado como solución del punto dentro de una velocidad razonable. Esto se realiza dentro de la condición “SnapDict= []”, como se muestra en el diagrama de la Figura 5.6. Cuando se encuentra la opción de forzar, se dirige a la parte principal de algoritmo que es el ruteo y determina si acepta o no la ubicación actual del punto. Si no se acepta, entonces se sigue con el resto del algoritmo.

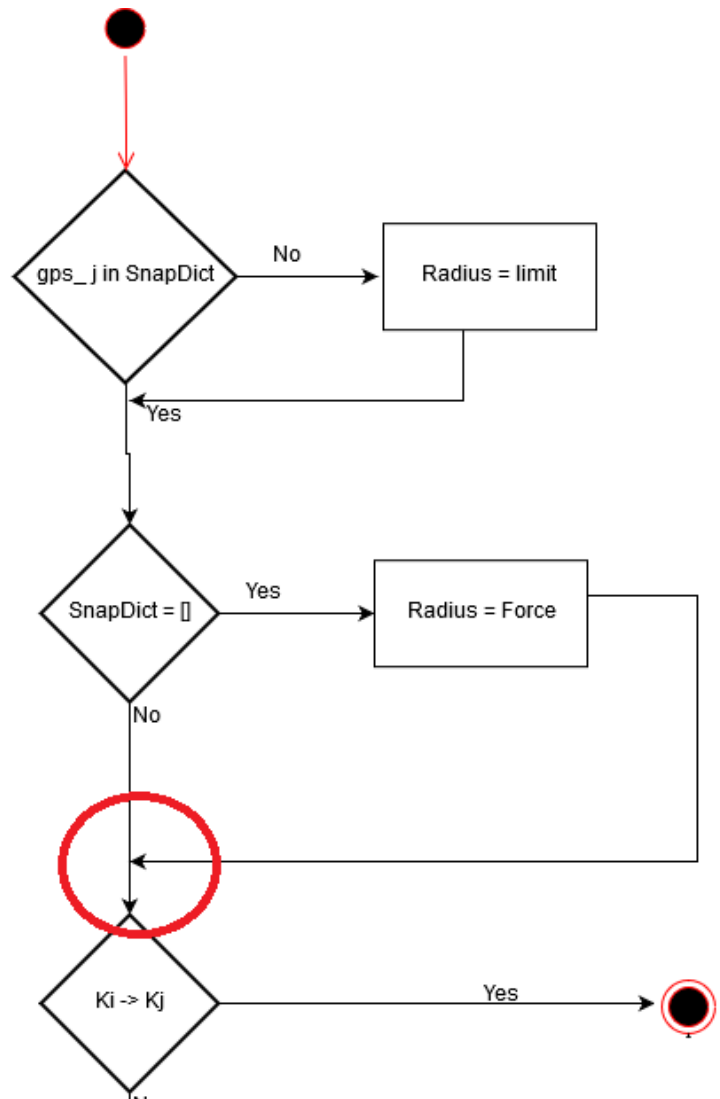
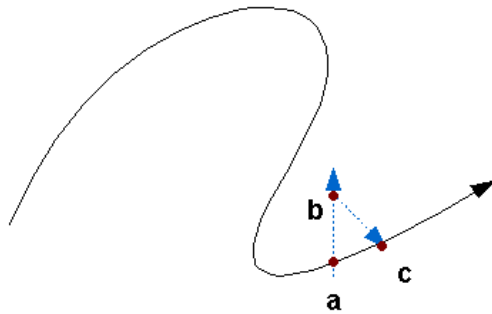


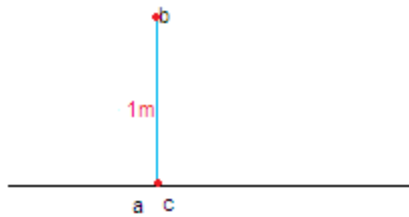
Figura 5.6: Diagrama algoritmo forzado con ruteo. Fuente: Elaboración propia

La condición del heading fue aplicada en cada punto de testeo y con dos funciones llamadas “heading ()” y “point\_c ()”. Esto inicia cuando se llama la función principal que es “heading ()”. Esta función está encargada de encontrar los dos ángulos de los sentidos de las vías. Como se explicó anteriormente, primero, se asocia el punto GPS a la calle que sería el punto “A”, luego se aumenta en un metro con respecto al norte magnético encontrando el punto “B” y por último con la función “point\_c” se obtiene el punto “C” que es el punto “B”, pero proyectado nuevamente sobre la calle. El objetivo de estos pasos es encontrar un triángulo rectángulo al igual como se observa en la Figura 5.6.1, y así calcular los ángulos para los dos sentidos de la calle con la arco tangente del lado “A”, es decir, (lado “C” / lado “B”) según fórmula matemática básica y además obtener el ángulo opuesto para los dos sentidos de la calle. Ahora verificando igualmente cómo se comporta en curvas, es tan diminuto el triángulo rectángulo que se

forma en un segmento de la curva que el algoritmo lo toma como recta dando el ángulo correcto.



**Figura 5.6.1: Triángulo empleado para obtener el ángulo acimut del arco en el punto “a”: Ponce, 2018**  
Existen dos casos especiales: si el punto “C” concuerda con el punto “A” tal como se puede divisar en la Figura 5.6.2. Cuando la calle es horizontal, el ángulo oscila entre  $0^\circ$  y  $180^\circ$ , Cuando la calle es vertical, entonces el punto “C” concuerda con el punto “B”, por lo tanto, el ángulo oscilaría entre  $90^\circ$  y  $270^\circ$ .



**Figura 5.6.2. Ejemplo caso especial del ángulo auxiliar: Ponce, 2018**  
Al obtener los dos ángulos de la calle, se realiza la condición de heading posterior al ruteo entre pares de puntos. Al igual que la velocidad, existe una tolerancia de aceptación de grados con respecto al sentido angular del punto snapeado versus los dos ángulos de la calle. El sentido del vehículo (heading) es capturado en cada medición GPS.



## Preparación y selección de datos y testing:

Al momento de haber efectuado el punto de la implementación de código armar un set de datos y testear en base a estos para la obtención de los resultados particulares de los casos.

### 1. Portage

Primero, se utilizó un set de datos para el radio dinámico que corresponde al Condado de Portage en Wisconsin, EE. UU. con un total de 170 puntos, con una frecuencia de captura de datos de 10 segundos. En la Figura 5.7 se muestra una parte de estos 170 puntos. Los datos en Portage son de excelente calidad ya que son DGPS y los datos que almacena cada punto maneja incluso el error del GPS.

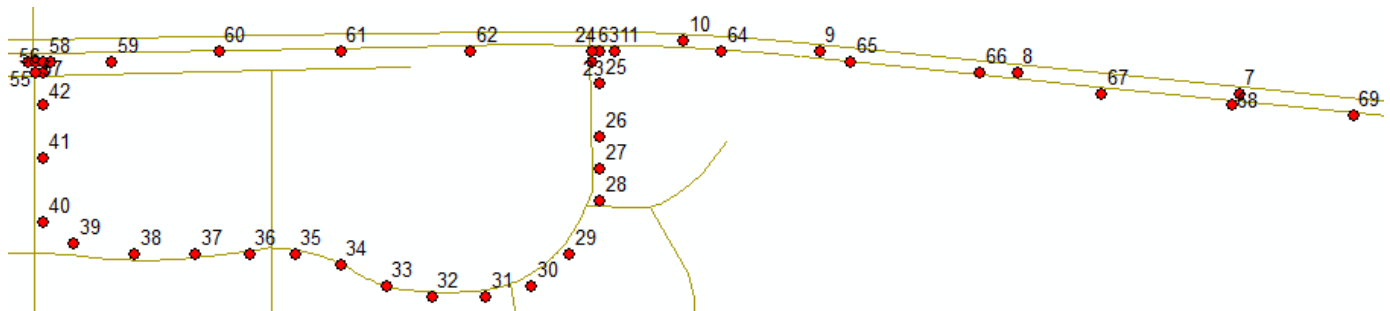


Figura 5.7: Data de 170 puntos DGPS Portage, EE. UU: ArcGIS 10.1

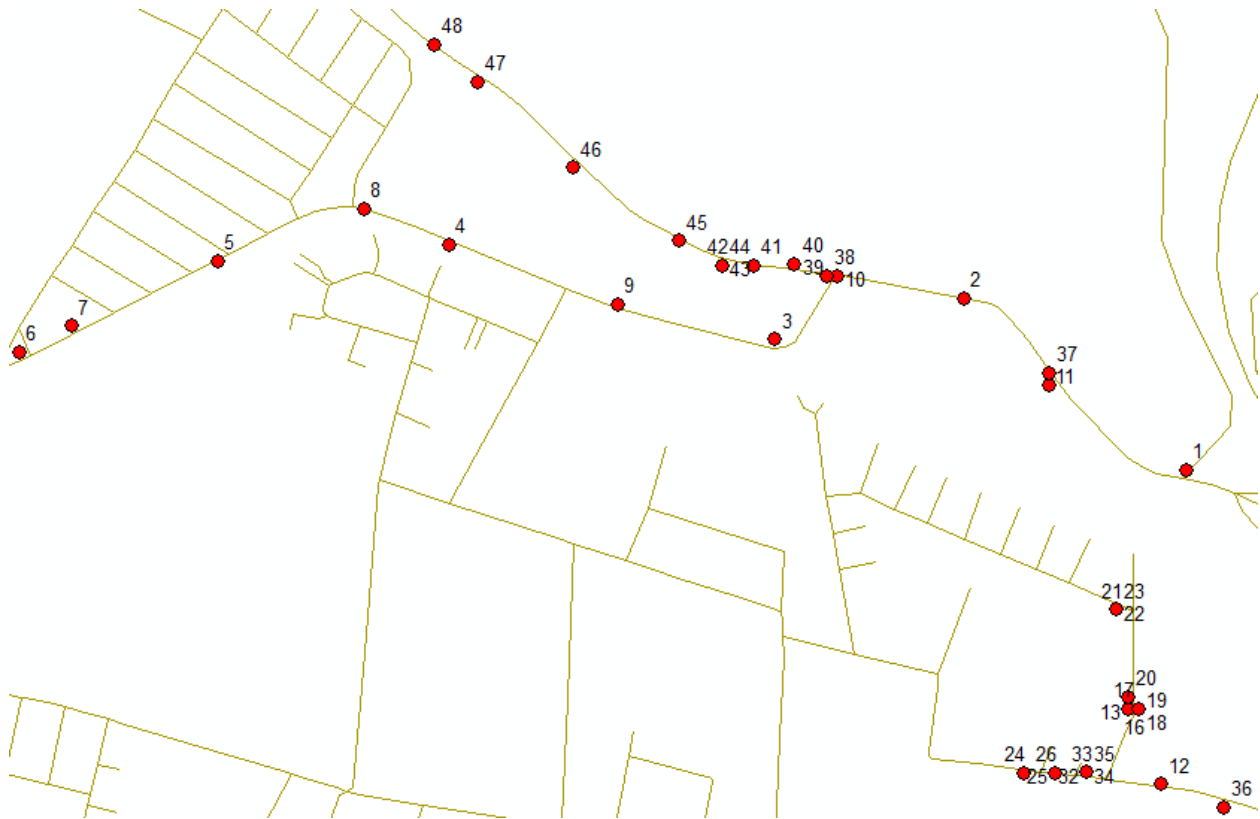
### 2. Renca

El algoritmo fue modificado para que pudiese funcionar con el sistema métrico. Por ejemplo, de pies a metros, millas/horas a kilómetros/hora, etc. Además, el sistema de coordenadas en el set de Renca es WGS84, por lo tanto, éste se proyectó a UTM, incluso se tuvo que cambiar el formato Unicode de cada velocidad a Integer que es lo que maneja el código.

Se proyectó a este nuevo sistema de proyección porque no se manejaba mucha información con Portage mientras que en Renca sí y podría hacer más comparaciones de resultados e incluso más desafiantes para el algoritmo porque son GPS, es decir de peor calidad que los DGPS, como por ejemplo más distantes los puntos a las calles o incluso datos nulos.

Posteriormente, se creó un set de datos de la comuna Renca, Chile de un total de 48 puntos. Los datos en Renca son de menor calidad comparado con el set anterior ya que son GPS y los datos que almacena cada punto contiene varios nulos. Por ejemplo, en las velocidades varios contienen el valor 0Km ya que los datos corresponden a vehículos recolectores de residuos sólidos domiciliarios y lo más probable es que se detuvieron a recolectar en varias ocasiones deteniendo el vehículo. Sin embargo, se trató de elegir la mejor parte del set completo de 425 puntos, considerando el menor de paradas del

camión recolector. La imagen de la data de los 48 puntos GPS para llevar a cabo la creación de la mejora del forzado sin ruteo se muestra en la Figura 5.8.



**Figura 5.8: Data de los 48 puntos GPS Renca, Chile: ArcGIS 10.1**

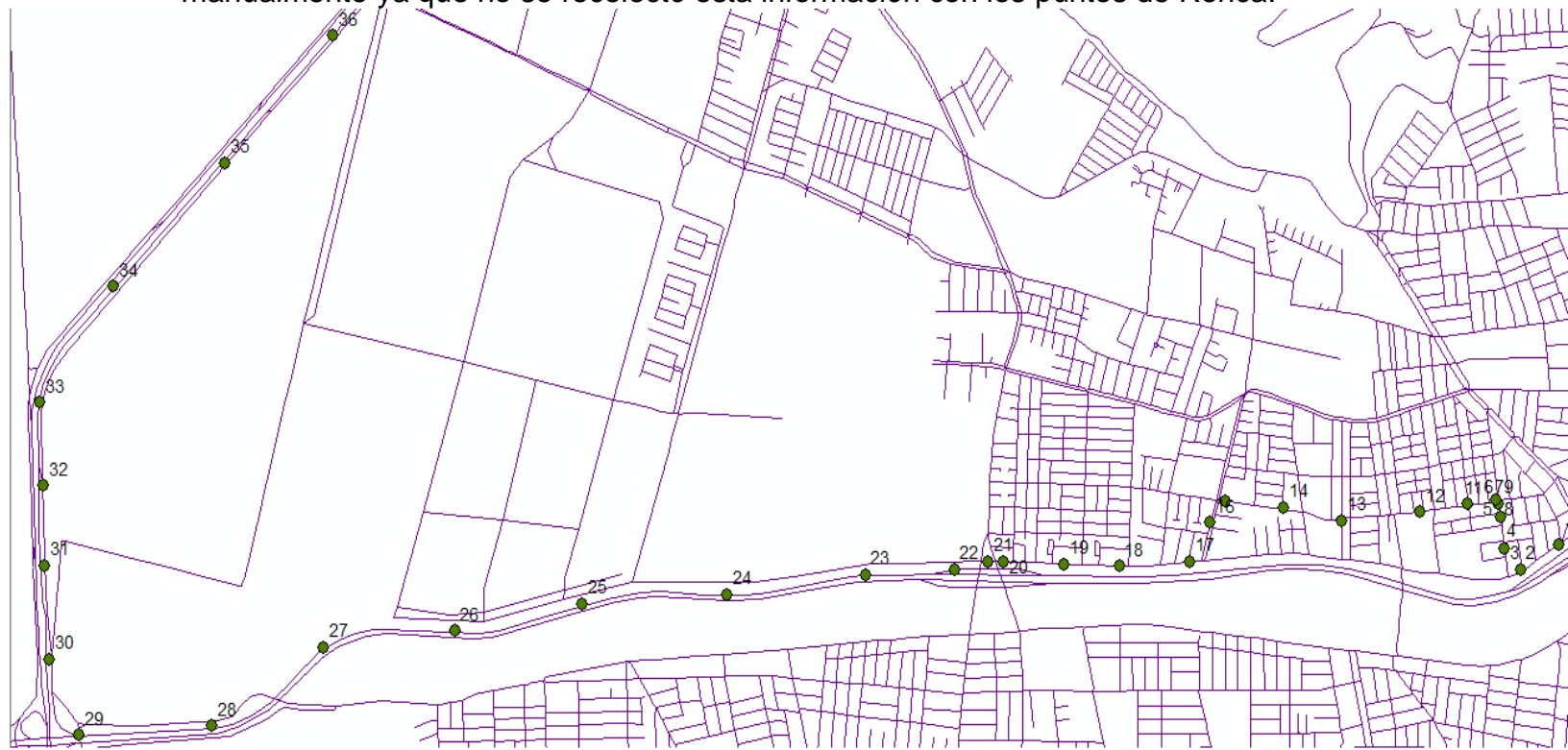
En el caso del forzado con ruteo, se generó y trabajó también con un set de datos ubicado en Renca, Santiago Chile. La cantidad de puntos del data set es de 22 puntos que siguen siendo obtenidos por tipo GPS (Ver Figura 5.9).



**Figura 5.9: Data de los 22 puntos GPS Renca, Chile: ArcGIS 10.1**

Para la mejora relacionada con el heading, se usó un set de datos de 36 puntos ubicado en Renca, Santiago Chile, donde lo más desafiante que se vio para la elección de este set es que debía contener calles cercanas al eje vertical, horizontal y contenga curvas, porque lo más desafiante para el heading son los dos casos especiales calles

horizontales, verticales y que contengan curvas ya que el ángulo puede variar harto (Ver Figura 5.9.1). Los grados del sentido por cada punto fueron agregados manualmente ya que no se recolectó esta información con los puntos de Renca.



**Figura 5.9.1: Data de los 36 puntos GPS Renca, Chile: ArcGIS 10.1**

Mejora de la estructura del algoritmo:

GitHub: <https://github.com/nachozamorano/Map-matching/tree/Version-Modulado>

Se reestructura el código resultante de las mejoras efectuadas para un mejor entendimiento ya sea siendo comentando, ordenando o modulando. Se moduló la función que lleva el mayor proceso y el centro de todo que es mapMatch (). Modulando cada una de sus funciones en subfunciones para tener un mayor orden en el código y cada función tenga su única tarea, pudiendo en un futuro saber si haya mayor tardanza en el resultado que función se lleva el mayor peso.

Agregando, se comentó las mejoras efectuadas.

Análisis de resultados:

Se efectuará un análisis de resultado a gran escala en base a los puntos correctamente asociados a la red vial de set de datos correspondiente y al tiempo de tardanza del algoritmo base (algoritmo final de cada hito) y el modificado ya que estos recursos son unos de los más influyentes en el sistema MMA.

Comparación entre la versión actual y versión mejorada del MMA:

Finalmente, se efectuará un estudio comparativo entre los dos casos de la versión base y modificada, evaluándose en mayor profundidad, relatando la raíz del por qué hubo mejora o no hubo mejora con el fin de llegar a una conclusión.

## Resultados

Los resultados obtenidos durante este proceso de avances se han generado en base a cuatro mejoras propuestas que son:

- Radio dinámico (Portage, EE. UU.).
- Forzado de puntos sin ruteo (Renca, Chile.).
- Forzado de puntos con ruteo (Renca, Chile.).
- Heading (Renca, Chile.).

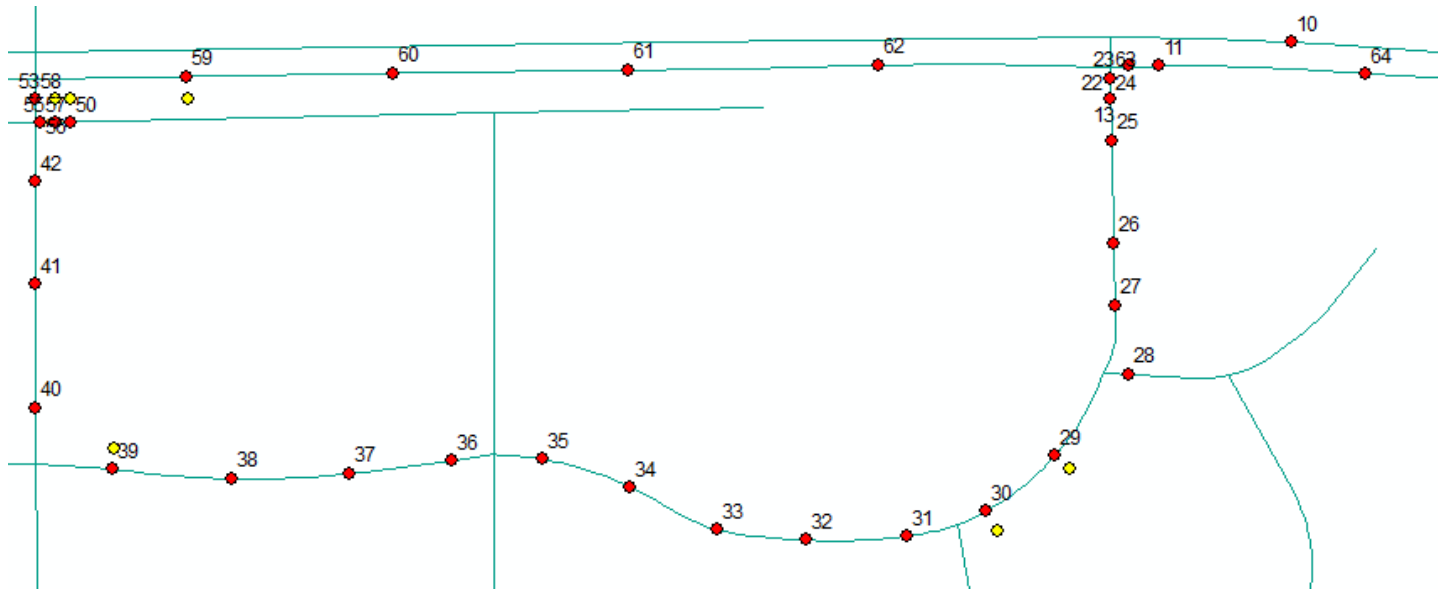
A continuación, se explican cada una de estas cuatro propuestas vistas anteriormente en la mejora de rendimiento en el algoritmo MMA (Map-matching).

Radio dinámico (Portage, EE. UU.):

GitHub: <https://github.com/nachozamorano/Map-matching/tree/Version-Radio-Dinamico>

Como se muestra en la Figura 6, esta mejora está ambientada en el Condado de Portage en Wisconsin, EE. UU.

Primero, se compararon dos algoritmos (base y mejora heading): uno con radio dinámico que consiste en el aumento de 35 pies a 45 pies mostrado con “color Rojo”, el otro con un radio estático de 35 pies mostrado con “color Amarillo”. En la Figura 6, se logra visualizar claramente que los puntos rojos que pertenecen al radio dinámico se relacionan mejor a la red vial formando una ruta casi perfecta ya que hay algunos puntos que se relacionan incorrectamente a otras vías, donde lo que se espera que se arreglen con la restricción de la dirección de la calle. Mientras que el resultado con el uso del radio estático, hay puntos que no alcanza a ser asignados a ninguna vía, quedando por fuera de la red vial.



**Figura 6: Comparación de los resultados sin y con la implementación del buffer dinámico Fuente: Elaboración propia**

Las Figuras 6.1 y 6.2 son el resultado en un sentido más numérico y en relación con la solución real del problema.

El total de puntos GPS analizados son “170”, de los cuales 136 puntos fueron asociados usando un buffer estático y 152 puntos asociados con un buffer estático y dinámico, dando en porcentaje un 9,4% de mejora en puntos asociados. Sin embargo, no quiere decir que se asociaron bien. La última variable para compararse sería el tiempo, ya que el estático se demoró 7,696 minutos  $\approx$  8 minutos en obtener una solución mientras que el dinámico se demoró 9,481  $\approx$  9 minutos. Cabe destacar nuevamente que la Figura 6.2 con radio dinámico fue medido de 35 pies a 45 pies mientras que el radio estático (Ver Figura 6.1) solo un radio de 35 pies.

La diferencia de tiempo fue generada porque el radio estático no alcanzaba a encontrar o interceptar calles a su alrededor, por lo cual no se encontrarían puntos alternativos para que estos puntos GPS sean añadidos a la calle, quedando como resultado puntos sin ser asociados a una calle y que el algoritmo los deje fuera de la solución sin insistir nuevamente en poder añadirlos y no seguir añadiendo tiempo al proceso, mientras que en el dinámico requiere más procesos como por ejemplo el aumentar gradualmente el radio hasta que encuentre alguna calle o hasta que llegue al límite máximo del tamaño del buffer. Por lo tanto, al tener mayores procesos y más puntos analizados que se abarquen mayor es el cómputo de tiempo.

```
{10: {1: 170}}  
{10: {15: [('35 Feet', 1, 136, 7.696)]}}
```

Figura 6.1: Radio Estático: Terminal

```
{10: {1: 170}}  
{10: {15: [('35:45 Feet', 1, 152, 9.481)]}}
```

Figura 6.2: Radio Dinámico: Terminal

Tras haber efectuado variadas pruebas, se vio que el radio dinámico en el ki no era necesario ya que se resolvía cuando el kj aumentaba, donde no arreglaba ningún punto solo lograba que tuviese un mayor tiempo de cómputo el algoritmo quedando de la siguiente forma (Ver Figura 6.3).

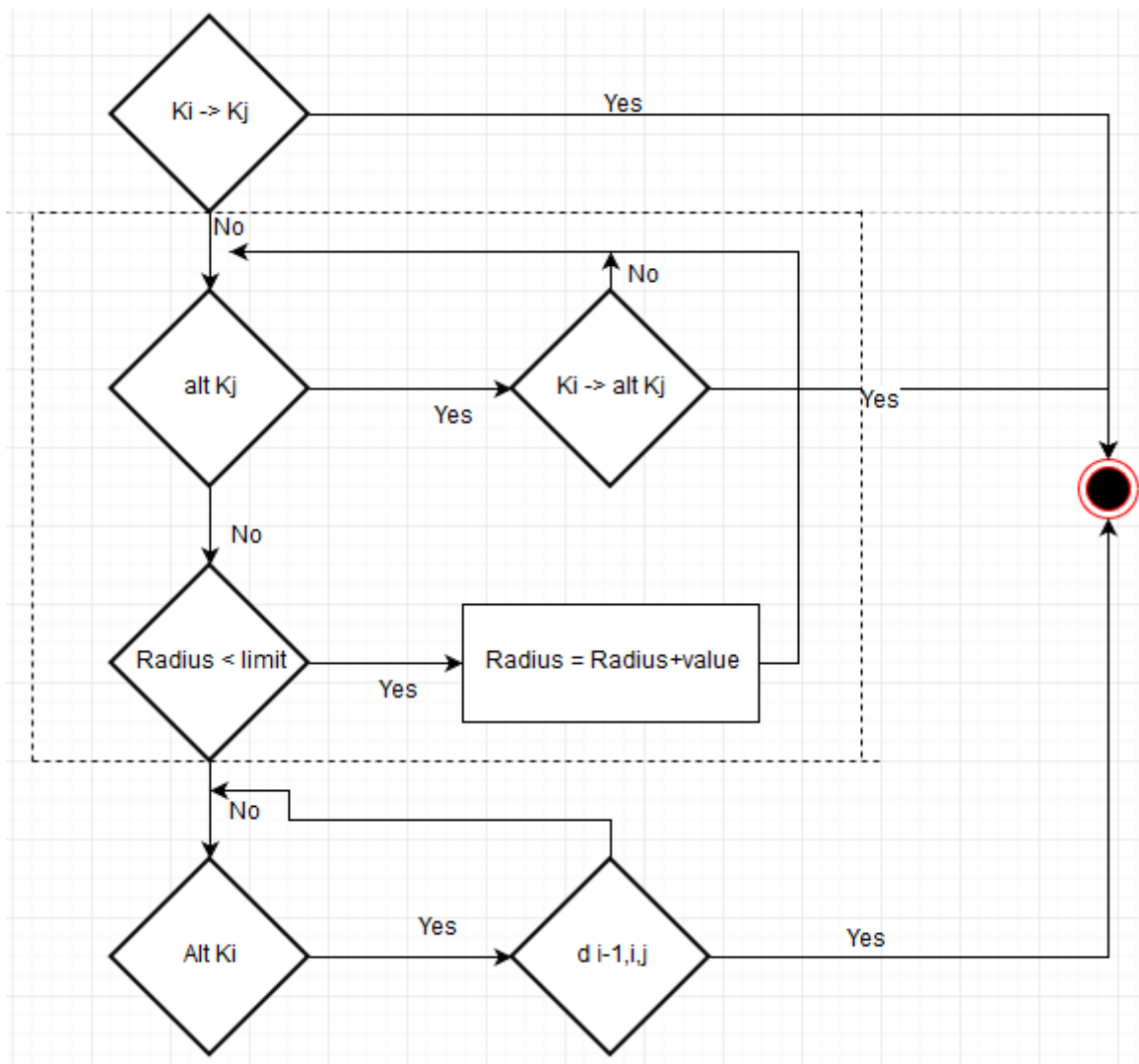


Figura 6.3: Radio Dinámico: Elaboración propia

Forzado de puntos sin ruteo (Renca, Chile.):

GitHub: <https://github.com/nachozamorano/Map-matching/tree/Version-forzado>

En esta otra mejora, se decidió cambiar el set de datos a la comuna de Renca en Santiago, Chile por el hecho que existe más información recolectada tal como la dirección (heading) de los vehículos. El cambio requirió cambiar las medidas a sistema métrico.

Otro punto que es integrado es el forzado de puntos que nace de la idea, por ejemplo, de la Figura 6.2 que, a pesar de usar un radio dinámico, existen 28 puntos que no fueron asignados de los 170 puntos total, es decir, cuando el radio dinámico no alcanza a interceptar ninguna calle quedan puntos sin ser asignados.

Cada punto GPS cuenta con un id propio, pero va ordenado de inicio a fin del trayecto del vehículo de forma gradual (1, 2, 3, etc.). Por lo tanto, todo este forzado se vio ya que se encontró esta lógica de sucesión con el id de los puntos GPS por las calles y la idea es la sucesión de lo que se vio en (Bang, Kim, & Yu, 2016) con el radio dinámico y mejorar en las falencias que se pueden encontrar en el radio dinámico y en conversaciones con mi profesora guía Carola Blázquez y en estudios anteriores.

El funcionamiento del forzado sin ruteo consiste en que tiene por defecto un valor para el radio del forzado si ruteo que incluya todos los puntos GPS analizados al menos a una o varias alternativas para ser asignado a la red vial. En este caso, para 48 puntos en la comuna de Renca se utilizó un valor de 11 metros, donde con 11 metros todos los puntos GPS al menos pueden ser asociados a una calle. Durante todo el proceso del algoritmo, se verifica inicialmente si haya algún punto que no esté siendo asignado a alguna calle logrando buscar las calles posibles cerca de ese punto por el radio del forzado sin ruteo, donde se fuerza a la calle referente al punto anterior que fue insertado correctamente y es parte de la solución.

Las tres imágenes siguientes representan el tiempo del algoritmo, un radio estático inicial de 4 metros en la Figura 6.4, un radio estático inicial de 5 metros en la Figura 6.5 y en el mejorado un radio dinámico de 4 a 5 metros y un forzado de 11 metros en la Figura 6.6.

El tiempo en que se demoró el algoritmo original de radio de 4 metros fue de 1.346 minutos, el de algoritmo original con un buffer de 5 metros fue de 1.589 minutos y el algoritmo mejorado con puntos forzados fue de 1.982 minutos siendo casi iguales los 3 resultados separándolos algunos segundos.

```
{10: {1: 48}}  
{10: {30: [('4 Meters', 1, 0, 1.346)]}}
```

Figura 6.4: Resultado Map-matching original radio 4: Terminal



```
{10: {1: 48}}  
{10: {30: [('5 Meters', 1, 0, 1.589)]}}
```

Figura 6.5: Resultado Map-matching original radio 5: Terminal

```
{10: {1: 48}}  
{10: {30: [('4 Meters to 5 Meters', 1, 0, 1.982)]}}
```

Figura 6.6: Resultado puntos forzados radio dinámico 4 - 5: Terminal

Otra comparación es la calidad de la solución obtenida de tal manera que se identifican los puntos que fueron asignados a la red vial.

En la Figura 6.7, existen tres atributos mostrados en las tablas que son:

OBJECTID: Es el id de cada punto GPS.

Shape: Es el tipo del objeto en este caso es un Point, es decir, punto.

FID: Es el id de la calle al cual está asociado el punto. Si es 0 es porque no se asoció a ninguna calle.

Por ejemplo, como se puede ver en la Figura 6.7 con el algoritmo original de radio de 4 metros fueron asignados 24 de 48 puntos (50% puntos asociados), en el algoritmo original de radio de 5 metros fueron asignados 35 de 48 puntos (72% puntos asociados) y en la mejora fueron asignados los 48 de 48 puntos (100% puntos asociados) a la red vial donde aumenta el radio de 4 a 5 metros de radio dinámico y forzado de 11 metros.

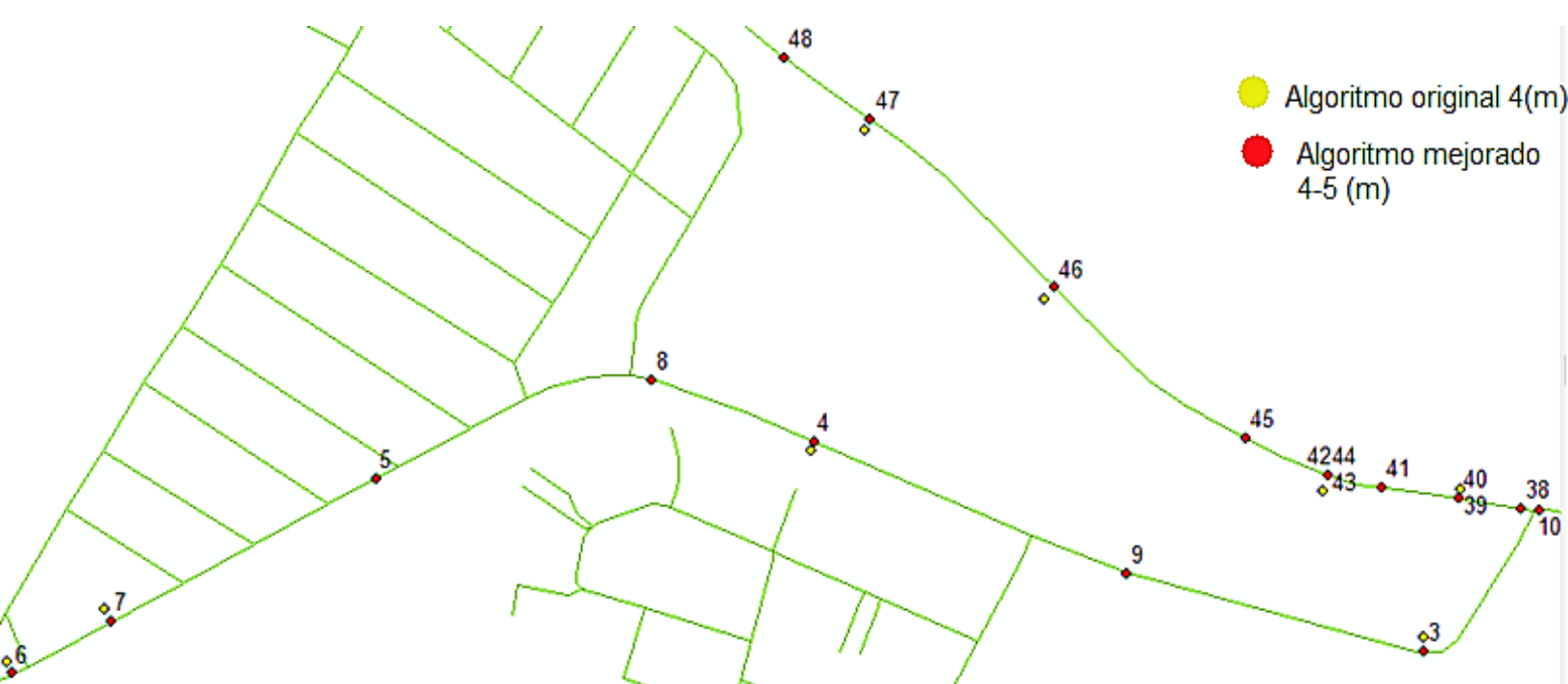
A pesar de que se haya demorado unos segundos logró con el objeto total de asignar el total de los puntos a las calles.

La mejora del buffer dinámico junto con el forzado sin ruteo entregó un 100% de los puntos asociados y en un poco mayor o casi igual de tiempo que los códigos originales separándolo algunos segundos, pero destacando que, a pesar de haber agregado dos mejoras, tenga más procesos y la separación de esos segundos de más, logre el objetivo que es solucionar la falencia del radio dinámico. Sin embargo, a pesar del 100% de la asociación por el forzado sin ruteo que soluciona falencias del radio dinámico, hay que destacar que probablemente varios de estos puntos forzados no hayan sido asociados a la calle acertada, ya que no se está verificando aún si está correctamente asociado bajo a la tolerancia de velocidad y para eso se debe comprobar que el punto asociado a la calle pertenece a la solución (ruteo).

Original radio 4			Original radio 5			Mejorado		
OBJECTID *	Shape *	FID	OBJECTID *	Shape *	FID	OBJECTID *	Shape *	FID
1	Point	4184	1	Point	4184	1	Point	4184
2	Point	1432	2	Point	1432	2	Point	1432
3	Point	0	3	Point	0	3	Point	1394
4	Point	0	4	Point	1376	4	Point	1376
5	Point	4193	5	Point	4193	5	Point	4193
6	Point	0	6	Point	0	6	Point	4194
7	Point	0	7	Point	0	7	Point	4193
8	Point	1376	8	Point	1376	8	Point	1376
9	Point	1394	9	Point	1394	9	Point	1394
10	Point	1432	10	Point	1432	10	Point	1432
11	Point	0	11	Point	0	11	Point	1432
12	Point	1520	12	Point	1520	12	Point	1520
13	Point	0	13	Point	1512	13	Point	1512
14	Point	0	14	Point	1512	14	Point	1512
15	Point	0	15	Point	1512	15	Point	1512
16	Point	0	16	Point	1512	16	Point	1512
17	Point	0	17	Point	1512	17	Point	1512
18	Point	0	18	Point	1512	18	Point	1512
19	Point	0	19	Point	1512	19	Point	1512
20	Point	0	20	Point	1512	20	Point	1512
21	Point	0	21	Point	0	21	Point	1475
22	Point	0	22	Point	0	22	Point	1475
23	Point	0	23	Point	0	23	Point	1475
24	Point	1510	24	Point	1510	24	Point	1510
25	Point	1511	25	Point	1511	25	Point	1511
26	Point	1511	26	Point	1511	26	Point	1511
27	Point	1512	27	Point	1512	27	Point	1512
28	Point	1512	28	Point	1512	28	Point	1512
29	Point	1512	29	Point	1512	29	Point	1512
30	Point	1512	30	Point	1512	30	Point	1512
31	Point	1512	31	Point	1512	31	Point	1512
32	Point	1512	32	Point	1512	32	Point	1512
33	Point	1512	33	Point	1512	33	Point	1512
34	Point	1512	34	Point	1512	34	Point	1512
35	Point	1512	35	Point	1512	35	Point	1512
36	Point	0	36	Point	0	36	Point	1520
37	Point	1432	37	Point	1432	37	Point	1432
38	Point	1373	38	Point	1373	38	Point	1373
39	Point	0	39	Point	1373	39	Point	1373
40	Point	0	40	Point	1373	40	Point	1373
41	Point	1373	41	Point	1373	41	Point	1373
42	Point	0	42	Point	0	42	Point	1373
43	Point	0	43	Point	0	43	Point	1373
44	Point	0	44	Point	0	44	Point	1373
45	Point	1373	45	Point	1373	45	Point	1373
46	Point	0	46	Point	0	46	Point	1373
47	Point	0	47	Point	0	47	Point	1373
48	Point	1373	48	Point	1373	48	Point	1373

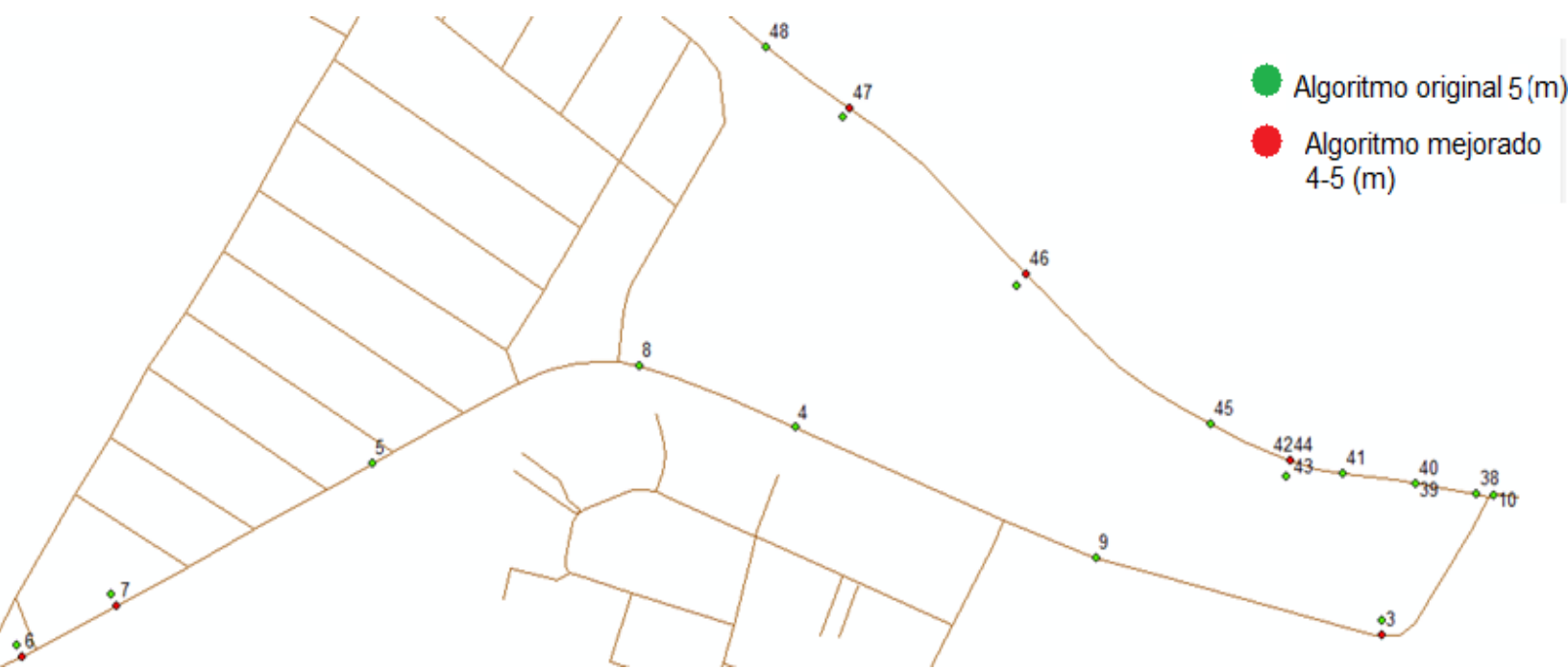
**Figura 6.7: Resultado versiones originales vs mejorado usando buffer dinámico: ArcGIS 10.1**

A continuación, se representa de otra forma las comparaciones que son plasmadas en un mapa. La primera comparación en la Figura 6.8 es del algoritmo original con un radio de 4 metros estático (color amarillo) versus a la agregación del forzado sin ruteo al algoritmo que contiene el radio dinámico (color rojo). Se puede observar claramente que el algoritmo mejorado relaciona más puntos a la red vial. La imagen es solo una parte del total de los 48 puntos.



**Figura 6.8: Resultado mapa algoritmo original radio estático de 4 metros vs mejorado radio de 4 a 5 metros**  
Fuente: Elaboración Propia

La segunda comparación en la Figura 6.9 es del algoritmo original con un radio 5 estático (color verde) vs al algoritmo mejorado (color rojo), viéndose nuevamente que la mejora logra asociar más puntos a pesar de que se aumentó el radio estático de 4 a 5 metros del algoritmo original. De esta manera, se cumple con el objetivo de esta mejora. La imagen es solo una parte del total de los 48 puntos.



**Figura 6.9: Resultado mapa algoritmo original radio estático de 5 metros vs mejorado radio de 4 a 5 metros:**  
ArcGIS 10.1

#### Forzado de puntos con ruteo (Renca, Chile.):

Los resultados a la nueva implementación efectuada al forzado con ruteo se comprobarán con el set de datos pertinente de 22 puntos y además con respecto al forzado anterior que es sin ruteo.

Al momento de ejecutar el algoritmo con la mejora del forzado con ruteo se obtuvo como resultado el 100% de puntos asociados a la red vial, con un tiempo de 1.391 min, obteniéndose que la asociación de los puntos a la red vial fue correcta.

OBJECTID *	Shape *	FID
1	Point	41936
2	Point	41940
3	Point	41942
4	Point	13543
5	Point	13287
6	Point	42011
7	Point	42048
8	Point	41959
9	Point	13057
10	Point	12893
11	Point	12559
12	Point	12560
13	Point	12893
14	Point	13057
15	Point	41958
16	Point	42052
17	Point	42011
18	Point	13287
19	Point	13543
20	Point	41942
21	Point	41939
22	Point	13766

```
{10: {1: 22}}  
{10: {30: [('1 Meters to 2 Meters', 1, 0, 1.391)]}}
```

**Figura 7: Resultado mapa algoritmo forzado con ruteo radio dinámico de 1 metro a 2 metros y 11 metros de forzado: ArcGIS 10.1**

Otra comparación importante es el forzado sin ruteo con el que contiene ruteo en base al set de datos anterior de 48 puntos y así poder obtener resultados con respecto a la calidad de la solución y tiempo de ejecución.

Los resultados de esta comparación se encuentran en la Figura 7.1 (con ruteo) y en las Figuras 6.6 y 6.7 que muestran las tabla mejorado" (sin ruteo). Como resultado, se asociaron el 100% de los puntos en los dos casos, de hecho, se llegó al mismo resultado en ambos casos, pero el tiempo de ejecución de los resultados fueron diferentes. Por ejemplo, el tiempo para el forzado sin ruteo fue de 1.982 minutos, mientras que el tiempo para el forzado con ruteo fue de 2.365 minutos, es decir, aproximadamente 23 segundos más.

OBJECTID *	Shape *	FID
1	Point	4184
2	Point	1432
3	Point	1394
4	Point	1376
5	Point	4193
6	Point	4194
7	Point	4193
8	Point	1376
9	Point	1394
10	Point	1432
11	Point	1432
12	Point	1520
13	Point	1512
14	Point	1512
15	Point	1512
16	Point	1512
17	Point	1512
18	Point	1512
19	Point	1512
20	Point	1512
21	Point	1475
22	Point	1475
23	Point	1475
24	Point	1510
25	Point	1511
26	Point	1511
27	Point	1512
28	Point	1512
29	Point	1512
30	Point	1512
31	Point	1512
32	Point	1512
33	Point	1512
34	Point	1512
35	Point	1512
36	Point	1520
37	Point	1432
38	Point	1373
39	Point	1373
40	Point	1373
41	Point	1373
42	Point	1373
43	Point	1373
44	Point	1373
45	Point	1373
46	Point	1373
47	Point	1373
48	Point	1373

```
{10: {1: 48}}
{10: {30: [('4 Meters to 5 Meters', 1, 0, 2.365)]}}
```

Figura 7.1: Resultado tabla de atributos del ArcGIS 10.1 usando algoritmo forzado con ruteo radio dinámico de 4 metro a 5 metros y 11 metros de forzado con Ruteo: ArcGIS 10.1 y Terminal

#### Heading (Renca, Chile.):

Los resultados de la implementación de la mejora del heading fueron un 100% de los 36 puntos (Renca, Chile) asociados a la red vial con un radio dinámico de 6 a 8 metros, y para el forzado con un buffer de 15 metros (Ver Figura 7.2). Con un tiempo de resultado de 2.851 minutos. Al momento de la ejecución, se puede divisar además que discrimina calles erróneas que son aceptadas por la tolerancia de velocidad, pero rechazadas por la tolerancia de heading<sup>6</sup>. Logrando posicionar el punto en una calle más lejana, pero en la acertada, explicándose más adelante en la Figura 7.4 donde, ocurrió con el punto 13 ya que estaba tan mal posicionado que se asociaba a cualquier calle cercana, pero con el heading se obtuvo un resultado correcto. Como conclusión, se obtuvo una ruta muy acertada ya que se revisó manual. El único punto en contra se podría dar en base a la razón del tiempo ya que hay muchas calles encontradas en un principio que no dan con el sentido de estas.

---

<sup>6</sup> Rango de grado mínima y máxima que logra condicionar si está dentro del rango la dirección del vehículo vs los dos grados de la calle.

OBJECTID *	Shape *	FID
1	Point	1738
2	Point	1756
3	Point	1742
4	Point	1714
5	Point	1700
6	Point	1700
7	Point	1700
8	Point	1700
9	Point	1700
10	Point	1687
11	Point	1691
12	Point	1699
13	Point	1707
14	Point	1698
15	Point	1689
16	Point	1720
17	Point	1747
18	Point	1751
19	Point	1749
20	Point	1746
21	Point	4100
22	Point	4100
23	Point	4101
24	Point	4101
25	Point	4101
26	Point	4101
27	Point	4101
28	Point	4101
29	Point	4101
30	Point	4102
31	Point	4102
32	Point	4102
33	Point	4102
34	Point	4150
35	Point	4150
36	Point	4151

```

{10: {1: 36}}
{10: {15: [('6 Meters to 8 Meters', 1, 0, 2.851)]}}

```

Figura 7.2: Resultado mapa Heading con radio dinámico de 6 metro a 8 metros y 15 metros de forzado: ArcGIS 10.1 y Terminal

En comparación con respecto al algoritmo anterior que sería el forzado con ruteo, corriendo el mismo set de datos de 36 puntos y las mismas condiciones de variables, se logró que se asociaron los 36 puntos a la red vial, como se puede ver en la Figura 7.3 y en un tiempo de 2 minutos. ¿Porque se compararon estas dos mejoras?, fue porque el de radio dinámico fue probado con otro set de datos del lugar de Portage y los demás fueron con Renca, Chile y se pueden llegar a comparar.

	OBJECTID *	Shape *	FID
▶	1	Point	1738
	2	Point	1754
	3	Point	1742
	4	Point	1714
	5	Point	1700
	6	Point	1700
	7	Point	1700
	8	Point	1700
	9	Point	1700
	10	Point	1687
	11	Point	1691
	12	Point	1699
	13	Point	1743
	14	Point	1698
	15	Point	1689
	16	Point	1720
	17	Point	1747
	18	Point	1751
	19	Point	1749
	20	Point	1746
	21	Point	4100
	22	Point	4100
	23	Point	4101
	24	Point	4101
	25	Point	4101
	26	Point	4101
	27	Point	4101
	28	Point	4101
	29	Point	4101
	30	Point	4102
	31	Point	4102
	32	Point	4102
	33	Point	4102
	34	Point	4150
	35	Point	4150
	36	Point	4151

{10: {1: 36}}  
{10: {15: [['6 Meters to 8 Meters', 1, 0, 2.057]]}}

Figura 7.3: Resultado mapa forzado con ruteo con radio dinámico de 6 metro a 8 metros y 15 metros de forzado: ArcGIS 10.1 y Terminal

Sin embargo, independiente que se hayan asociado los 36 puntos en su totalidad, existen dos puntos (puntos 2 y 13) asociados en el mapa forzado con ruteo que no concuerdan en la ruta, como se puede ver en la Figura 7.4. Mientras que el algoritmo con la mejora del heading logra asociar bien esos puntos a la ruta con un valor de tolerancia de heading de 10°. Donde cada uno de estos puntos fueron inventados en cada punto GPS en grados.





Figura 7.4: Comparación de los dos algoritmos Forzado con ruteo y heading: ArcGIS 10.1

#### Tabla de resultados de mejoras:

En esta sección, se mostrarán variados resultados que son en base a 10 set de datos diferentes capturados por vehículos recolectores de residuos sólidos domiciliarios en Renca, Santiago. Cada uno de estos sets de datos fueron usados con distintos valores para cada algoritmo de mejora: a) radio Dinámico, b) Forzado con ruteo y c) heading (por separado o en combinación entre ellas) y d) el algoritmo base. Cabe destacar cuando se ejecutó cada algoritmo de mejora va acompañado con el algoritmo base ya que cada mejora tiene la intención de sumar al algoritmo base.

Cada uno de estos resultados fueron resumidos en tres tablas que se muestra el “% Solución”<sup>7</sup>, “Tiempo”<sup>8</sup> referente a cada caso. Además, se agregó una columna que contiene el número de veces que se llama a la función “Route solver”, ya que esta función es la que se lleva la mayor carga de computo dentro de los algoritmo ya sea siendo base o mejora. Esta función es la que está encargada de obtener la mejor ruta posible entre dos puntos y esto lo hace por medio de Dijkstra<sup>9</sup> entregando una lista de las calles más cercanas y las más lejanas dentro del radio que se esté analizando. La idea es que el algoritmo entra más a la función Route solver al hacer mayores verificaciones de esta función, por lo cual afecta gravemente en el tiempo.

La elección de los 10 sets de datos son por el hecho que son un gran reto para el algoritmo, como fue mencionado antes los puntos GPS traen un gran error en la posición y además la red vial consta de errores. Por lo tanto, es un gran reto para el algoritmo.

Los 10 sets de datos utilizados son:

<sup>7</sup> Es el porcentaje de la solución que entrega el algoritmo de puntos correctos asignados en la red vial del set de dato, considerando a los puntos GPS que no se asocian a una calle o sí se asocia a una calle, pero de forma incorrecta.

<sup>8</sup> Es el tiempo que se demora el algoritmo en entregar la solución.

<sup>9</sup> También llamado algoritmo de caminos mínimos, donde es un algoritmo para la determinación del camino más corto.

El primer set de dato consta de 201 puntos GPS que está ubicado en la comuna de Renca Santiago de Chile (Ver Figura 7.5).



Figura 7.5: Primer set de datos Renca, Santiago: ArcGIS 10.1

El segundo set de dato consta de 200 puntos GPS que está ubicado en la comuna de Renca Santiago de Chile. (Ver Figura 7.6).

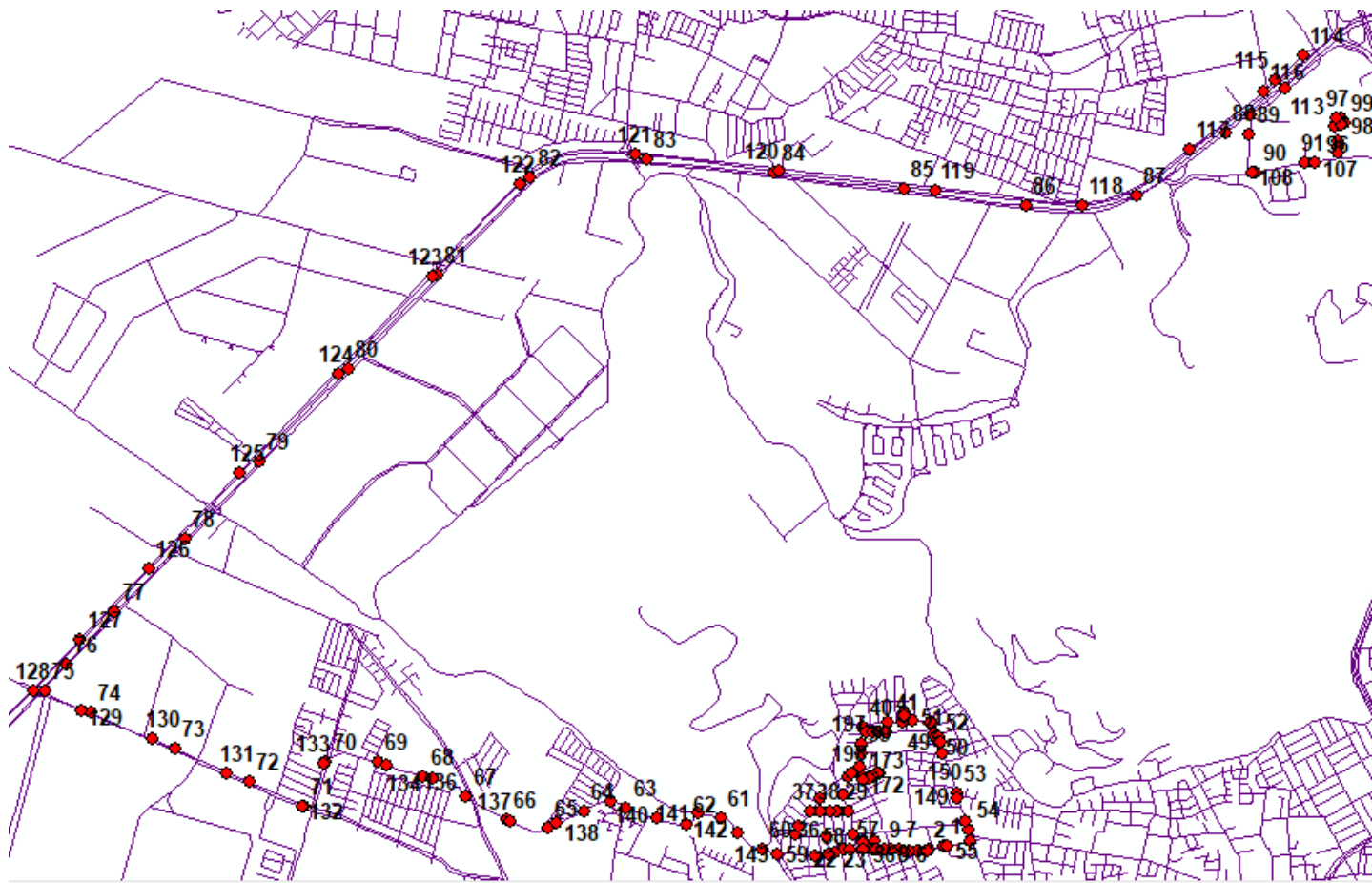


Figura 7.6: Segundo set de datos Renca, Santiago: ArcGIS 10.1

El tercer set de dato consta de 129 puntos GPS que está ubicado en la comuna de Renca Santiago de Chile. (Ver Figura 7.7).

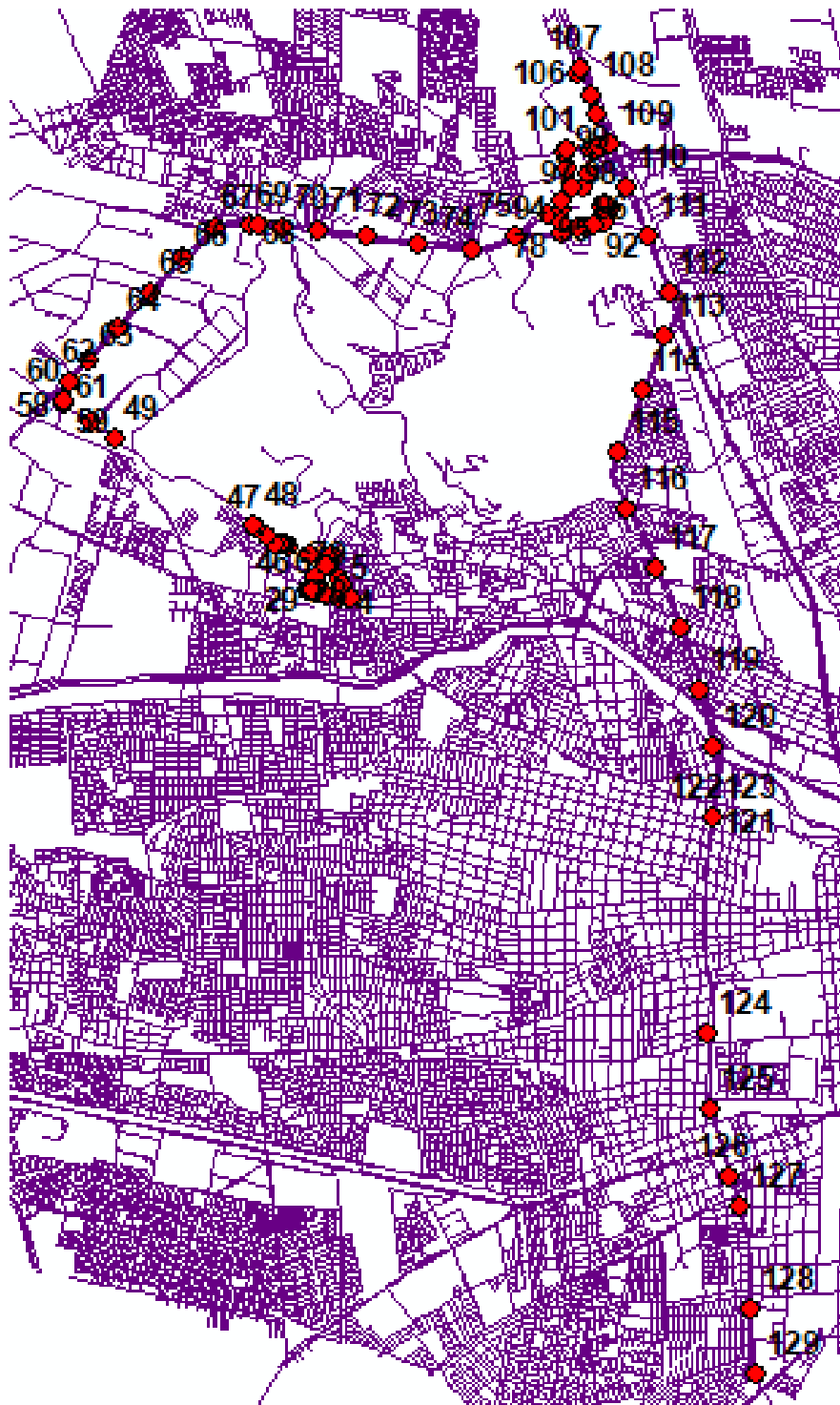


Figura 7.7: Tercer set de datos Renca, Santiago: ArcGIS 10.1

El cuarto set de dato consta de 228 puntos GPS que está ubicado en la comuna de Renca Santiago de Chile. (Ver Figura 7.8).

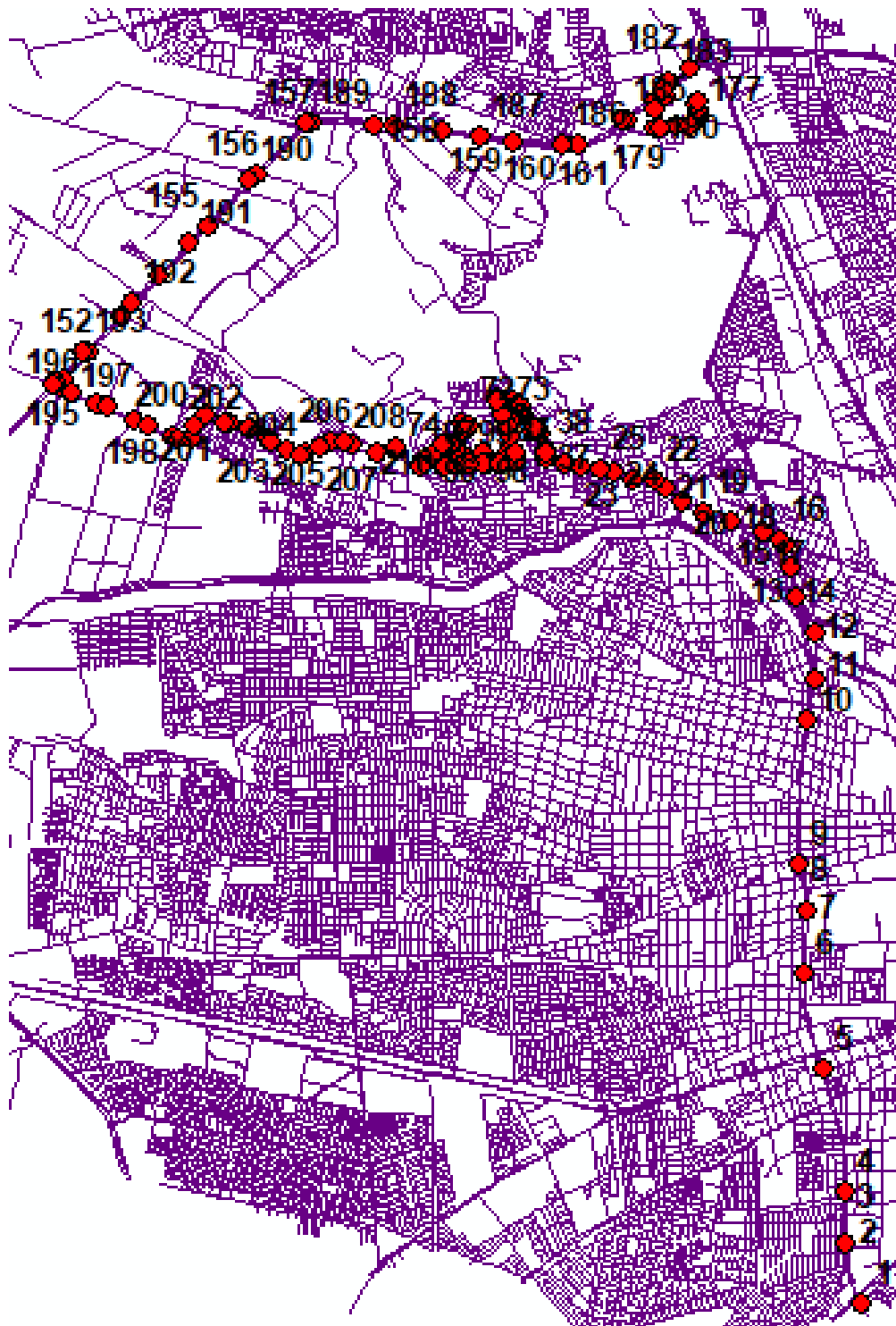
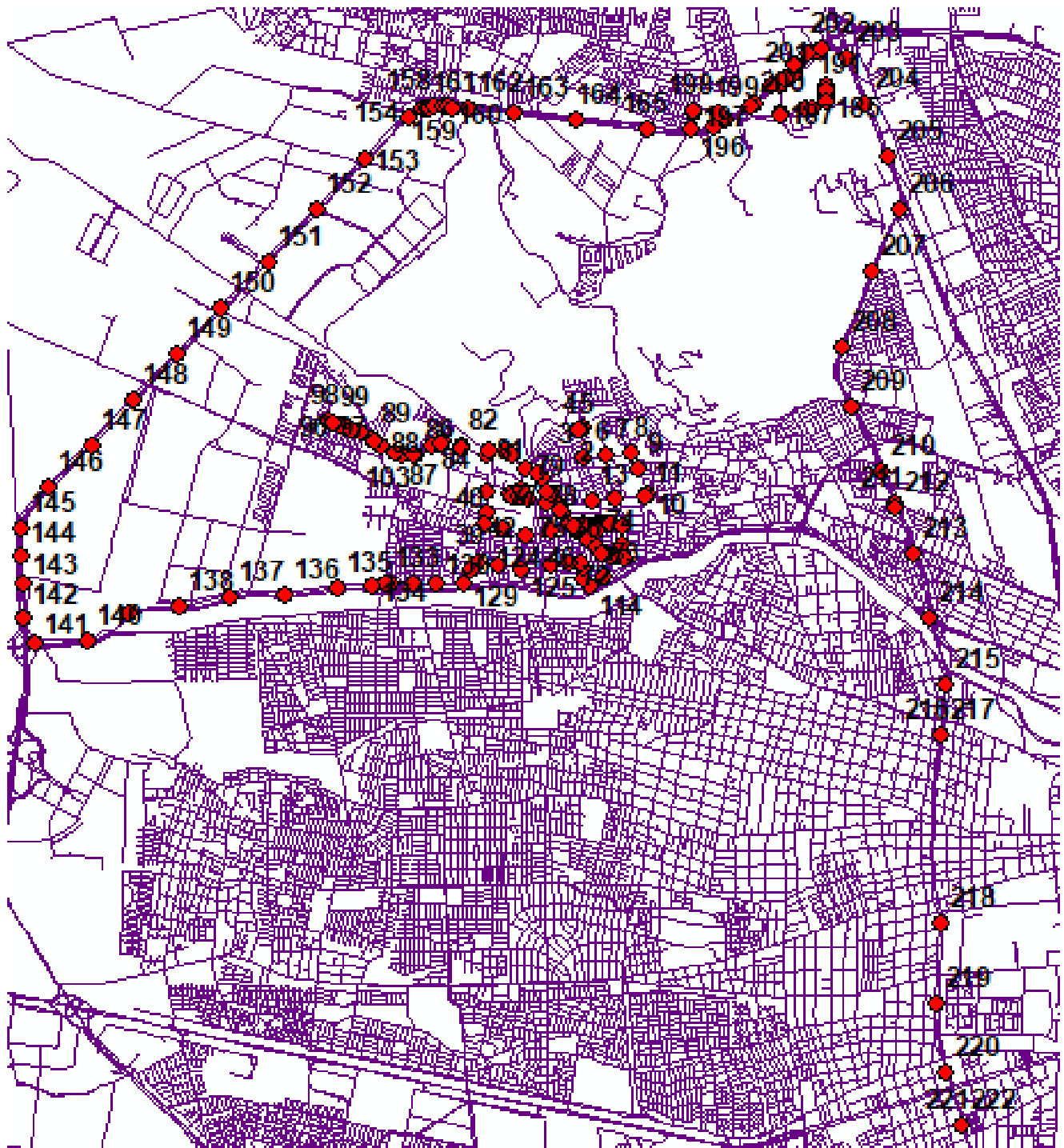


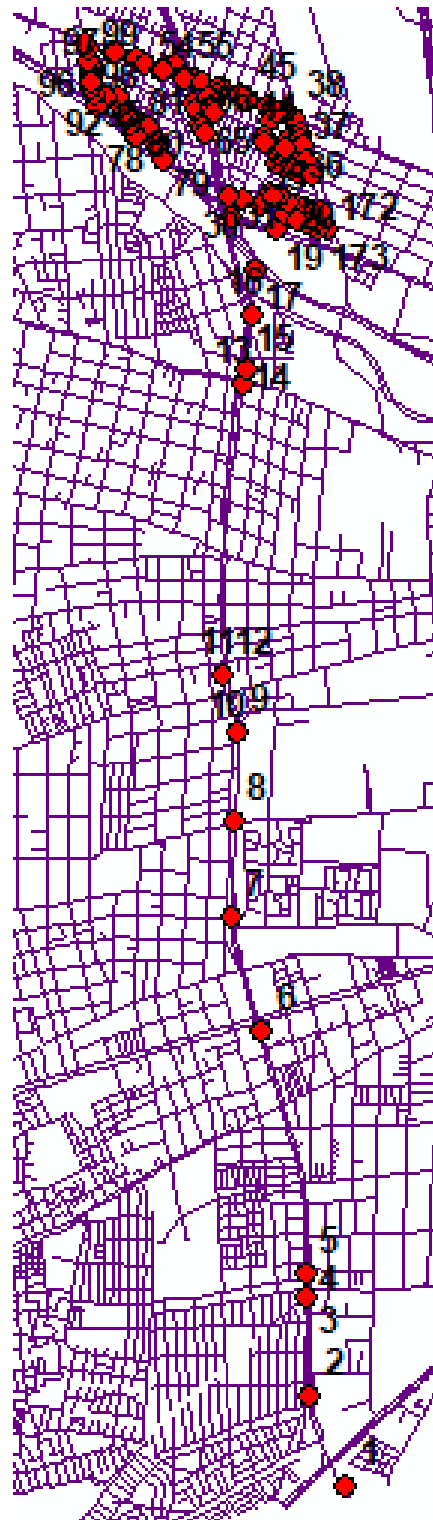
Figura 7.8: Cuarto set de datos Renca, Santiago: ArcGIS 10.1

El quinto set de dato consta de 223 puntos GPS que está ubicado en la comuna de Renca Santiago de Chile. (Ver Figura 7.9).



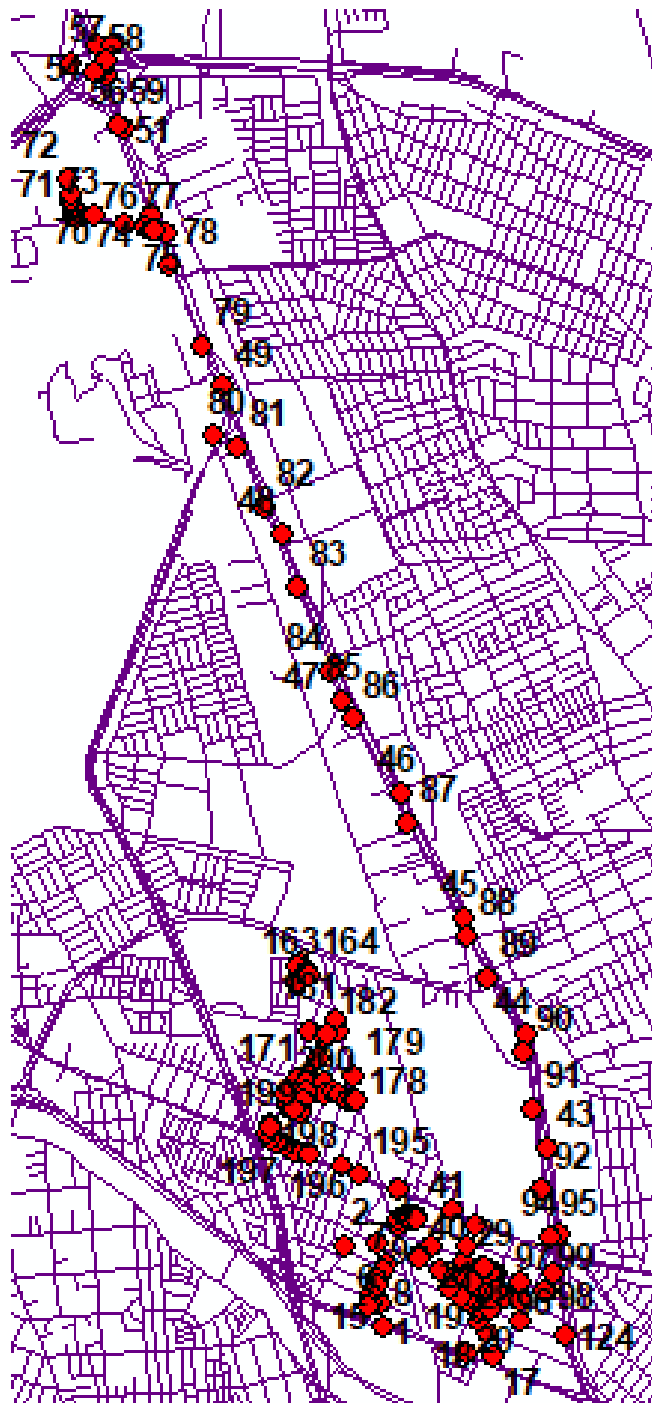
. Figura 7.9: Quinto set de datos Renca, Santiago: ArcGIS 10.1

El sexto set de dato consta de 181 puntos GPS que está ubicado en la comuna de Renca Santiago de Chile. (Ver Figura 8).



. Figura 8: Sexto set de datos Renca, Santiago: ArcGIS 10.1

El séptimo set de dato consta de 200 puntos GPS que está ubicado en la comuna de Renca Santiago de Chile. (Ver Figura 8.1).



. Figura 8.1: Sexto set de datos Renca, Santiago: ArcGIS 10.1



Los tres siguientes sets de datos contiene la dirección del vehículo en grados (heading). Cabe mencionar que algunos puntos se asignaron incorrectamente para ver el comportamiento del algoritmo.

El octavo set de datos consta de 113 puntos GPS que está ubicado en la comuna de Renca Santiago de Chile (Ver Figura 8.2). Además, sólo un grado esta incorrecto que es en el punto 2.

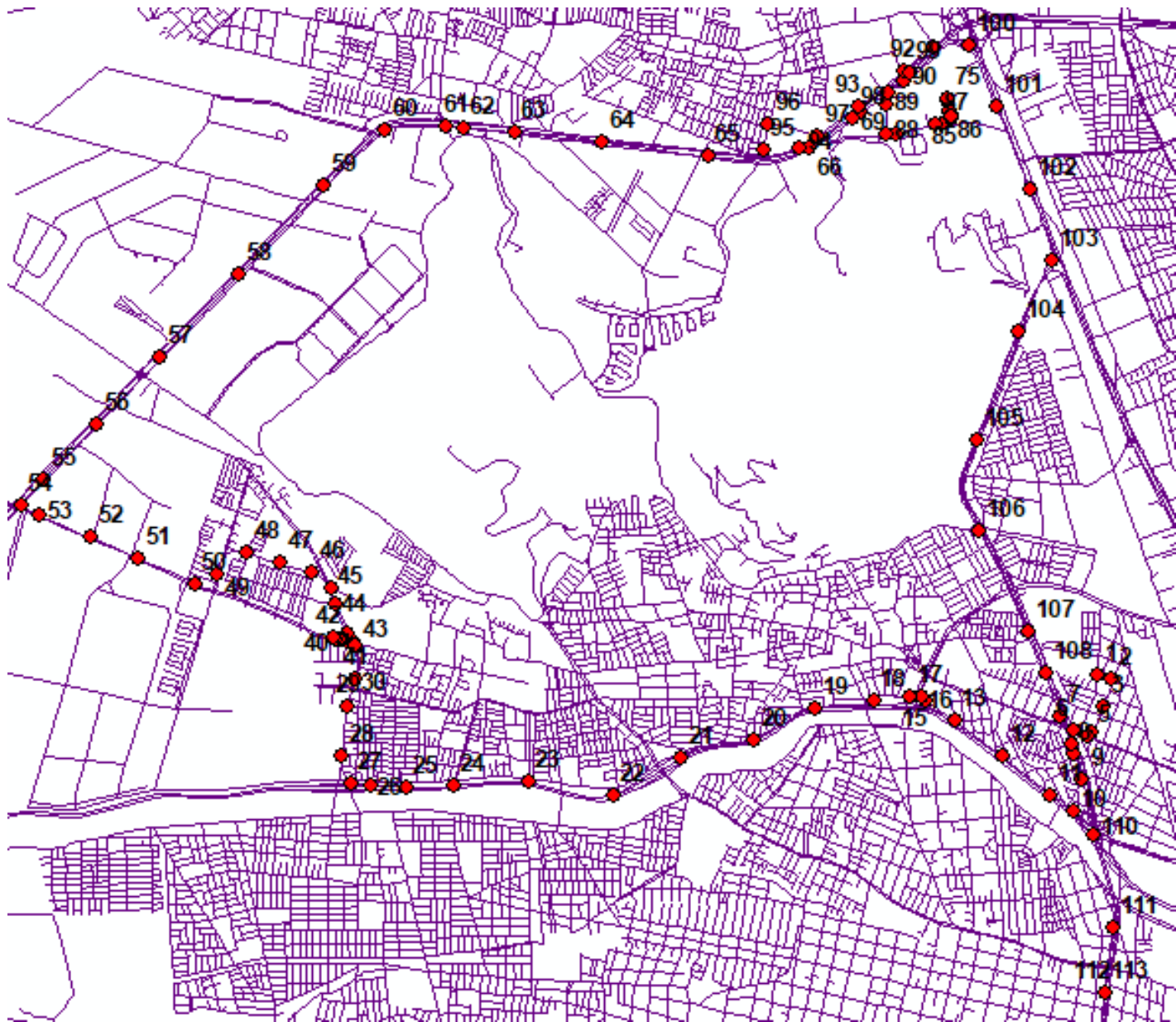


Figura 8.2: Octavo set de datos Renca, Santiago: ArcGIS 10.1

El noveno set de datos consta de 100 puntos GPS que está ubicado en la comuna de Renca Santiago de Chile. (Ver Figura 8.3). Además, se colocaron solo dos direcciones en grados que están incorrectos que son los puntos 4 y 10.



Figura 8.3: Noveno set de datos Renca, Santiago: ArcGIS 10.1

El décimo set de datos consta de 100 puntos GPS que está ubicado en la comuna de Renca Santiago de Chile. (Ver Figura 8.4). Además, se asignó sólo una dirección en grados incorrecto que es el punto 53.

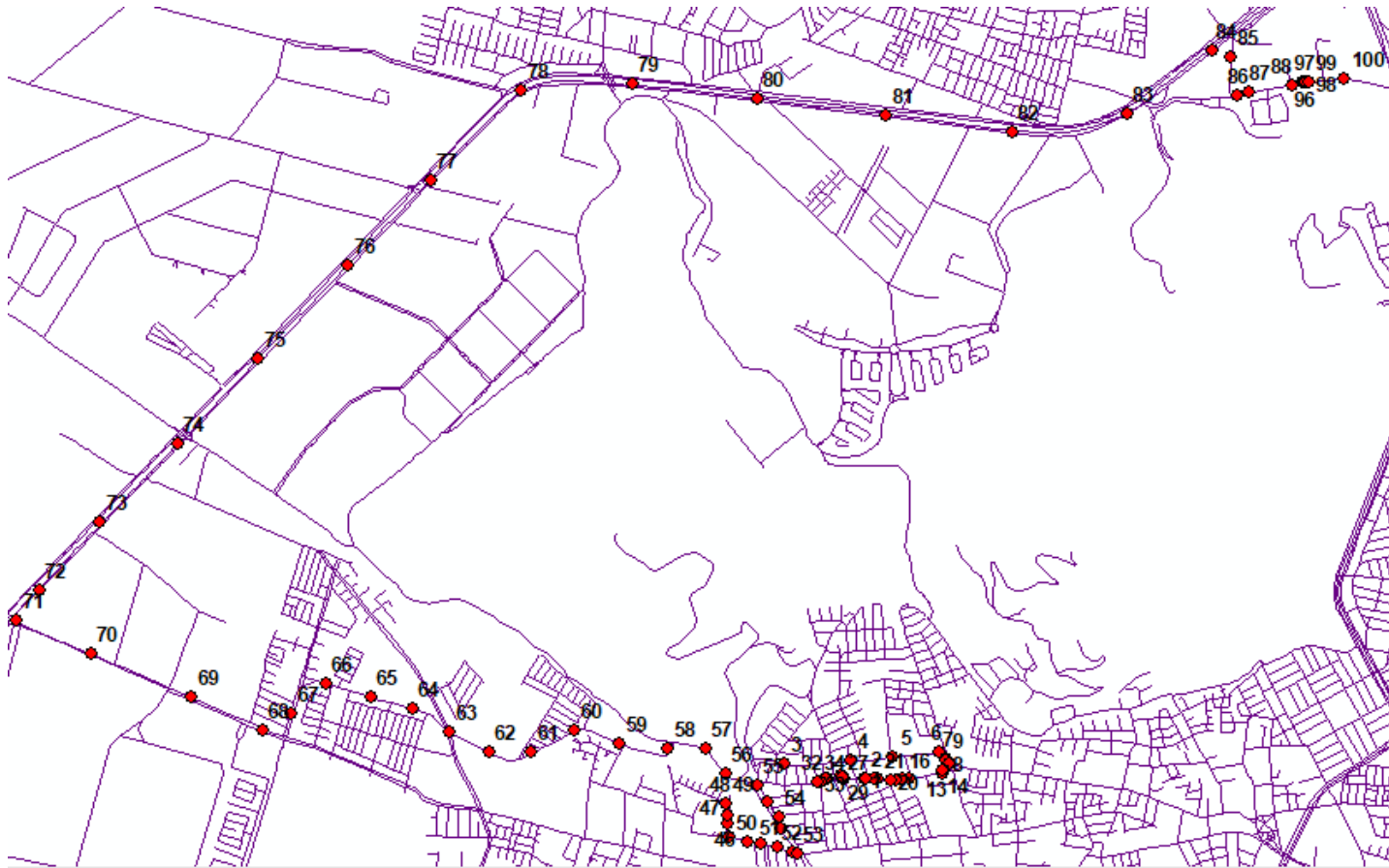


Figura 8.4: Decimo set de datos Renca, Santiago: ArcGIS 10.1

Los puntos con grados incorrectos de los 3 últimos sets de datos se pueden reflejado en las siguientes imágenes:

Se asignó un valor incorrecto de 20° al punto 2 del set de dato N° 8, como se puede ver en la Figura 8.5, el punto debiese tener un valor del heading de 335°.

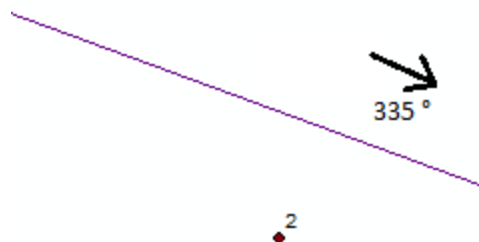
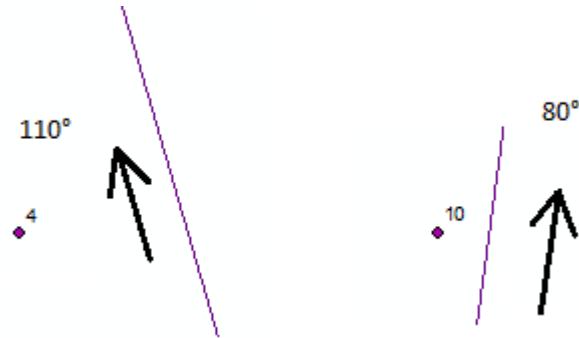


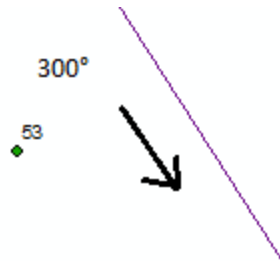
Figura 8.5: Incorrecto grado set de datos N°8 Renca, Santiago: ArcGIS 10.1

Para el set de dato N°9, hay dos grados incorrectos que corresponden a los puntos 4 y 10. Los grados puestos en el set de datos fueron 0° para ambos puntos. Como se puede ver en la Figura 8.6, los grados correspondientes son 110° para el punto 4 y 80° para el punto 10.



**Figura 8.6: Incorrecto grado set de datos N°9 Renca, Santiago: ArcGIS 10.1**

Para el último set de dato que corresponde al N°10, existe un punto (53) con grado incorrecto como se puede ver en la Figura 8.7, que tienen el valor asignado de 331° siendo que el valor que debe ser de 300°.



**Figura 8.7: Incorrecto grado set de datos N°10 Renca, Santiago: ArcGIS 10.1**

### Los resultados finales obtenidos:

La tabla 8.8 muestra los resultados generados por cada mejora por separado y además la base para ser comparado con cada uno de ellos en cuanto a porcentaje de solución y tiempo. Los valores asignados a cada variable que se encuentran en la columna “Datos Renca” se colocaron al probar variados casos y obtener la mejor situación de ellos.

Primero, se partió con el radio base del buffer y luego se fue aumentando gradualmente de 4 a 6 metros hasta obtener el mejor caso. Luego, el radio dinámico se incrementó de 7 a 11 metros junto con el mejor caso del radio base, y después el forzado con ruteo se estableció con la aplicación ArcGIS 10.1 que da la posibilidad de saber la cantidad de metros de radio posibles para asociar todos los puntos a la red vial. En el caso de las variables de tolerancia, la tolerancia de velocidad primero probó con una tolerancia de 15 a 30 km/h, pero hubo solo dos sets de datos que se tuvo que aumentar a 35 km/hr (noveno y décimo) ya que había puntos que estaban sumamente mal posicionados y la tolerancia de velocidad 30 km/h era muy bajo. Por lo tanto, se tuvo que subir a 35 km/hr. Para el caso de la tolerancia del heading, primero, se probó con 10° donde se vio que excluye demasiados puntos al tratar de ser aceptado. Luego, se pasó a 15° donde aún había puntos que se excluyen muchos más puntos aún. Por último, se probó con 20° que ya se tomaban ya bien todos los puntos.

Antes que todo se debe aclarar nuevamente que solo los 3 últimos set de datos (octavo, noveno y décimo) se pueden probar con la mejora del heading ya que a estos set de datos se les agregó manualmente la dirección en grados.

Entonces ¿Cuál de las tres mejoras por sí solas o simplemente la base fue la mejor solución?

Se puede observar que en la columna “Forzado con ruteo”, se obtuvieron todos los mejores resultados. Pasando por ejemplo en la fila N°1 de un 82% de solución con el base, a un 97% de solución con la mejora del Forzado con ruteo.

Además, se notó que en el set de datos N°8, se obtuvo el mismo resultado para la solución de mejora solo Forzado con ruteo vs solo Dinámico ¿Por qué sucedió esto?, ¿De que el radio dinámico llegase alcanzar el mismo resultado de solución del Forzado con ruteo? Es porque el set de datos N°8 no requería un radio más grande que el radio límite del dinámico. Por lo tanto, los puntos GPS del set de datos estaban cerca de los segmentos de calles a asociar, pudiéndose asociar fácilmente.

El Forzado con ruteo fue mejor que las otras mejoras porque éste fuerza la asociación de los puntos a la vía mientras que el Base y el Dinámico no lo hacen. Luego, se le aplica a cada punto forzado un ruteo para verificar que los puntos estuvieron asociados correctamente con el forzado con ruteo.

Datos Renca							Base		Radio Dinamico		Forzado con ruteo		Heading	
N°	Cantidad de puntos	Radio Base (m)	Dinamico (m)	con ruteo (m)	Radio de Velocidad (m)	Radio de Gravedad (m)	% Solución	Tiempo (min)	% Solución	Tiempo (min)	% Solución	Tiempo (min)	% Solución	Tiempo (min)
1	201	6	10	40	30		82%	8.723	90.5%	9.062	97%	10.098		
2	200	6	10	32	30		78%	7.296	94.5%	7.927	97.5%	9.243		
3	129	5	10	31	30		68.2%	5.058	89.1%	5.842	97.6%	6.626		
4	228	5	11	22	30		67.1%	8.244	0,93	12.773	97.8%	11.187		
5	223	4	10	20	30		61.4%	8.142	94.6%	10.154	0,96	10.978		
6	181	4	9	12	30		71.8%	6.867	95.1%	8.37	98.3%	8.61		
7	200	4	9	12	30		71%	6.997	96%	8.082	98%	9.733		
8	113	4	9	20	30	20°	61.9%	4.562	93.8%	5.475	94.6%	5.733	63.3%	8.207
9	100	4	7	13	35	20°	59%	4.583	83%	4.611	99%	5	59%	7.925
10	100	4	7	15	35	20°	52%	3,30	84%	4,33	96%	5,03	53%	6,61

Tabla 8.8: Primera parte de la primera tabla de resultados: Elaboración propia

Ahora bien ¿Qué sucede si se hacen combinaciones entre estas 3 mejoras?, ¿Habrá algún cambio?

En una segunda parte, se generó esta duda y para responderla se combinaron todas las mejoras (Tabla 8.9), junto con el base que siempre está presente en cada caso.

Todas las mejoras		Dinámico y forzado con ruteo		Forzado con ruteo y heading	
% Solución	Tiempo (min)	% Solución	Tiempo (min)	% Solución	Tiempo (min)
		97%	11.104		
		97.5%	9.438		
		97.6%	7.24		
		97.8%	11.776		
		0,96	10.581		
		98.3%	9.442		
		98%	10.094		
94.6%	10.528	93.8%	5.436	94.6%	12.358
98%	11.347	99%	5.623	98%	11.204
99%	11,65	96%	5,03	99%	11,49

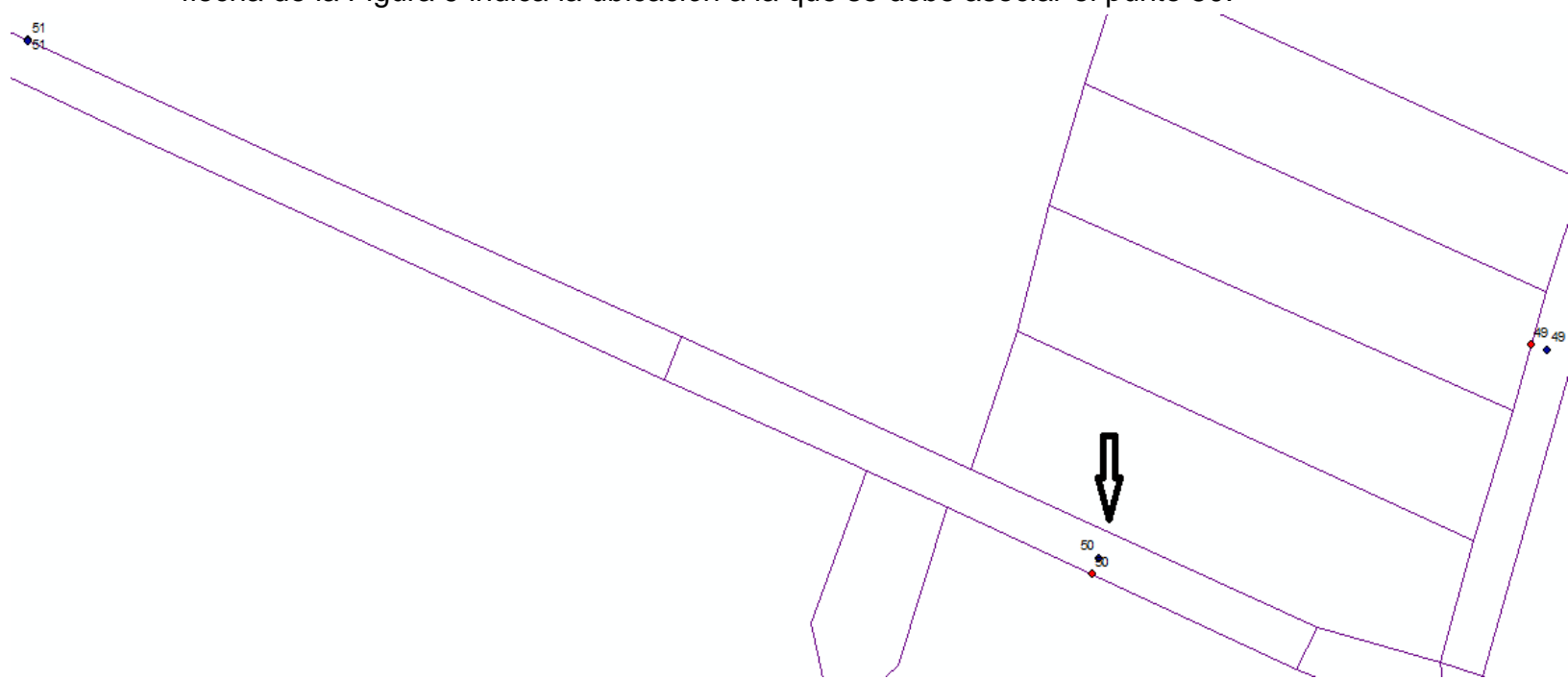
Tabla 8.9 : Segunda parte de la primera tabla de resultados: Elaboración propia



Los resultados fueron muy buenos, si los comparamos con el Forzado con ruteo de la Tabla 8.8 donde fue la mejora con mejor desempeño. Se nota que el Heading hace su trabajo de asociar bien los puntos que estaban mal asociados ya sea por el Base, el Dinámico y el Forzado con ruteo.

Analizando los resultados de la combinación de Todas la mejoras en el N°8 el 94.6% ¿Por qué no llego al 100% si estaban correcto los ángulos puestos manualmente excepto uno?

Como se puede observar en la Figura 9, un punto está asociado incorrectamente en el set de datos N°8, donde hay calles que se asemejan mucho en ser paralelas y otras carreteras son paralelas. Entonces si los segmentos de vía están dentro del radio, sumado a que el punto GPS esté tan mal posicionado (punto color azul) más cerca de la calle incorrecta y que a cualquiera de las dos calles se acepte la tolerancia de velocidad y tolerancia de grados, finalmente este punto se asocia incorrectamente. La flecha de la Figura 9 indica la ubicación a la que se debe asociar el punto 50.



**Figura 9: Asociación incorrecta Heading: ArcGIS 10.1**

Además, comparando la implementación de todas las mejoras con Forzado con ruteo, logran el mismo porcentaje de solución excepto en el último set de datos que el Forzado con ruteo baja el porcentaje en comparación con todas las mejoras de un 99% vs un 96%. El aumento de este porcentaje fue por el Heading que filtra mayormente al cual calle es más correcta al momento de asociar cada punto GPS. ¿Pero ¿cuál es la desventaja de esto?, es el tiempo ya que, si lo comparamos con el Forzado con ruteo, el tiempo de ejecución del algoritmo es casi el doble.

Se buscó otra combinación hablando con mi profesora guía Carola Blazquez, donde se junta el Dinámico y forzado con ruteo, y Forzado con ruteo y heading.

Además, se revisó si las mejoras relacionadas con el Dinámico y Forzado con ruteo logran obtener un mejor tiempo con un alto %solución. Sin el heading ya que, con este aumenta el tiempo y ver cómo se comporta solamente en la combinación Dinámico y Forzado con ruteo.

En cambio, la combinación del Forzado con ruteo y heading se eligió ya que el heading por sí solo falló muchas veces, ya que el radio base era muy pequeño no pudiendo asociar todos los puntos más lejanos. Así que en esta combinación el Forzado con ruteo alcanzaría los puntos GPS más lejanos. Y con el heading puedan asociarse a la calle correcta.

Al comparar el buffer con radio dinámico junto con el Forzado con ruteo vs solo el Forzado con ruteo se obtuvieron los mismos porcentajes, pero en un menor tiempo en la mejora sola del Forzado con ruteo, excepto en un set de datos Número 8 que mejoró el tiempo en la combinación del Dinámico junto con el Forzado con ruteo. ¿Por qué sucedió que esta combinación entregue resultados en menos tiempo que el Forzado sólo? y ¿En el set N°8 porque dio unos segundos menos?

Siguiendo la lógica del algoritmo, primero es el radio base, donde si no alcanza el punto GPS, pasa al buffer dinámico y aumenta gradualmente hasta cierto límite y luego si no alcanza nuevamente es forzado el punto. Entonces en la combinación del Dinámico con el Forzado con ruteo, cuando no se encuentre con el buffer Dinámico y se debe encontrar en el Forzado con ruteo tuvo que pasar por el loop completo del Dinámico y eso agrega un poco más tiempo.

¿Por qué se obtienen los mismo resultado entre el dinámico y forzado con ruteo vs el forzado con ruteo? Porque cuando un radio selecciones ciertas calles se ordenan en una prioridad de la calle más cercana a la más lejana. Por lo tanto, los dos van a llegar a la misma lista, pero una más rápida que otra por lo que fue explicado anteriormente.

Se analiza el forzado con ruteo y heading para determinar si mejora el Heading solo. En esta caso lo mejora dentro de un rango de un 31% a 46%, llegando al mismo porcentaje de solución como si fuesen Todas las mejoras juntas. El único set de datos que afecta al heading es cuando están muy mal posicionado los puntos GPS como ocurre con el set de dato N°8 y se asocian a calles que estén relativamente paralelas.

La última parte de la Tabla 9.1 es referente a la función "Route \_solver" que está relacionado con el computo del algoritmo y el tiempo. La idea es contar cuantas veces se pasa por esta función al ejecutar cada mejora y determinar por cual mejora de forma independiente o en combinación con otras mejoras pasa menos veces y tenga buen resultado.

Entonces analizando la Tabla 9.1 que muestra esta última parte de la Tabla 9.1 hubo menores ejecuciones de esta función Route solver con el algoritmo Base. Sin embargo, se logró un porcentaje muy bajo de solución y el más alto en número de ejecuciones con Todas las mejoras ya que si tiene que llegar al Forzado con ruteo debe pasar por el



Base, Dinámico y recién llega al Forzado con ruteo., Por lo tanto, cada vez que pase la función Route solver con el algoritmo base suma al contador de veces, al igual que con el buffer Dinámico. Al subir el contador de veces que se ejecuta la función Route solver, aumenta el tiempo.

Dentro del menor y mayor rango de las ejecuciones, se pueden encontrar las otras pruebas de mejoras, pudiéndose analizar que los más centrales del conteo Route solver vs solución y tiempo son las mejoras de solo Forzado y la combinación de Dinámico con Forzado con ruteo.

Route solver						
Base	Dinámico	Forzado con ruteo	Heading	Todas las mejoras	Base, Dinámico y forzado con ruteo	Forzado con ruteo y heading
182	207	249			282	
149	202	203			207	
79	135	156			172	
143	255	261			271	
118	247	248			239	
117	204	184			193	
117	205	205			211	
69	122	129	97	166	124	138
58	98	108	72	128	112	124
36	89	103	59	122	101	117

**Tabla 9.1: Tercera parte de la primera tabla de resultados: Elaboración propia**

Entre los 3 últimos sets de datos, la mejor solución se obtuvo al agregar el heading, con Todas las mejoras y la combinación de Forzado con ruteo con Heading. Sin embargo, toma mucho tiempo la respuesta, el doble o más que las demás mejoras, mientras que el otro grupo que destacó fue los que contenían Forzado con ruteo que son el solo Forzado con ruteo y Forzado con ruteo junto con el dinámico que tuvieron el mismo resultado, pero todos arriba del 90%.

Además, se decidió crear la Tabla 9.2, pero que baje el radio del buffer con el algoritmo Base junto con el buffer Dinámico para todos los sets de datos por igual y bajando las dos tolerancias de la velocidad y del heading, pero aumentando aún más el radio del Forzado con ruteo para determinar el comportamiento en un peor caso.

Datos Renca							Base		Radio Dinamico		Forzado con ruteo		Heading	
N°	Identidad de punto	Radio Base (metros)	Radio Dinamico (metros)	Radio con ruteo (metros)	Radio de Velocidad (metros)	Tolerancia de grado	% Solución	Tiempo (min)	% Solución	Tiempo (min)	% Solución	Tiempo (min)	% Solución	Tiempo (min)
1	201	4	7	45	25		62.1%	7.919	83.5%	9.797	96.5%	10.316		
2	200	4	7	45	25		58.5%	6.109	84%	8.905	97.5%	9.478		
3	129	4	7	45	25		53.4%	5.001	79.8%	6.209	98.4%	6.759		
4	228	4	7	45	25		57.8%	8.141	82.8%	11.85	97.8%	11.652		
5	223	4	7	45	25		61.4%	8.538	91.4%	10.932	93.7%	12.566		
6	181	4	7	45	25		71.8%	7.001	93.3%	8.716	97.2%	9.185		
7	200	4	7	45	25		71%	7.834	93%	9.309	97%	9.403		
8	113	4	7	45	25	15°	61.1%	4.778	90.2%	5.919	91.1%	5.972	62.8%	12.149
9	100	4	7	45	25	15°	58%	4.637	83%	5.295	99%	5.412	58%	8.011
10	100	4	7	45	25	15°	52%	3.595	83%	4.46	95%	4.861	54%	6.866

**Tabla 9.2: Primera parte de la segunda tabla de resultados: Elaboración propia**

Comparando con los resultados de la Tabla 8.8 bajan un poco el tiempo, pero no baja el % solución como se esperaba en el Forzado con ruteo solo, lo que indica que es bien robusto. Incluso en este caso, aumentó en el set de datos N°3 de un 97.6%, como se observa en la Tabla 8.8 columna Forzado con ruteo, a un 98.4% con el Forzado con ruteo solo en la Tabla 9.2. Esto ocurre porque con el Forzado con ruteo en la Figura 9.2 se utilizó un radio más grande con los puntos que se estaban asociando mal, su calle correcta estaba más lejos que 31 metros como se puede ver en la distancia de la columna Datos Renca, con el Forzado con ruteo En esta Tabla 9.2, se aumentó el radio del Forzado a 45 metros dándole la oportunidad a estos puntos mal asociados para poder asociarse bien.

Un ejemplo puede explicar esto. En la Tabla 9.3, se identifican tres puntos. Uno es el set de dato N°3 (color verde) que esta sin asociar, otro es el Forzado con ruteo de la primera tabla (color rojo) que está asociado a la derecha y por último el Forzado con ruteo de la segunda tabla que está asociado a la izquierda. En la Figura 9.3, se puso un forzado con ruteo de radio 31 metros por lo cual si se ven en la Tabla 9.3 con 31 metros no alcanza a asociar a la calle correcta, que es el de la izquierda y verifica solamente la calle que está a la derecha y se fuerza a esa. Mientras que, si el radio el buffer es más grande, como en el caso de la tabla 9.2 de 45 metros, entonces se fuerza la asociación del punto GPS a la segunda calle izquierda.

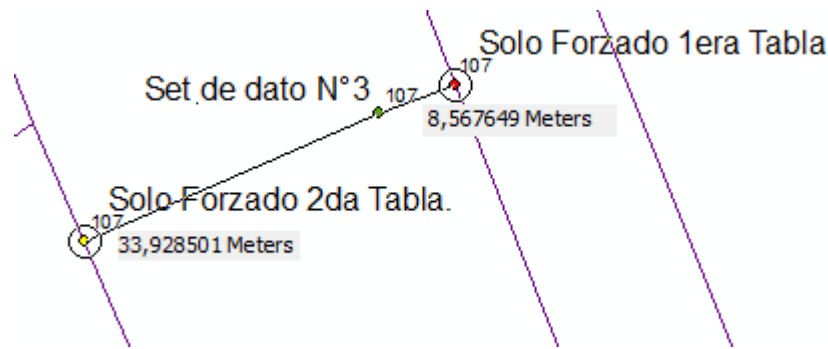


Figura 9.3: Ejemplo del Forzado con ruteo Fuente: Elaboración propia

Se podría preguntar ¿ si la prioridad de cómo se asocia un punto es del más cercano a más lejos, se puede asociar al más lejano?, es decir, si se tiene un radio de 45 metros (como es en este caso) y el algoritmo encuentra una lista de calles candidatas al cual asociarse, se ordena de menor distancia a mayor distancia del punto, se escoge a la primera calle que cumpla con las tolerancias y por último, se asocia. Siguiendo la idea de los 45 metros o si el buffer es de 100 metros, se debería forzar a la calle de la derecha porque está más cerca del punto. Pero la respuesta es no, porque este punto 107 no pasó la prueba de tolerancia de velocidad, pero se asociaban igual.

Entonces ¿Por qué se asoció de a la calle de la derecha si no se aceptaba la tolerancia de velocidad?

Porque existe una función al final de todo el algoritmo, antes que se dé por desasociado el punto GPS, que tiene por nombre “KiBetweenKj”.

Un ejemplo para dar a conocer esta función se muestra en la Figura 9.4:

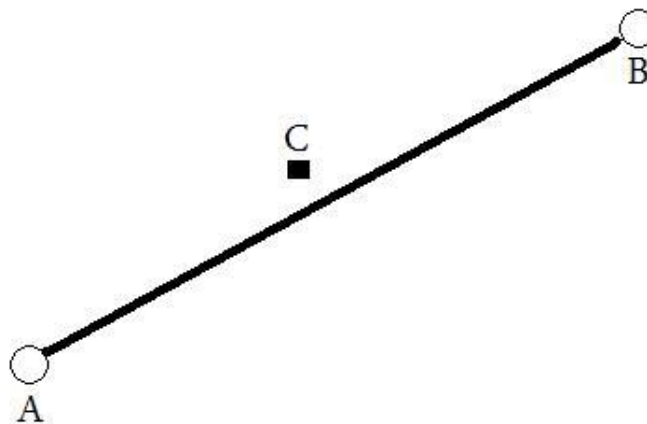


Figura 9.4: Función KiBetweenKj Fuente: [gis.stackexchange.com](https://gis.stackexchange.com)

La Figura 9.4 consta de 3 puntos A, C y B. Primero, consideraremos el punto A y C donde A sería Ki y C es Kj, entonces C está tratando de asociarse a la calle, pero fue rechazada la tolerancia de velocidad, quedando C hasta ahora desasociado. Luego, pasa a ser B el Kj. Se hace el ruteo entre los puntos Ki y Kj, y se asocian quedando en

la calle tal cual como se muestra B en la Figura 9.4. Entonces ahora se pregunta el algoritmo ¿Existe puntos entre A y B que no fueron asociados? Y ¿Ese punto con el radio que se está analizando toma alguna posible calle? La respuesta es sí, que es el punto C. Por lo tanto, C es forzado a la calle entre A, B y asociándose a dicha calle. Como punto final, todo esto sucede porque el radio no encuentra más calles candidatas y esa calle es la única opción que se tiene.

Ahora volviendo a la génesis del tema en la Figura 9.3, si existe un radio de 31 metros como es en la tabla 8.8, solo tiene una única opción de calle, que sería la de la derecha y pasa justamente lo que se explicó anteriormente. Ahora si se tiene un radio de 45 metros como en la tabla 9.2, con un radio de 45 metros, entonces seleccionó no una posible calle, si no que dos calles posibles. Por lo tanto, como la primera calle no se acepta por la tolerancia de velocidad, pasa a verificar a la segunda calle que es el de la izquierda y en ese sí se acepta la tolerancia de velocidad, por lo cual quedando asociado a ella.

Ahora bien ¿Qué sucede de nuevo con la comparación Forzado con ruteo vs todas las combinaciones?

Se puede apreciar en la Tabla 9.5 que aun comparando el Forzado con ruteo a solas( Tabla 9.2) vs Todas las mejoras (Tabla 9.5), sigue siendo el mejor en tiempo y %solución , donde en Todas las mejoras aún el tiempo es casi el doble pero esta vez no se destaca tanto por el hecho que la tolerancia del heading es mucho más bajo y eso le afecto bajando el porcentaje de solución o puntos asociados correctamente, al igual que afecto levemente en el Forzado con ruteo. Por lo tanto, aún sigue siendo uno de los mejores.

Todas las mejoras		Dinámico y forzado con ruteo		Forzado con ruteo y heading	
% Solución	Tiempo (min)	% Solución	Tiempo (min)	% Solución	Tiempo (min)
		96.5%	11.655		
		97.5%	10.25		
		98.4%	7.623		
		97.8%	13.779		
		93.7%	12.615		
		97.2%	9.852		
		97%	10.224		
83.1%	17.238	91.1%	6.111	83.1%	17.338
98%	13.014	99%	6.875	98%	14.383
97%	12.366	95%	5.179	97%	11.088

**Tabla 9.5: Segunda parte de la segunda tabla de resultados: Elaboración propia**

Comparando el Forzado con ruteo solo (Tabla 9.2) vs a la combinación del buffer Dinámico y el Forzado con ruteo (tabla 9.5, siguen siendo el mismo porcentaje solución entre ambos, Por lo tanto, no ha cambiado referente a lo que se dijo en la Tabla 8.9, pero sí bajó el porcentaje de solución entre un 2% si comparamos entre la Tabla 8.9 y es por la tolerancia de velocidad. Esto se debe a que el punto GPS necesita aún más tolerancia de velocidad para ser aceptado.

Otra comparación es entre Todas las mejoras vs la combinación de Forzado con ruteo y heading que bajan a los mismos porcentajes entre un 4%, si comparamos con la tabla previa que es la 8.9 y le afectó al bajar la tolerancia del heading, donde se discrimina más las calles y baja el resultado de solución. El único porcentaje que no bajó fue en el set de datos 9 manteniéndose con un 98% de resultado correcto de asociación. Afecto también el bajar la tolerancia de heading y de velocidad, en el tiempo, más se ve reflejado en las mejoras que contienen el heading ya que a los puntos al momento de asociar los hace pasar más veces por la función "Route solver" para la verificación y aceptación del punto GPS. Eso sí en Todas las mejoras, el tiempo subió 7 minutos más referente a la segunda parte de la Tabla 8.9 y es porque al tener menos la tolerancia de heading verifica hasta el último dato en la lista y llamando a cada función del algoritmo más veces aumentando el tiempo de computo.

Como última comparación es el contador de Route solver en la tabla 9.6, donde se puede concluir que si se bajan las tolerancias sube las veces que pasa por la función Route solver. Además, hace más comparaciones con respecto al Route solver de la primera Tabla 9.1. Hay algunos que son menores como es en el caso del primer set de datos de la Tabla 9.1 que pasa 287 veces por Route solver, mientras que en la Tabla 9.6 pasa 267 veces. Lo que sucede es porque como el radio del Base es más pequeño y el radio del Dinámico al igual es más pequeño siendo comparado con los radios de la primera tabla que entrega los valores la Tabla 8.8, entonces no toma ninguna calle cercana, por lo cual pasa directamente al Forzado con ruteo y no pasando previamente por varias comprobaciones anteriores de la función Route solver.

Route solver						
Base	Dinámico	Forzado con ruteo	Heading	Todas las mejoras	Base, Dinámico y forzado con ruteo	Forzado con ruteo y heading
140	194	253			267	
119	182	203			210	
129	116	160			167	
157	218	276			273	
154	230	265			272	
146	201	207			228	
149	198	208			208	
85	129	135	102	266	130	223
70	99	126	73	196	147	195
56	91	105	61	129	103	125

**Tabla 9.6: Tercera parte de la segunda tabla de resultados: Elaboración propia**

En la Tabla 9.7 se evaluó de la siguiente forma. Ya que se vio que con el Forzado con ruteo independiente utilizando bajos valores para las dos tolerancias y el radio base junto con el radio dinámico siéndose obtiene uno de los mejores en tiempo y resultado el forzado con ruteo. Ahora bien ¿Qué sucede si bajo drásticamente el Forzado con ruteo? mientras que las demás variables las aumento? ¿Bajará el % de solución? y ¿Habrá más desasociados?

Entonces el radio Base se subió de 4 a 5 metros, en el radio Dinámico se subió de 7 a 8 metros, la tolerancia de velocidad se subió de 25 a 35 Km/h y la tolerancia de grados se subió de 15° a 25°. sin embargo, el radio del Forzado con ruteo se bajó de 45 a 15 metros. Todos estos datos se pueden ver reflejado en las primeras columnas de la Tabla 9.7.

Datos Renca							Base		Radio Dinamico		Forzado con ruteo		Heading	
N°	cantidad de puntos	Radio Base (metros)	Radio Dinamico (metros)	Radio con ruteo (metros)	Radio de Velocidad	Tolerancia de grado	% Solución	Tiempo (min)	% Solución	Tiempo (min)	% Solución	Tiempo (min)	% Solución	Tiempo (min)
1	201	5	8	15	35		72%	10.845	89.5%	9.561	92.5%	10.642		
2	200	5	8	15	35		66.5%	8.021	92.5%	9.192	95%	9.515		
3	129	5	8	15	35		67.4%	7.431	83.7%	6.171	87.5%	6.452		
4	228	5	8	15	35		66.2%	9.708	88.1%	12.004	96.4%	12.065		
5	223	5	8	15	35		81.6%	10.533	92.3%	10.291	92.8%	10.876		
6	181	5	8	15	35		82.3%	8.078	93.92%	12.486	98.3%	9.624		
7	200	5	8	15	35		81.5%	8.998	94.5%	9.15	97%	10.211		
8	113	5	8	15	35	25°	81.4%	5.478	92.9%	5.943	92%	6.045	82.3%	11.312
9	100	5	8	15	35	25°	69%	5.59	88%	4.982	99%	5.821	69%	10.015
10	100	5	8	15	35	25°	64%	4.118	92%	4.689	96%	5.229	67%	7.619

**Tabla 9.7: Primera parte de la tercera tabla de resultados: Elaboración propia**

Como resultados se obtuvieron que el Base y en Dinámico por separado aumentaron en el porcentaje de solución versus la Tabla 9.2 en la columna Base y Dinámico. Pero no superando al porcentaje solución en la columna Base y Dinámico de la Tabla 8.8.

Con respecto al Forzado con ruteo, se nota la baja en el porcentaje de solución vs a la Tabla 8.8 vs la Tabla 9.2 que fue respondida anteriormente. El hecho que haya bajado mucho es porque el radio de Forzado es muy pequeño. En realidad, el set de datos que necesitan un radio de 30 metros, en este caso, quedarían desasociados, por lo cual baja el porcentaje de solución. Aclarando como fue dicho anteriormente, el porcentaje de solución considera los puntos GPS no asociados a una calle y los que fueron asociados a una calle, pero incorrectamente. A pesar de todos esto, el Forzado con ruteo es el mejor.

Por último, cabe destacar además que, si nos fijamos, los porcentajes de solución de la Tabla 9.7, aumentó en la columna del Heading. Esto se debe porque el radio de la Base en la Tabla 9.2 fue de 4 metros para los sets de datos 8, 9 y 10. Mientras que en la columna heading de la Tabla 9.7, se subió a 5 metros, entonces es más probable que en ese metro de diferencia encuentre calles cercanas para puntos GPS que quedaron desasociados y ahora fuesen asociados pasando de 37 puntos GPS desasociados de la Tabla 9.2 a 15 puntos GPS desasociados de la Tabla 9.7, es decir en ese metro se asociaron 15 puntos más. No habría problemas con las tolerancias ya que son altas y el tiempo se mantuvo en el mismo rango de los otros resultados de las otras tablas.

Analizando los resultados de la Tabla 9.8 con respecto a las combinaciones de las mejoras. En Todas las mejoras hubo un aumento de porcentaje de solución en comparación a la Tabla 9.5 columna Todas las mejoras, el hecho fue porque se aumentó la tolerancia del heading. Pero ¿Si disminuyó el radio del Forzado porque no lo afectó tanto?

Todas las mejoras		Dinámico y forzado con ruteo		Forzado con ruteo y heading	
% Solución	Tiempo (min)	% Solución	Tiempo (min)	% Solución	Tiempo (min)
		92.5%	10.675		
		95%	10.49		
		87.5%	7.167		
		96.4%	14.982		
		92.8%	11.235		
		98.3%	9.297		
		97%	9.8		
91.1%	14.998	92%	5.917	91.1%	14.498
98%	12.89	99%	6.118	98%	12.964
99%	11.663	96%	5.39	99%	11.93

**Tabla 9.8: Segunda parte de la tercera tabla de resultados: Elaboración propia**

Fue porque los sets de datos 8, 9 y 10 no requieren tantos metros para llegar a ser forzado. Por ejemplo, en la Tabla 8.8 el radio del Forzado en el set de datos N°8 es 20 metros y entre medio de esos 20 metros eran pocos puntos los que no se asocian. Por lo cual en la Tabla 9.8 quedaron desasociados los puntos GPS que tienen un radio mayor a 15 metros, ya que ese es el radio que se está utilizando en el Forzado con ruteo (Ver Tabla 9.7).

En el Dinámico con el Forzado con ruteo mantiene la misma relación con el Forzado con ruteo solo de la primera parte de la tercera tabla. Al igual que Todas las mejoras con el Heading con Forzado con ruteo.

Y si comparamos el tiempo de las combinaciones es como el más neutral obtenido de las 3 tablas.

La tercera parte de la última tabla que es el cuantas veces se llama a la función Route solver Tabla 9.9 donde siempre el menor que tiene es el Base junto con el Dinámico porque no alcanzan todos los puntos y no es necesario que entregue una lista de rutas por lo tanto coincide que tiene los menores tiempos. El que pasa por mayor cantidad es en Todas las mejoras y Forzado con ruteo, ya que como fue explicado anteriormente es porque verifica muchas y descarta las calles que no están en la tolerancia de grados. Y el más neutral es el Forzado con ruteo y el la combinación con el Dinámico y Forzado con ruteo, donde coincide con el tiempo y solución que son los más neutrales al igual.

Lo que también pasó en las 3 tablas fue que sí el Heading hace subir más el contador del Route solver ¿Por qué en el Heading solo no supera el Forzado con ruteo por ejemplo o al Dinámico o al Base?

La respuesta es simple por el mismo hecho que el Heading trabaja con el radio de Base por ejemplo 5 metros que es en este caso los puntos que para encontrar alguna



calle para asociarse sea más de 5 metros, no pasa por esta función Route solver porque no tiene que calles comprobar. Mientras que el Forzado con ruteo alcanza para todos los puntos o en este caso de la tercera tabla, la mayoría.

Route solver						
Base	Dinámico	Forzado con ruteo	Heading	Todas las mejoras	Dinámico y forzado con ruteo	Forzado con ruteo y heading
159	207	211			208	
142	199	198			202	
97	121	134			136	
176	235	267			321	
199	235	244			243	
165	203	183			189	
171	197	208			207	
108	126	130	120	161	135	132
89	103	122	94	139	116	145
69	97	103	75	116	101	126

**Tabla 9.9: Tercera parte de la tercera tabla de resultados: Elaboración propia**

El diagrama resultante reflejada en la figura 10 es la agregación de las tres mejoras propuestas que son radio Dinámico que aumenta gradualmente hasta cierto límite de metros, Forzado con ruteo que fuerza los puntos hacia una calle y comprueba ruteo, Heading que compara la direcciones de la calle y la dirección del automóvil, por ejemplo. Y condiciones que estaban insertadas en estudios previos. Por último, al hacer el ruteo “ki->kj” se agregó dentro de la función el Heading para obtener los dos ángulos de la calle y una nueva condición al ruteo de tolerancia de heading que compara los dos ángulos de la calle versus el sentido del punto GPS que fue agregado manualmente.

El proceso del algoritmo inicial antes de empezar título tiene la misma idea que al que se llegó como resultado final. Lo único que cambian son las mejoras implementadas y la función de “KiBetweenKj” que esta después de cada vez que se asoció o no se asoció un punto, viendo por consiguiente sí hay entre Ki y Kj puntos sin asociar y ver la oportunidad de asociarlos.

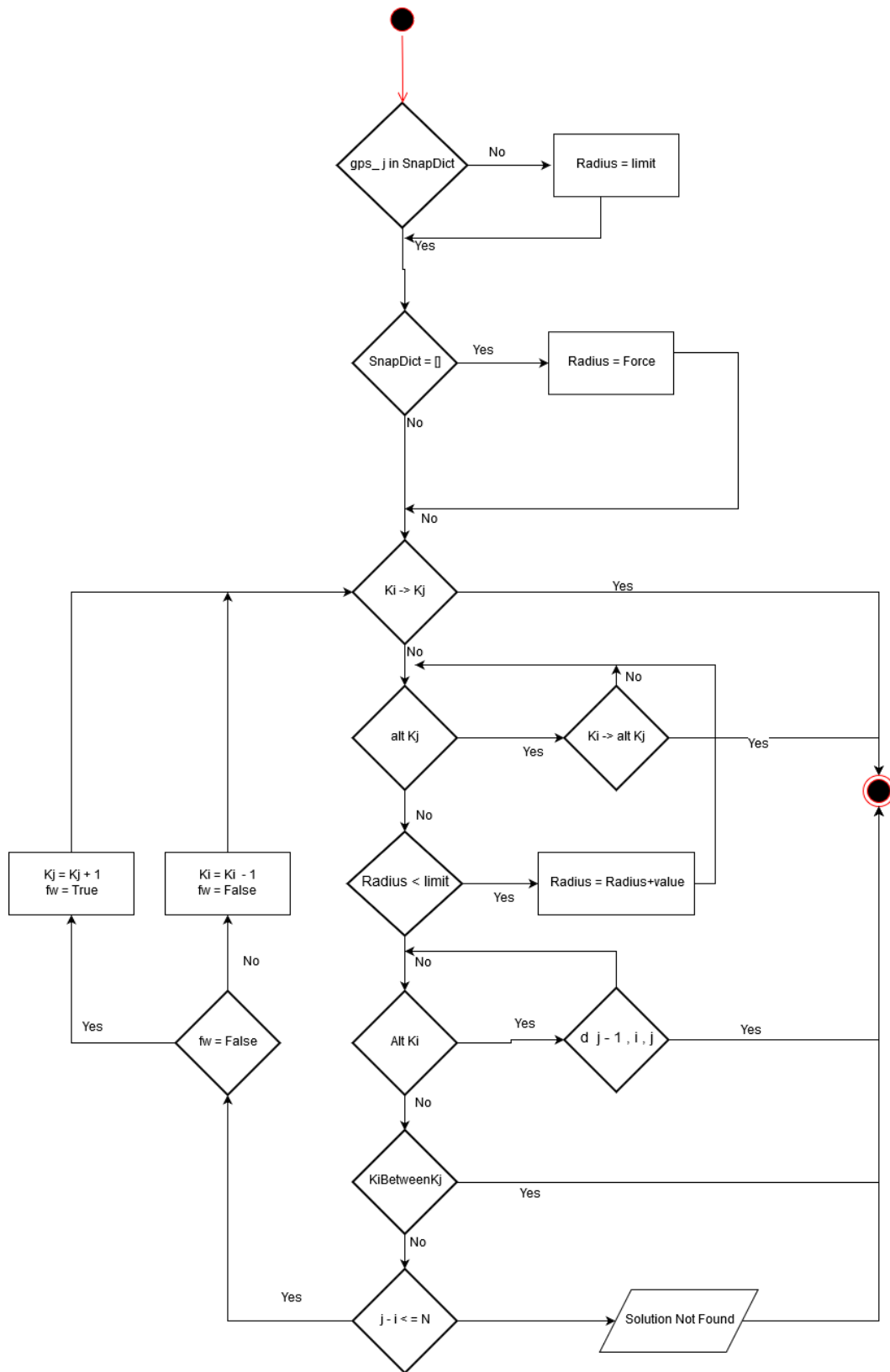


Figura 10: Diagrama Map-matching resultante: Elaboración propia

## Conclusión

El algoritmo en que se está trabajando, puede ser utilizado en un gran ámbito de vehículos, conociendo los puntos GPS cercanos a la red vial, es decir todo este sistema no es trabajado en tiempo real ya que requiere los puntos GPS total.

Una de las mayores complejidades ha sido compatibilizar el sistema de coordenadas en ARCGIS y adaptar toda la información de Renca, Santiago al sistema MMA.

Y se vio claramente que las mejoras se lograron con éxito marcando su diferencia en resultado positivo, comparado con el sistema base.

### Comenzar a hacer:

Investigar, analizar y desarrollar nuevas mejoras en el algoritmo existente, con respecto al posicionamiento correcto de los puntos GPS, ya que se están asociando todos a una calle no quiere decir que este correctamente asociado pero cada vez más preciso que antes en el resultado.

### Mas de:

Lograr ser más ordenado con las ideas y datos al momento de implementar la mejora y al de redactarlo en la memoria.

### Seguir haciendo:

Trabajar todos los días un poco para descubrir los fallos o problemas con el mayor tiempo de anticipación para resolverlos y seguir los pasos de la metodología.

### Menos de:

Dejar de usar malas prácticas al momento de implementar las mejoras y ser desordenado en estos.

### Dejar de hacer:

Almacenar códigos localmente sin un respaldo.

## Bibliografía

- A.Quddus, M. (2007). *Current map-matching algorithms for transport applications*. Retrieved from Current map-matching algorithms for transport applications:  
<https://www.sciencedirect.com/science/article/pii/S0968090X07000265>
- al, C. B. (2012). *An Instance-Specific Parameter Tuning Approach Using Fuzzy Logic for a Post-Processing Topological Map-Matching Algorithm*. Viña del mar.
- Bang, Y., Kim, J., & Yu, K. (2016). *An Improved Map-Matching Technique Based on the Fréchet Distance Approach for Pedestrian Navigation Services*. Seoul 08826, Korea.
- Brunsdon, C. (2004). *A test-bed simulator for GPS and gis integrated navigation and positioning research*. Retrieved from A test-bed simulator for GPS and gis integrated navigation and positioning research: [https://www.researchgate.net/publication/228777761\\_A\\_test-bed\\_simulator\\_for\\_gps\\_and\\_gis\\_integrated\\_navigation\\_and\\_positioning\\_research-bus\\_positioning\\_using\\_gps\\_observations\\_odometer\\_readings\\_and\\_map](https://www.researchgate.net/publication/228777761_A_test-bed_simulator_for_gps_and_gis_integrated_navigation_and_positioning_research-bus_positioning_using_gps_observations_odometer_readings_and_map)
- Duggan, N. (2017). *Making Height maps*. Retrieved from Making Height maps:  
<https://www.xyht.com/spatial-itgis/making-heightmaps/>
- GISGeography. (2018). *GPS Accuracy: HDOP, PDOP, GDOP, Multipath & the Atmosphere*.
- Hexagon. (2018). *Step 2 - Propagation*. Retrieved from Step 2 - Propagation::  
<https://www.novatel.com/an-introduction-to-gnss/chapter-2-basic-gnss-concepts/step-2-propagation/>
- Karimi, H. A. (2006). *GPS accuracy estimation using map matching techniques: Applied to vehicle positioning and odometer calibration*. Retrieved from GPS accuracy estimation using map matching techniques: Applied to vehicle positioning and odometer calibration:  
[https://www.researchgate.net/publication/223667159\\_GPS\\_accuracy\\_estimation\\_using\\_map\\_matching\\_techniques\\_Applied\\_to\\_vehicle\\_positioning\\_and\\_odometer\\_calibration](https://www.researchgate.net/publication/223667159_GPS_accuracy_estimation_using_map_matching_techniques_Applied_to_vehicle_positioning_and_odometer_calibration)
- Ponce, Á. (2018). *Heading*. Viña del Mar.
- Tinkering, W. (2018). *Getting from Point A to Point B*. Retrieved from Getting from Point A to Point B:  
<https://sites.google.com/site/wayneholder/self-driving-rc-car/getting-from-point-a-to-point-b?tmpl=%2Fsystem%2Fapp%2Ftemplates%2Fprint%2F&showPrintDialog=1>