



INAOE

Mecanismo para la generación de mapas de calles georeferenciados y con información contextual

por

Saúl Alejandro Romero Barrueta

Tesis sometida como requisito parcial para obtener el
grado de

**MAESTRO EN CIENCIAS EN EL ÁREA DE
CIENCIAS COMPUTACIONALES**

en el

**Instituto Nacional de Astrofísica, Óptica y
Electrónica**

Agosto 2012

Tonantzintla, Puebla

Supervisada por:

Dr. Saúl Eduardo Pomares Hernández, INAOE

Dra. María del Pilar Pilar Gómez Gil, INAOE

©INAOE 2012

El autor otorga al INAOE el permiso de reproducir y
distribuir copias en su totalidad o en partes de esta tesis



Resumen

Este trabajo de investigación propone un método para la generación de mapas georeferenciados con información contextual desde un enfoque de sensado móvil. Esto significa que se utilizan las capacidades de distintos sensores equipados en dispositivos móviles (como los teléfonos inteligentes) para extraer información del contexto de un individuo. Esta información además puede combinarse para obtener información que no podría conseguirse de manera aislada.

La motivación detrás de esto es de múltiples orígenes. Principalmente, el segmento de sensado móvil en la investigación es incipiente, y previos trabajos han propuesto la búsqueda de métodos de sensado híbridos para la obtención de información contextual. Por otra parte, la aplicación de este tipo de métodos ofrece un alternativa para la generación de mapas, así como la obtención de información de características de los lugares referenciados por los mapas. La fortaleza de estas alternativas reside en el creciente número de dispositivos móviles en los que este tipo de métodos es aplicable, así como el creciente número de sensores de los que disponen, su poder de cómputo y de la infraestructura que los soporta.

Los resultados obtenidos demuestran que la combinación de sensado de acelerometría y localización permiten obtener un mejor trazado de mapas respecto a los métodos que solamente utilizan localización, mientras que la tipificación de terrenos mediante el uso de redes neuronales alcanza un 80% de precisión. Finalmente, se propone trabajo futuro según las oportunidades de mejora de este proyecto y desde la perspectiva de los sistemas distribuidos.

Índice general

Índice general	III
Acrónimos	VII
1. Introducción	1
1.1. Motivación	1
1.2. Descripción del problema	2
1.2.1. Generación de mapas	2
1.2.2. Tipificación del terreno	3
1.3. Objetivos	3
1.3.1. Objetivo general	3
1.3.2. Objetivos específicos	3
1.4. Solución propuesta y contribución	3
1.5. Organización del documento	4
2. Fundamentos teóricos	5
2.1. Sistemas conscientes del contexto y geolocalización	5
2.1.1. El contexto	5
2.1.2. Servicios conscientes del contexto y servicios basados en localización.	7
2.1.3. Geolocalización y el sentido de la localización	7
2.1.4. GPS	9
2.1.5. Sistemas de información geográfica	10

2.1.6.	Sensado con teléfonos móviles.	10
2.1.6.1.	La arquitectura del sensado móvil	10
2.2.	Procesamiento de señales y aprendizaje automático.	11
2.2.1.	Procesamiento de señales y preprocesamiento de datos	11
2.2.1.1.	Agrupamiento de muestras en ventanas	13
2.2.1.2.	Transformada Wavelet Discreta	15
2.2.2.	Aprendizaje automático y redes neuronales artificiales	15
2.2.3.	El perceptrón y la red perceptrón multicapa	17
3.	Estado del arte	21
3.1.	Generación de mapas para la representación del conocimiento	22
3.1.1.	Posicionamiento y mapeo de interiores	22
3.1.2.	Posicionamiento y mapeo de exteriores	23
3.2.	Generación automática de mapas para sistemas de información geográfica y transporte inteligente.	25
3.2.1.	Procesamiento de imágenes satelitales	25
3.2.2.	Minería de datos	26
3.2.3.	Análisis de vectores de movimiento	27
3.3.	Extracción de información contextual en caminos utilizando acelerometría.	29
3.3.1.	Frameworks o marcos de trabajo para la clasificación de movimien- tos utilizando acelerometría.	29
3.3.2.	Reconocimiento de actividades usando acelerometría.	30
4.	Generación de mapas a partir de información de localización y acele- rometría.	33
4.1.	Introducción.	33
4.2.	Modelo del sistema.	33
4.3.	Análisis de la información proporcionada por los sensores	35
4.3.1.	Análisis de la información porporcionada por el aGPS	35

4.3.2.	Análisis de la información proporcionada por el acelerómetro . . .	41
4.4.	Detección de inconsistencia de trazos aGPS	44
4.4.1.	Algoritmo de detección de inconsistencia	44
4.5.	Corrección de trazos aGPS con acelerometría.	45
4.5.1.	Algoritmo de corrección de trazos en rectas	45
4.5.2.	Algoritmo de corrección de trazos en curvas	46
4.6.	Ejemplo de funcionamiento del algoritmo	46
4.7.	Resultados experimentales	57
4.7.1.	Configuración del experimento	58
4.8.	Análisis de resultados	64
4.9.	Conclusiones	66
5.	Tipificación del terreno.	71
5.1.	Introducción.	71
5.2.	Modelo del sistema	71
5.3.	Análisis de la información proporcionada por el acelerómetro	72
5.4.	Determinación de las características	74
5.5.	Determinación de las clases	76
5.6.	Construcción de un clasificador	77
5.7.	Ejemplo de funcionamiento del clasificador	78
5.8.	Resultados experimentales	79
5.8.1.	Configuración del experimento	80
5.8.2.	Hardware y software	80
5.8.3.	Vehículo	80
5.8.4.	Configuración del acelerómetro	81
5.8.5.	Recolección de datos	81
5.8.6.	Preprocesamiento de datos	82
5.8.7.	Clasificación	83
5.9.	Análisis de resultados	83

5.10. Conclusiones	84
6. Conclusiones y trabajo futuro	85
Bibliografía	87

Acrónimos

La notación utilizada en este documento se muestra en el cuadro 1.

Cuadro 1: Notación general

Símbolo	Significado
aGPS	GPS asistido. La red (internet) asiste al GPS en su inicialización.
GPS	Sistema de posicionamiento global.
DDTW	Desplazamiento derivativo de tiempo discreto (Derivative Discrete Time Warping).
DWT	Transformada discreta wavelet.
FFT	Transformada rápida de Fourier.
Fix	Coordenadas geográficas proporcionadas por el sistema GPS.
GIS	Sistema de información geográfica.
LBS	Sistemas basados en localización.
UTM	Proyección transversal universal de Mercator, por sus siglas en inglés.
SIG	Sistema de Información geográfica

Capítulo 1

Introducción

1.1. Motivación

Sensado móvil, o *mobile sensing*, es una disciplina emergente dentro del área de cómputo ubicuo [CK] que explora las capacidades de sensado de los dispositivos móviles y presenta muchas propuestas interesantes para resolver los llamados “problemas de las grandes ciudades” como son el monitoreo y el control de tráfico, la respuesta rápida a accidentes y la protección civil. Para traer a la realidad este conjunto de posibilidades, deben superarse algunos retos como son el manejo de la privacidad de los usuarios y el procesamiento de datos ruidosos recolectados por los dispositivos [LML⁺10].

Además de estos retos, la investigación reciente también señala la importancia de desarrollar sistemas de sensado híbrido, tal como *Master Location System* [RLUC], para mejorar la precisión y confiabilidad de los servicios que reportan alguna característica de interés del entorno. Gran parte del trabajo relacionado a esta área está motivado por esta necesidad de encontrar métodos de fusión de sensores, aplicados a la extracción de características del terreno en una carretera y a la mejora de los actuales sistemas de generación automática de mapas.

De forma complementaria, la explosión reciente en el uso de dispositivos móviles [LML⁺10] pone a la disposición de los investigadores y desarrolladores una gran base de usuarios de los cuales se pueden obtener volúmenes importantes de datos, sin mencionar la creciente capacidad computacional y de sensado de los dispositivos móviles modernos. Éste último factor no solamente sugiere facilidad para realizar pruebas en grupos grandes

de voluntarios, sino que apunta a la posible implementación de servicios y protocolos en aplicaciones útiles en un futuro cercano.

1.2. Descripción del problema

La generación de mapas es una actividad que recibe mucha atención en la actualidad. Como lo sugieren los datos del INEGI [INE00], es notorio el interés de las entidades competentes por la utilización de técnicas de automatización en la generación de mapas debido a su incursión en los SIG. En cuanto a la tipificación de terrenos, automatizar de igual manera la extracción de características puede facilitar el proceso de generación de mapas en términos de tiempo y de recursos empleados.

Este trabajo de tesis considera la generación de mapas y la tipificación de las calles a partir de información contextual de localización y aceleración. Se considera la participación de un solo dispositivo móvil, equipado con un acelerómetro triaxial y un aGPS. Este problema se halla en la intersección de algunas disciplinas como son el procesamiento de señales, el sensado móvil, el aprendizaje automático y los servicios conscientes del contexto. Para facilitar la investigación en torno a este problema, lo dividimos en dos partes: generación de mapas y tipificación de terreno, las cuales se explican a continuación.

1.2.1. Generación de mapas

Se considera el problema de trazar el mapa de una calle utilizando un algoritmo de recolección de puntos de ubicación o *fixes* (posiciones reportadas por el sistema) conocidos y que contiene errores derivados de las limitaciones del aGPS. Estos errores deben ser detectados y corregidos utilizando la información de otro sensor, como es el acelerómetro. Este problema se ataca bajo un enfoque de *colaboración y conflicto* para determinar cuál es el sensor con la información más confiable.

1.2.2. Tipificación del terreno

La segunda parte del problema consiste en la extracción de información del entorno a partir de datos de aceleración. De manera concreta, esta información es el *tipo* de terreno sobre el cual un automóvil se desplaza. El problema se ataca a través de clasificación de datos utilizando aprendizaje automático, consiguiendo resultados prometedores. Las señales a clasificarse se obtienen de los mismos sensores que se utilizan para asistir la generación de mapas.

1.3. Objetivos

1.3.1. Objetivo general

Diseñar un mecanismo para la generación de mapas de calles extendidos con información contextual.

1.3.2. Objetivos específicos

1. Diseñar un algoritmo para trazar calles georeferenciadas.
2. Diseñar un algoritmo para obtener información contextual a partir de datos GPS.
3. Diseñar un algoritmo para obtener información contextual a partir de datos del acelerómetro.

1.4. Solución propuesta y contribución

Para la generación del mapa en este trabajo, la solución gira en torno a aplicar técnicas de *integración de rutas* y la validación de información de distintas fuentes.

Integración de rutas consiste en calcular la posición del móvil de interés a partir de una posición conocida además de su curso y velocidad. La validación de información considera una fase de detección de puntos inconsistentes y un fase de corrección de trazos

utilizando un enfoque geométrico. Un punto inconsistente es aquel en el que la información de posicionamiento reportada por el receptor aGPS y el acelerómetro no coinciden. Esta inconsistencia tiene su origen en los errores ocasionales del sistema GPS, derivados de la atenuación de la señal de los satélites o de su falta de disponibilidad. El algoritmo propuesto, que considera los aspectos de detección de inconsistencias en los puntos GPS y su corrección constituye una aportación en el trazado de mapas automatizado.

En cuanto a la obtención de la información contextual, se utiliza un acelerómetro en conjunto con técnicas de aprendizaje automático para distinguir el estado de las calles. Específicamente, se eligió utilizar una red neuronal multicapa para la clasificación de los datos obtenidos, además se aplicaron algunas técnicas de preprocesamiento y transformación de las señales a dominios más aptos para la extracción de características. Por esto, el mecanismo de tipificación de terrenos es una aportación en el estudio de la obtención de información contextual.

Atacar el problema por estos dos flancos proporciona, respectivamente, dos componentes que pueden utilizarse de manera independiente como punto de partida para futura investigación o para la integración en una aplicación o servicio.

1.5. Organización del documento

Esta tesis está dividida en 6 capítulos; el capítulo 1 abarca la motivación, problemática, solución y metodología. En el capítulo 2, se exponen los fundamentos teóricos que soportan este trabajo. Durante el capítulo 3, el trabajo relacionado a generación automática de mapas y tipificación del terreno son discutidos.

En el capítulo 4 se propone el método para trazar caminos a partir de puntos GPS e información de un acelerómetro, de manera similar al capítulo 5, que trata del método para tipificar el terreno de un camino con un acelerómetro. Los capítulos 4 y 5 incluyen una sección de resultados experimentales. Finalmente, en el capítulo 6 se discuten las aportaciones y el trabajo futuro propuesto.

Capítulo 2

Fundamentos teóricos

Los conceptos fundamentales implicados en este trabajo de tesis abarcan áreas de las ciencias computacionales como el cómputo ubicuo y el aprendizaje automático, además de mantener una relación con otras áreas del conocimiento como la geodesia y la cartografía. Estos conceptos se presentan agrupados en las categorías de *Sistemas conscientes del entorno y aprendizaje automático y procesamiento de señales*, especialmente construidas bajo el enfoque que se le ha dado al problema a resolver, presentándose a continuación.

2.1. Sistemas conscientes del contexto y geolocalización

2.1.1. El contexto

El concepto de contexto es vago por su naturaleza. Todo en el mundo ocurre dentro de cierto ámbito, pero es posible acotarlo al campo de acción de nuestro interés, que en este caso es el cómputo ubicuo. Una definición que satisface nuestra necesidad es proporcionada por Chen y Kotz [CK]:

“Contexto es el conjunto de estados y configuraciones ambientales que ya sea determinan el comportamiento de una aplicación o en que un evento de la aplicación ocurre y es interesante para el usuario.”

Axel Küpper en su texto *“Location Based Services, Fundamentals and Operation”* [Axe05] distingue dos niveles de contexto, primario y secundario, definidos de la siguiente manera (ver figura 2.1) .

- **Contexto primario:** toda la información que puede ser capturada por sensores. Ejemplos de este tipo de contexto son la identidad, localización, intensidad de la luz, tiempo, presión atmosférica y humedad entre otros.
- **Contexto secundario:** también conocido como de alto nivel, es la información que se puede inferir a partir de la mezcla o filtrado del contexto primario. Un ejemplo de esto es el contexto espacial, que resulta de combinar información de localización e identidad.

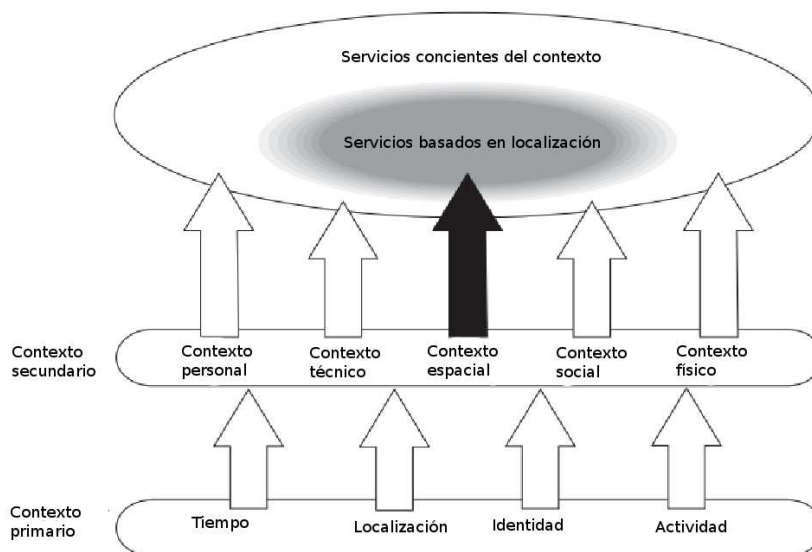


Figura 2.1: Niveles de contexto. [Axe05]

2.1.2. Servicios conscientes del contexto y servicios basados en localización.

Habiendo definido el contexto, podemos exponer el concepto de los servicios conscientes del contexto, que son los servicios o aplicaciones que pueden adaptar su comportamiento a alguno o varios parámetros que reflejan el entorno del usuario.

Por su parte, los servicios basados en localización o *LBS* por sus siglas en inglés, se consideran como un subconjunto especializado de los servicios conscientes del contexto, siendo el contexto espacial su especialidad. Como ejemplo de un LBS tenemos aplicaciones como *Google Latitude* [Lat] o *Foursquare* [Fou]. Ambos explotan el contexto espacial, siendo el objetivo de *Google Latitude* la interacción con contactos cercanos al usuario y siendo el objetivo de *Foursquare* la competencia entre usuarios de la aplicación en la exploración de los sitios interesantes de un mapa.

2.1.3. Geolocalización y el sentido de la localización

Apoyándonos en el análisis de los sistemas de geolocalización existentes [RGWNSM07], podemos definir la geolocalización como la actividad de ubicar la localización espacial de un objeto de interés dentro del marco de referencia del planeta Tierra. También se refiere a la localización espacial en sí dentro del mismo marco de referencia.

Es importante, entonces, definir qué es la localización y cómo se expresa. Existen distintos tipos de localización, como son la *localización descriptiva*, que es la localización en términos coloquiales: referencias de lugares conocidos. Este tipo de localizaciones son los que utilizamos en situaciones cotidianas, como por ejemplo, quedar de acuerdo para una cita “en la cafetería de la universidad”. También existen *localizaciones de red*, que se refieren a la topología de una red informática, como puede ser la red GSM o la jerarquía de la red IP.

Pondremos especial atención en lo que llamamos *localización espacial*, o posición. Este tipo de localización está expresada por vectores de números, representando un punto único en el espacio Euclidiano. Este concepto provee la base para efectuar procesamiento

de la información espacial en un sistema automático y es a la vez un puente para establecer equivalencias entre los tipos de localización. La localización espacial, explotada por los LBS, está siempre expresada con sistemas de referencia bien definidos, que divide alguna región de la Tierra en unidades de forma y tamaño comunes. Este sistema de referencia consiste de los siguientes elementos:

1. Un sistema de coordenadas: utilizado para representar o referenciar de manera única a algún punto o lugar, mediante un vector de números (ver figura 2.2). Estos sistemas de coordenadas pueden ser cartesianos o elipsoidales y se definen por:
 - a) **Un origen:** Que es la intersección entre ejes.
 - b) **Una escala:** La división de los ejes en unidades comunes.
 - c) **Orientación:** Que fija la dirección de los ejes en el espacio.

2. Un *datum*: Que es la definición del tamaño y la forma de la Tierra, y el origen y la orientación del sistema de coordenadas respecto a la Tierra. En la práctica, el sistema GPS calcula las posiciones desde el centro de la tierra, y usando un datum llamado *WGS-84 ellipsoid* [Axe05], lo referencia a longitud, latitud y altitud con respecto al elipsoide mencionado.

3. Una proyección de mapa: después de obtener una referencia en la superficie de la tierra, es necesario representarla en un espacio bidimensional (como un mapa o una pantalla) Esto es precisamente de lo que se encarga una proyección. Debemos considerar que las proyecciones en un mapa siempre acarrearán algún grado de distorsión. Entre los tipos de proyecciones de la superficie terrestre tenemos las cónicas, cilíndricas y planas. Para nuestro trabajo se utiliza la proyección UTM que es una proyección de tipo cilíndrica y que está rotada 90° respecto al eje de rotación de la tierra [Axe05].

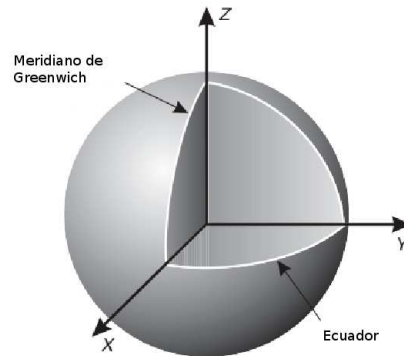


Figura 2.2: Sistema de coordenadas centrado en la tierra [Axe05].

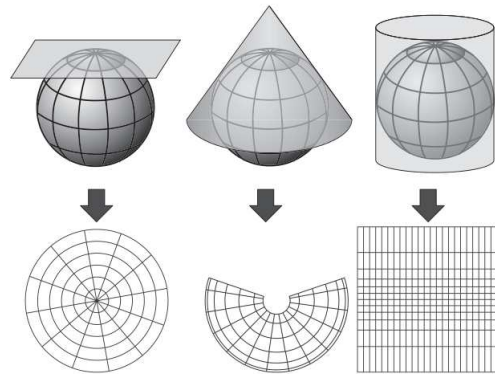


Figura 2.3: Tipos de proyecciones [Axe05].

2.1.4. GPS

El Sistema de Posicionamiento Global, o *GPS* por siglas en inglés, es un servicio a escala mundial que utiliza una red de satélites como referencia fija para determinar la localización espacial de un receptor terrestre. Para lograr un *fix*, es necesario que el receptor esté en la línea de visión de al menos cuatro satélites.

Entre la información que ofrece el sistema GPS, además de la información de localización en la superficie terrestre, tenemos que es capaz de reportar velocidad y dirección del movimiento. Entre sus limitaciones está que la señal de los satélites está considerablemente atenuada al llegar a la superficie terrestre, haciéndolo especialmente sensible a los efectos de reflexión y bloqueo de señal por estructuras urbanas, que se encuentran en gran cantidad en las ciudades. Aunque estas limitaciones han sido reducidas gracias

a la introducción del *aGPS*, cabe señalar que este sistema de asistencia al GPS requiere de tener acceso a Internet.

2.1.5. Sistemas de información geográfica

Los Sistemas de información geográfica o *GIS*, por sus siglas en inglés, son sistemas cuyo propósito es capturar, analizar y manipular información geográfica. Aunque tradicionalmente, sus fuentes de datos han sido la captura de mapas con programas estilo CAD o análisis de imágenes satelitales, las tendencias actuales y su estrecha relación con los LBS han establecido un marco de cooperación con dichas aplicaciones.

2.1.6. Sensado con teléfonos móviles.

Se refiere a un paradigma emergente de sensado del contexto [LML⁺10], cuyo objetivo es recolectar información del ambiente a través de los sensores presentes en los dispositivos inteligentes actuales. Su principal fortaleza es el potencial volumen de datos al que se tiene acceso al convertir en objetos de uso común a los sensores de intensidad luminosa, aceleración, proximidad, imagen, sonido y señales de radiofrecuencia, además de su conectividad a internet.

2.1.6.1. La arquitectura del sensado móvil

Aunque no existe un consenso establecido para describir la arquitectura de sensado móvil actual [LML⁺10], se distinguen tres bloques esenciales que lo componen como un proceso, como se ilustra en la figura 2.4:

1. Sensado: la información recogida por cada móvil.
2. Aprendizaje: el procesamiento de los datos, sin importar si es en el móvil o en un servicio distribuido en la nube.
3. Compartir: La distribución de los resultados, que puede ser a nivel personal o grupal.

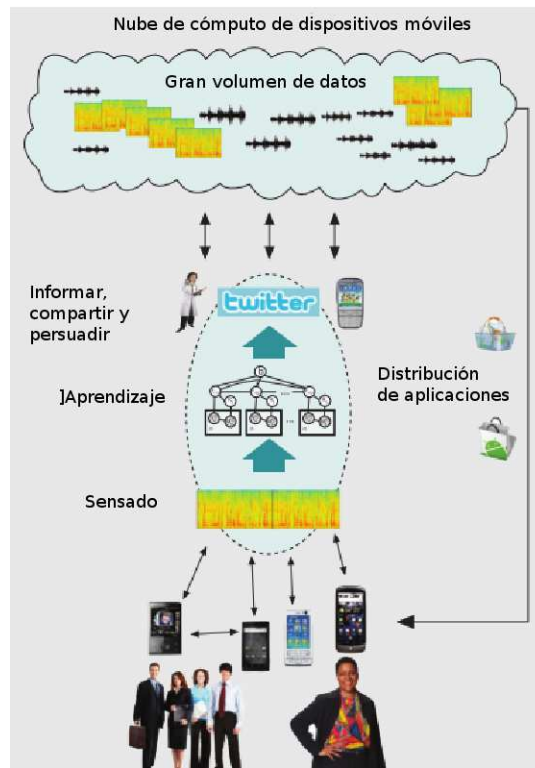


Figura 2.4: Arquitectura del sensado móvil.[LML⁺10]

2.2. Procesamiento de señales y aprendizaje automático.

En muchas aplicaciones de aprendizaje automático se capturan señales del entorno, que son después analizadas para la extracción de características y entrenamiento de un sistema que tendrá algún objetivo específico, como la clasificación de futuras señales que serán presentadas al sistema y que es el caso de este trabajo. A continuación se exponen los conceptos relacionados a este tema.

2.2.1. Procesamiento de señales y preprocesamiento de datos

El procesamiento de señales es la disciplina que trata con la representación, transformación y manipulación de las señales, así como de la información que contienen [Opp]. Además de su relevancia en este trabajo de investigación, su aplicación en sistemas modernos de telecomunicaciones, entretenimiento y de cómputo le hacen de gran importancia para la vida cotidiana.

En esta sección se revisan los conceptos de señal, frecuencia de muestreo, transformación y preprocesamiento de datos por ser esenciales en el desarrollo de esta tesis.

Señales y sistemas discretos en el tiempo

El término de señal se aplica a la representación del estado de un sistema físico. Esta señal, además de ser recogida del sistema, puede sintetizarse para transmitir información. La representación matemática de una señal es una función de una o mas variables independientes, siendo la convención establecer la variable independiente como el tiempo. [Opp].

Una *señal continua* o *señal análoga* está definida en un tiempo continuo y está representada por una variable independiente continua. En contraparte, una *señal discreta* está definida en tiempos discretos y está representada por una variable independiente discreta, en otras palabras, una secuencia de números. Además de la variable independiente, también la amplitud de la señal puede ser continua o discreta, teniendo las *señales digitales* una amplitud y variable independiente discretas.

Una señal discreta se define por una secuencia de números x en la cual el n ésimo número de la secuencia se denota $x[n]$, que formalmente se expresa como:

$$x = x[n], \quad -\infty < n < \infty \quad (2.1)$$

Donde n es un entero. En la práctica, estas señales se extraen de un muestreo periódico de una señal continua. Para este caso, el valor del n ésimo número en la secuencia es igual al valor de la señal análoga $x_a(t)$ en el tiempo nT , en otras palabras:

$$x[n] = x_a(nT), \quad -\infty < n < \infty \quad (2.2)$$

Donde la cantidad T es el *periodo de muestreo*, y su recíproco es la *frecuencia de muestreo*. Esta cantidad afecta sensiblemente a la cantidad de información que puede representarse en una señal digital, y debe considerarse siempre con detenimiento. Un

ejemplo de una representación de una señal discreta en el tiempo se muestra en la figura 2.5.

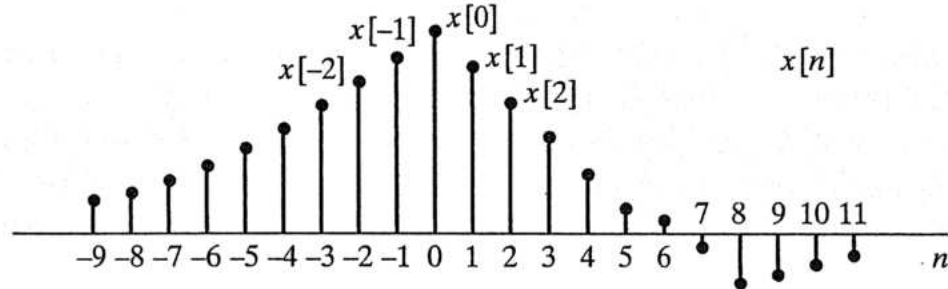


Figura 2.5: Representación gráfica de una señal discreta en el tiempo. [Opp]

Teorema de Nyquist-Shannon

Otro concepto clave para la captura de señales que serán después utilizadas en sistemas de aprendizaje automático es el teorema de Nyquist-Shannon [Sha98], que establece:

Si una función $x(t)$ contiene frecuencias no mayores a B hertz, se puede reconstruir totalmente al dar sus ordenadas en series de puntos distanciados por $\frac{1}{2B}$ segundos.

La utilidad de este teorema emerge cuando necesitamos determinar la frecuencia de muestreo necesaria para poder hacer un análisis de frecuencia de la señal capturada.

De manera similar a la etapa de recolección de señales, es importante someter estos datos a un proceso de preparación antes de ser presentado a un sistema de aprendizaje automático.

Durante el desarrollo de este trabajo se utilizan las siguientes técnicas de preprocesamiento de datos.

2.2.1.1. Agrupamiento de muestras en ventanas

Para un sistema de aprendizaje automático, es importante obtener características significativas de los datos que le son presentados. En este trabajo, ésto se logra mediante

el agrupamiento de datos en ventanas de tiempo. Agrupar las muestras tomadas por sensores de esta manera permite obtener medidas estadísticas que representan cualidades de la señal sensada. Un ejemplo de este tipo de agrupamiento se muestra en la figura 2.6.

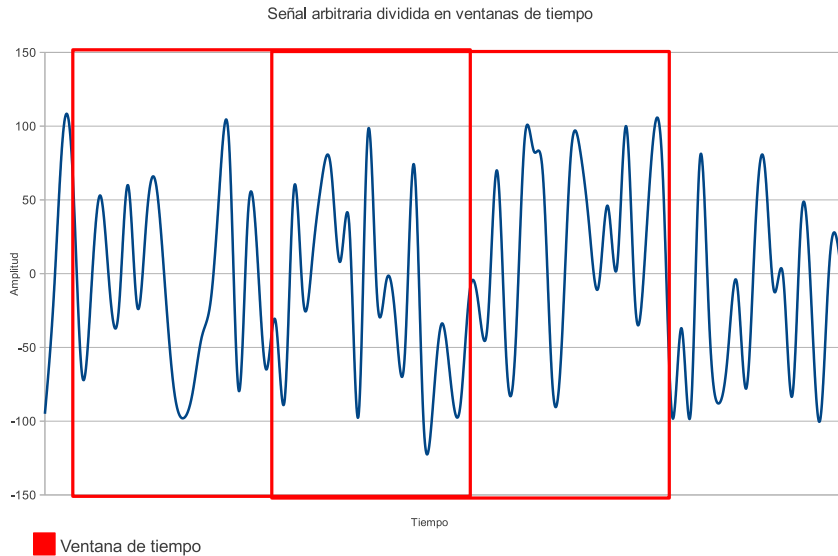


Figura 2.6: Ejemplo de división en ventanas de tiempo.

El tamaño de la ventana dependerá del problema a atacar, dependiendo de la información que se deba extraer de los datos. Aunque para el problema de la tipificación de terrenos no existe investigación publicada hasta la fecha sobre esto, sí existen recomendaciones basadas en las observaciones de otros trabajos para elegir tamaños de ventanas de tiempo para la extracción de atributos en el reconocimiento de actividades, por ejemplo [HS05]. Para el caso de la validación de puntos GPS, la ventana de tiempo está dada por el periodo de tiempo comprendido ente dos puntos GPS, puesto que no se busca la extracción de otros atributos más que el promedio de la señal en una sección de un recorrido.

Como señala el trabajo de [HS05], es recomendable definir un tamaño de ventana para cada tipo de actividad a reconocer, analizando cuidadosamente los atributos que desean

extraerse y el tipo de actividad a reconocer. Como ejemplo, para reconocer la actividad de saltar, se sugiere un tamaño de ventana de 4 segundos, mientras que para la actividad de estar en reposo (de pie) se sugiere una ventana de medio segundo. Para ambas actividades se considera una frecuencia de muestreo de 500hz. Revisando el ejemplo, podemos notar que actividades que pueden considerarse como “fuertes” requieren ventanas amplias en comparación con actividades “suaves” debido a su mayor varianza en sus características.

2.2.1.2. Transformada Wavelet Discreta

Un *wavelet*, en el ámbito de una transformada wavelet discreta, es una función ortogonal que puede aplicarse a un grupo finito de datos. Funcionalmente, es muy parecida a la transformada discreta de Fourier en su ortogonalidad, puesto que una señal que pasa dos veces por la transformación se mantiene sin cambios, y porque la señal de entrada debe ser un conjunto de muestras discretas en el tiempo. Ambas transformaciones son convoluciones [Tra92]. Sin embargo, son diferentes en el sentido de que la transformada wavelet puede capturar información de la frecuencia y su localización (localización en el tiempo).

El trabajo presentado aquí, utiliza la transformada wavelet perteneciente a la familia *daubechies* .

2.2.2. Aprendizaje automático y redes neuronales artificiales

Aprendizaje automático y algoritmos de aprendizaje

El aprendizaje automático es la adquisición de descripciones estructurales a través de ejemplos [MT]. Estas descripciones pueden utilizarse para tareas de predicción, clasificación y explicación. Esto se logra a través de diversos algoritmos de aprendizaje, que podemos clasificar principalmente en dos categorías [Hay98]:

1. Aprendizaje supervisado: el conocimiento a adquirir se codifica en pares de entradas y salidas deseadas para el sistema, llamado ejemplos. Estos ejemplos se presentan a la red, que recibe una señal de error que resulta de comparar la respuesta de

la red con la salida esperada. Esta señal se utiliza para ajustar los parámetros del sistema de aprendizaje y minimizar el error. Cuando este error alcanza un valor deseado, el sistema puede tratar el problema que se le presenta por sí solo.

2. Aprendizaje no supervisado: al contrario que el aprendizaje supervisado, no existe una entidad externa al sistema que evalúe la respuesta al estímulo presentado. Se distinguen dos divisiones de este paradigma :

a) Aprendizaje por refuerzo: el mapeo de una entrada a una salida se logra mediante la continua interacción con el entorno del sistema, minimizando uno índice de desempeño. Esto se logra construyendo el sistema alrededor de un *crítico*, que recibe una señal primaria de refuerzo del entorno y la convierte en una señal de mejor calidad para el sistema llamada señal heurística de refuerzo. El sistema observa un retraso temporal, que es necesario para la generación de la señal heurística de refuerzo. El objetivo principal de este tipo de aprendizaje es minimizar una función de costo acumulativa, que le permite al sistema descubrir qué acciones son las mejores para el aprendizaje del sistema. Este tipo de aprendizaje es difícil, porque no hay un agente externo que evalúe la respuesta en cada paso del proceso, y porque el retraso en la señal de refuerzo implica que el sistema debe ser capaz de discriminar las mejores acciones a través del tiempo que toma todo el proceso.

b) Aprendizaje auto organizado: adicionalmente a la falta de un supervisor, se elimina el crítico dentro del sistema. Para lograrlo, se utiliza una regla de aprendizaje competitiva. Esta regla de aprendizaje competitiva es una medida de la calidad de la representación del conocimiento que es independiente de la tarea que realiza la red. Cuando el sistema logra adaptarse a las regularidades estadísticas de los datos de entrada, el sistema puede codificar las características de las entradas y crear clases automáticamente.

En este trabajo se usan algoritmos de aprendizaje supervisado, por la facilidad que se tuvo para etiquetar los ejemplos para el sistema de reconocimiento.

Redes neuronales artificiales

Una definición de red neuronal, desde la perspectiva de una máquina adaptativa es [Hay98]:

Una red neuronal es un procesador masivamente paralelo y distribuído hecho de unidades simples de proceso que tienen una propensión natural a almacenar conocimiento experimental y hacerlo disponible par su uso. Se parece al cerebro en dos aspectos:

1. La red adquiere el conocimiento de su entorno a través de un proceso de aprendizaje
2. La fuerza de conexión entre las neuronas, conocida como pesos sinápticos, se usan para almacenar el conocimiento adquirido.

Es importante observar que entre las capacidades de las redes neuronales encontramos la predicción, la clasificación y la aproximación de funciones, y que estas capacidades hacen de las redes neuronales una valiosa herramienta para el aprendizaje automático.

Para ajustar los pesos sinápticos de una red neuronal se utiliza un algoritmo de aprendizaje, como los descritos en la subsección anterior.

2.2.3. El perceptrón y la red perceptrón multicapa

El perceptrón

El perceptrón es un caso especial de red neuronal: puede considerarse como una red de una sola capa y de un solo elemento de procesamiento. El perceptrón es capaz de clasificar correctamente en dos clases los estímulos aplicados en sus entradas si las clases son linealmente separables. Este consiste en un combinador lineal, que suma las entradas del neurón y en una función de activación, que limita su salida y permite la asignación de la clase.

La operación del combinador lineal está descrita por la siguiente sumatoria:

$$v = \sum_{i=1}^m w_i x_i + b \quad (2.3)$$

Donde los pesos sinápticos se denotan por w_1, w_2, \dots, w_m . De la misma forma, las entradas (el estímulo) aplicado al perceptrón se denotan por x_1, x_2, \dots, x_m . El sesgo externo se denota por b . El resultado v de esta suma se pasa a la función de activación, que puede ser una función escalón y cuya salida es 1 o -1 y corresponde a la clase elegida por el perceptrón 2.4.

$$\text{sgn}(v) = \begin{cases} +1 & \text{para } v > 0 \\ -1 & \text{para } v < 0 \end{cases} \quad (2.4)$$

La red de perceptrones multicapa

Entre las limitaciones del perceptrón, tenemos que solamente es capaz de resolver problemas de clasificación donde las clases son linealmente separables. Una red de perceptrones multicapa resuelve estas limitaciones gracias a las características que la definen [Hay98]:

1. Funciones de activación no lineales en sus neuronas, mismas que son diferenciables.
2. Una o mas capas de neuronas ocultas entre la entrada y la salida de la red.
3. Alto grado de conectividad.

Entre las funciones $\phi()$ de activación que podemos usar en una red perceptrón multicapa, tenemos la *función logística*, definida por:

$$\phi(v_j(n)) = \frac{1}{1 + \exp(-av_j(n))}, a > 0 \quad (2.5)$$

Donde $v_j(n)$ es el campo local inducido de la neurona j -ésima, y a es un parámetro positivo ajustable. Otra función usada como función de activación, es la *tangente hiperbólica*, definida por :

$$\phi(v_j(n)) = \text{atanh}(bv_j(n)) \quad (2.6)$$

Donde a y b son enteros positivos. Para ambos casos, $v_j(n)$ se define como:

$$y_j(n) = \sum_{i=0}^m w_{ji}(n)y_i(n) \quad (2.7)$$

Para el entrenamiento se utiliza el algoritmo de retropropagación que es un algoritmo de aprendizaje supervisado y se divide en dos fases principales: propagación de la señal de entrada, y propagación del error, para el ajuste de los pesos sinápticos. El algoritmo de retropropagación se describe a continuación [Hay98].

1. Inicialización: los pesos deben ser iniciados con valores aleatorios pequeños, poniendo atención en escoger pesos con una distribución uniforme, media en cero y con una varianza tal que la desviación estándar de los valores que se pasan a la función de activación sigmoide queden entre su parte lineal y estándar. TODO: Que es la parte estandar?
2. Presentación de los ejemplos de entrenamiento: Se le presenta a la red el conjunto de ejemplos de entrenamiento, y para cada uno de ellos, se ejecuta los siguientes pasos (3,4)
 - a) Propagación de la señal de entrada: Siendo un ejemplo de entrenamiento y su salida deseada $x(n)$ y $d(n)$ respectivamente. El ejemplo se presenta en la entrada de la red, y la salida deseada en la capa de salida de la red. Los campos localmente inducidos $v_j^l(n)$ para la neurona j y la capa l se calculan capa por capa como se expone en la ecuación (2.8). Para esta ecuación, $y_i^{l-1}(n)$ es la salida de la neurona i en la capa anterior $l - 1$ durante la iteración n , y $w_{ji}^{(l)}$ es el peso sináptico de la neurona j en la capa l que recibe información de la neurona i de la capa $l - 1$. Cuando $i = 0$, tenemos $y_0^{l-1}(n) = 1$ y $w_{j0}^{(l)}(n) = b_j^{(l)}(n)$ es el sesgo aplicado a la neurona j en la capa l . Asumiendo

el uso de la función sigmoide de transferencia, la salida de la neurona j en la capa l está dada por (2.9). Si la neurona j se encuentra en la primera capa escondida, entonces $y_j^0(n) = x_j(n)$, donde $x_j(n)$ es el j -ésimo elemento del vector de características. Si el neurón j está en la capa de salida, entonces $y_j^{(L)} = o_j(n)$. A continuación, se calcula la señal de error, $e_j(n)$ utilizando (2.10), donde $d_j(n)$ es el j -ésimo elemento del vector de la salida esperada.

$$y_j^l(n) = \sum_i w_{ji}^{(l)}(n) y_i^{l-1}(n) \quad (2.8)$$

$$y_j^l(n) = \phi_j(v_j(n)) \quad (2.9)$$

$$e_j(n) = d_j(n) - o_j(n) \quad (2.10)$$

b) Propagación de la señal de error: En esta fase, se calculan los gradientes locales, que están definidos por (2.11). La apóstrofe en $\phi'()$ denota diferenciación respecto al argumento. Después de obtener los gradientes, se puede aplicar la regla delta generalizada en la capa l , de acuerdo a (2.12), donde η es la tasa de aprendizaje, y α es la constante de momento.

$$\delta_j^{(l)}(n) = \begin{cases} e_j^{(L)}(n) \phi_j'(v_j^{(L)}(n)) & \text{para el neurón } j \text{ en la capa de salida } L \\ \phi_j'(v_j^l(n)) \sum_k \delta_k^{l+1}(n) w_{kj}^{l+1}(n) & \text{para el neurón } j \text{ en la capa oculta } l \end{cases} \quad (2.11)$$

$$w_{ji}^l(n+1) = w_{ji}^l(n) + \alpha[w_{ji}^l(n-1)] + \eta \delta_j^{(l)}(n) y_i^{(l-1)}(n) \quad (2.12)$$

3. Iteración: Se iteran los pasos 3 y 4 para cada ejemplo presentado a la red, hasta alcanzar el criterio de parada para el entrenamiento.

Capítulo 3

Estado del arte

Sensado móvil consciente del entorno, o *context aware mobile sensing* es un área relativamente nueva [CK] que , aunque últimamente ha recibido mucha atención a causa del actual florecimiento de los dispositivos móviles inteligentes y la extensión de sus capacidades, aún presenta grandes retos [CK] en términos de la infraestructura requerida para su aplicación, además de cuestiones de seguridad y privacidad.

A continuación se presentan los avances del área relevantes para la generación automática de mapas y obtención de información contextual de los caminos. Dichos avances se presentan agrupados bajo grandes categorías, consideradas según su objetivo:

1. Generación de mapas para la representación del conocimiento, en aplicaciones de robótica.
2. Generación de mapas para su uso en aplicaciones GIS, transporte inteligente o monitoreo de tráfico.

Habiendo establecido estas categorías, podemos considerar a la segunda como la categoría relevante para el presente trabajo de tesis, presentando a la primera con el objetivo de ofrecer una visión completa del trabajo relacionado a la generación automática de mapas.

3.1. Generación de mapas para la representación del conocimiento

Para algunas aplicaciones en el campo de la robótica, es necesario contar con una representación del entorno con el que interactúa el robot u agente en cuestión. Distinguimos el posicionamiento y mapeo de interiores además del posicionamiento y mapeo de exteriores como dos categorías, que se exponen a continuación.

3.1.1. Posicionamiento y mapeo de interiores

Bajo este enfoque se agrupan métodos para posicionar a algún agente en los confines de una ubicación cerrada, como una habitación u oficina.

Entre el trabajo más representativo, encontramos la Localización Topológica y Mapeo simultáneo [CN01], o SLAM por sus siglas en inglés *Topological Simultaneous Localization and Mapping*. Este trabajo propone un método para generar un mapa y ubicar un robot en el mismo, utilizando sensores de corto alcance y codificando el espacio explorado en un grafo de Voronoi generalizado (GVG). La utilidad de este grafo radica en que los nodos del GVG pueden utilizarse para la planeación de rutas en robots. Los resultados reportados por este trabajo demuestran que la efectividad de este acercamiento depende directamente de la riqueza topológica del área a mapear. Un escenario en el que el método propuesto no funcionaría correctamente es un cuarto de forma cuadrada debido a que la simetría del cuarto limita la información topológica que se puede codificar en el mapa.

Por su parte, en el artículo titulado “Indexado espacial para sistemas conscientes del contexto”, o *Spatial indexing for location aware-systems* [Har07], el autor propone una estructura de datos para representar la ubicación de sistemas conscientes del entorno que ya cuentan con información sobre el espacio en el que interactúan. La estructura utiliza una combinación de *quadtrees* y de *Rtrees* para acceder eficientemente a la información de localización e interactuar con el el entorno. Los *quadtrees* son árboles que codifican áreas de interés dividiendo el espacio en zonas cuadradas, de manera recursiva. Cada

nodo del árbol tiene cuatro hijos, que representan un cuarto del área de su nodo padre. Por su parte, los *Rtrees* son similares a los *quadtrees* por ser árboles que codifican un espacio, pero sus nodos representan zonas rectangulares y son árboles balanceados.

Como resultado del trabajo de indexado espacial para sistemas conscientes del contexto, los autores reportan su implementación en un *middleware* para sistemas conscientes del contexto.

3.1.2. Posicionamiento y mapeo de exteriores

En esta categoría se considera la generación de mapas para ubicar robots en exteriores, con el fin de habilitarles para navegar evitando obstáculos. Un ejemplo representativo es el propuesto por Triebel y Pfaff en el artículo titulado “Mapas de superficie multinivel para el mapeo de terrenos exteriores y cerrado de bucles”[TPB06] donde se propone una estructura de datos, al estilo de una rejilla bidimensional donde se almacena información de la elevación del terreno. Dicho mapa se denomina como *Multilevel Surface Map*, y puede ser actualizado utilizando información obtenida con sensores. La información almacenada en las celdas que componen este mapa es útil para determinar si un camino es apto para ser atravesado, y para cerrar bucles en caminos. Los resultados reportados por los autores demuestran que la representación propuesta es equivalente a otras representaciones del terreno, al mismo tiempo que ahorran espacio de almacenamiento de datos [TPB06].

Para finalizar esta sección de mapas para la representación del conocimiento, es importante señalar los aspectos que motivaron a separar el tema de generación de mapas en las dos categorías mencionadas en esta sección:

1. Alcance y campo de acción. Ya sea para aplicarse en interiores o exteriores, los mapas generados para la representación del conocimiento están localizados en áreas pequeñas, cuando se les compara con las áreas que cubren los sistemas GIS o de transporte inteligente. Esta condición establece premisas de trabajo muy distintas para las dos categorías de mapas propuestas, por ejemplo el número de partici-

pantes en el sensado, el tamaño del área a sensar o el esquema de comunicación necesario para los participantes en el sensado.

2. Actividades objetivo. En el caso de los métodos expuestos en esta sección, es notorio que los mapas son utilizados en actividades muy específicas, como la navegación automática, la interacción con algunos elementos del medio o la localización de personal en edificios. De manera opuesta, para un mapa que será utilizado en un sistema GIS o de transporte inteligente, es deseable que su contenido pueda ser utilizado como una fuente de datos para distintos propósitos. Derivado de este hecho, encontramos que la información que se utiliza en la representación del conocimiento difícilmente tiene significado fuera de la aplicación para la que utiliza.
3. Sensores utilizados: En las aplicaciones descritas en esta sección, se hace uso de sensores que no son de uso común, o *commodities*. Si bien se hace amplio uso de sensores de video, audio, localización, acelerometría (en el caso de los sistemas que utilizan técnicas de integración de rutas), señales de radio frecuencia como wi-fi e infrarrojo, es una práctica común la utilización de sensores ultrasónicos o láser, que no se encuentran en los aparatos de uso diario.
4. Modelo de distribución de tareas: Aunque existen propuestas en cuanto a modelos distribuidos [HH94], la mayor parte de las propuestas consideran arquitecturas centralizadas, o agentes individuales. Por su cuenta, las aplicaciones GIS y similares deben tratar con grandes volúmenes de datos y de agentes participantes. Aunque el presente trabajo de tesis no considera una arquitectura distribuida, ésta no se descarta como trabajo futuro. De la misma forma, las consideraciones de seguridad y privacidad son muy distintas entre estas dos categorías.

3.2. Generación automática de mapas para sistemas de información geográfica y transporte inteligente.

Los mapas resultantes de estos sistemas tienen un significado topográfico y apto para la interpretación de una persona. Es aquí donde cobra sentido el atributo de “georeferenciado” en estos mapas. También dentro de esta categoría distinguimos tres divisiones de los métodos propuestos:

1. El procesamiento de imágenes satelitales.
2. La minería de datos
3. El procesamiento de vectores de movimiento.

Estas subcategorías están definidas por las técnicas que utilizan para generar los mapas, y se exponen a continuación.

3.2.1. Procesamiento de imágenes satelitales

Esta técnica es ampliamente usada en aplicaciones SIG o GIS. Su objetivo es obtener imágenes satelitales del área de la que se busca obtener un mapa y aplicar diversas técnicas de procesamiento de imágenes. Como ejemplo de estos modelos tenemos el trabajo titulado “*El mapa del terreno del Reino Unido 2000: Construcción de un mapa de paquetes de vectores a partir de imágenes satelitales*”, [Fuller, Smith], que es una colección de mapas obtenidos de diversas regiones del Reino Unido a través del procesamiento de imágenes satelitales y que está disponible para el uso del público en general.

Por su parte, Cooper y Wood [Pau87] han propuesto un método para generar modelos digitales a partir de imágenes satelitales. A partir de dichas imágenes, los autores proponen un algoritmo basado en la comparación de los bordes detectados en cada imagen para recuperar la altura y características del terreno. Esta información de altura y características se codifica en un modelo digital del terreno que es equivalente a un mapa

de curvas. Los resultados son modelos de terreno digitales que mantienen un error de menos de 100 metros respecto a modelos de terreno digitales de referencia.

3.2.2. Minería de datos

Esta clase de métodos utiliza grandes volúmenes de datos para generar los mapas de interés, como lo demuestran los trabajos de Lima [F. 09]. En cuanto a los datos que se utilizan, la información de localización proporcionada por el GPS es la elección más popular.

El procesamiento de datos se hace habitualmente en forma de consultas SQL, como lo propone Lima [F. 09] en su trabajo. Aunque se ha dicho que las consultas SQL son práctica común de esta categoría, [SMSL09] muestra una alternativa partiendo también de cantidades masivas de muestras. Los métodos propuestos por estos autores se describen a continuación:

“Minería datos espaciales de puntos GPS para la extracción automática de la red de carreteras” [F. 09]. Utilizando mas de 30 millones de puntos GPS y consultas SQL, se logra determinar la configuración de carreteras de una ciudad de Portugal, llamada Arganil. De este gran volumen de puntos de ubicación, se seleccionan solamente los que hayan sido adquiridos de cinco o más satélites. Posteriormente, los puntos se dibujan en una rejilla de celdas que representan la región que será mapeada. Cada una estas rejillas contiene información sobre la probabilidad de formar parte de una carretera, según la cantidad de *fixes* GPS que contengan. Teniendo esta representación, se calculan los centroides de los posibles caminos recorridos por los receptores GPS usando un criterio de concentración de puntos GPS en las celdas. Estos centroides son utilizados después para obtener un mapa de conexión topológica, que reflejará la conectividad de las calles. En cuanto a los resultados, los autores reportan una correspondencia del 77% entre el mapa generado y los mapas previamente generados de la ciudad, así como la correcta clasificación de la dirección de la carretera en el 96% de los casos. Cabe aclarar que

una parte del porcentaje de no-correspondencia se debe a que su método fué capaz de encontrar nuevos caminos que no estaban registrados en los mapas de la ciudad.

De una forma muy similar, Wenhuan, en su artículo titulado “Generación automática de la red de carreteras desde trayectorias GPS masivas de vehiculos” [SMSL09] , a partir de una gran cantidad de muestras tomadas en la ciudad de Jilin, China, construye un mapa de las calles. Para lograrlo, se extraen las trayectorias de las muestras para generar un mapa de bits que es procesado con técnicas como dilatación y apertura morfológica, además de un algoritmo propuesto por los autores llamado *combustión*. Es importante destacar que una diferencia clave entre estos trabajos es que Lima y colaboradores construyen los mapas a partir de consultas SQL, mientras que Wenhuan realiza la minería de datos solamente en una fase inicial, y toma un enfoque de procesamiento de imágenes después de adquirir un mapa de bits inicial de las muestras del GPS. En cuanto a los resultados obtenidos, solo se presenta una imagen del mapa generado por el método que se compara con un mapa de la ciudad previamente generado donde puede notarse que coinciden.

3.2.3. Análisis de vectores de movimiento

De forma similar a la minería de datos, los trabajos considerados bajo esta categoría utilizan muestras de puntos GPS para construir el mapa por donde pasan los caminos. Sin embargo, una característica distintiva de estos métodos, es que los puntos GPS son considerados como vectores que son tratados con técnicas de aprendizaje automático, como el clustering de [ANN10] que se describe a continuación:

“Inferencia de mapas de caminos robusta y precisa” [ANN10]: este método guarda una relación de similitud con el expuesto por Wenhuan, pero mantiene una diferencia clave en el criterio para determinar las líneas centrales de los caminos. Mientras Wenhuan [SMSL09] trabaja sobre bitmaps, Agamennoni y colaboradores utilizan agrupamientos con criterios de distancia y ángulo entre trayectorias. En este trabajo se utilizan radios GPS dedicados que se montan en los camiones de carga que circulan en una mina. El

resultado es un mapa que describe las trayectorias reales de los camiones, donde es posible distinguir el sentido de los caminos donde transitan y los cruces de los mismos. Dicho resultado se evalúa mediante la comparación con dos métodos de inferencia de caminos. El primer método [DaBH06] pertenece a la categoría de procesamiento de imágenes, donde se observa que la inferencia de mapas analizando vectores ofrece información adicional sobre el mapa, como el sentido de la calle. El segundo método [?], también utiliza agrupamientos para trazar las carreteras, aunque muestra una menor precisión para localizar los puntos de intersección de caminos.

Hasta este punto se han revisado algunos de los métodos para generar mapas a partir de puntos GPS e imágenes satelitales, de los cuales se puede resumir lo siguiente:

1. Buenos resultados: los métodos reportados son capaces de generar mapas con un buen grado de precisión, además de aportar un poco de información contextual, como el sentido de las calles y algunas reglas de tránsito.
2. Fuentes de datos homogéneas: Aunque ha sido señalado por otros autores [LMD⁺07] dentro del dominio del posicionamiento en interiores, ninguno de los métodos revisados combina información de distintos sensores. Es importante considerar los beneficios que este acercamiento puede aportar a la generación de mapas en exteriores.
3. Volumen de información requerida: en cuanto a los métodos que utilizan minería de datos, es notable la necesidad de un gran volumen de datos de distintos participantes para lograr un buen rendimiento de la aplicación.
4. Infraestructura necesaria: si bien es cierto que muchos de los métodos propuestos usan equipo de uso común como son los receptores GPS, algunos requieren acceso a datos ya recolectados o el uso de hardware dedicado, como también pueden ser receptores GPS montados en unidades especialmente designadas para la recolección de muestras.

3.3. Extracción de información contextual en caminos utilizando acelerometría.

En cuanto a la extracción de información contextual del terreno, hay aún muy poco trabajo relacionado, siendo la investigación más cercana la referente al monitoreo colaborativo de tráfico y condiciones del camino [MPR08] y el reconocimiento de baches [EGH⁺08]. Actualmente, el contexto mas explorado por la acelerometría es el de la actividad física humana, como el andar, por su aplicación al campo de la medicina y la asistencia personal. [MCS⁺07].

Dividimos el trabajo relacionado en dos categorías, que se explican a continuación:

3.3.1. Frameworks o marcos de trabajo para la clasificación de movimientos utilizando acelerometría.

En el artículo titulado “Clasificación de movimientos básicos diarios usando un acelerómetro triaxial”, los autores M.J. Mathie *et al.* [Cos00] proponen un marco de trabajo construido alrededor de un árbol binario, cuyo propósito es organizar de manera jerárquica los movimientos humanos para facilitar su clasificación. Con esto, los autores diseñan un clasificador de movimientos simples que trabaja con las señales de un acelerómetro montado en la cintura de algunos voluntarios. Cabe señalar que dentro de su ejemplo, construido para probar su marco de trabajo, utilizan un clasificador publicado anteriormente por los mismos autores y que funciona detectando periodos de actividad e inactividad en la señal del acelerómetro. La importancia de este trabajo, es que sugiere un método para diseñar un clasificador para movimientos registrados por un acelerómetro, que podría usarse no solo para movimientos humanos. El funcionamiento del clasificador se demuestra utilizando un grupo de 26 personas ejecutando un conjunto de actividades básicas, logrando clasificar correctamente el 97.7% de los casos.

3.3.2. Reconocimiento de actividades usando acelerometría.

En el artículo “Reconocimiento de actividades desde datos del acelerómetro” [RDML], Nishkam Ravi *et al.* reportan sus resultados al crear un reconocedor simple de actividades cotidianas utilizando un acelerómetro triaxial. Estas actividades cotidianas son correr, caminar, lavarse los dientes y aspirar. Se usan técnicas habituales de pre-procesamiento de datos como filtrado de bandas, y se usan clasificadores conocidos como *SVM*, agrupamiento de vecinos cercanos y *Naive Bayes* obteniendo resultados satisfactorios, con porcentajes de clasificación que llegan al 99.57% de precisión. Es importante mencionar que estos clasificadores fueron utilizados en el marco de los metaclassificadores, como voto y *bagging/boosting*.

En el artículo “Prediciendo el modo de transporte con datos de aceleración de un iPhone” [NSY09], reportan la correcta clasificación de algunas actividades utilizando un acelerómetro triaxial de un iPhone 3G. Las actividades clasificadas fueron el medio de transporte de una persona (caminar, correr, andar en bicicleta y manejar). Utilizando SVM, la precisión que logran en la clasificación es del 93.88%.

En el artículo “Reconocimiento de eventos de manejo usando modelos ocultos de Markov”, [Mit99] D. Mitrovik plantea una manera de reconocer eventos al manejar como son frenados o vueltas usando modelos ocultos de Markov. Cabe señalar, que en su trabajo no se utilizan acelerómetros, mas bien se infiere la aceleración de la misma información del GPS antes de ser presentada al modelo concebido por el autor. Los resultados reportados muestran una precisión mayor al 80% para los eventos tratados en el trabajo. Este tipo de métodos demuestran cómo es posible extraer información contextual secundaria con un solo sensor.

En el artículo “Clasificación de actividades de movimiento a través de DDTW” en datos de acelerómetro [MCS⁺07], el método presentado por los autores realiza una clasificación similar a la descrita por Ravi y Dankedar, con la diferencia de que se usa *derivative dynamic time warping*. DDTW es una técnica que permite trasladar una señal en el tiempo para ajustarla a otra señal similar, y en el trabajo mencionado se usa

con el propósito de lograr una correcta clasificación de actividades como caminar y subir o bajar escaleras aún con variaciones de velocidad. Para reportar los resultados, los autores comparan su método con otros métodos similares como *linear time warping* o *dynamic time warping* y clasifican los movimientos de caminar, subir escaleras y bajar escaleras. Adicionalmente, en cada movimiento distinguen dos componentes de movimiento: radial y longitudinal. En cuanto a los movimientos radiales, su método no da resultados significativamente superiores a los métodos contra los que se comparan. Por otra parte, en los movimientos longitudinales logran una clasificación perfecta (100% de las señales reconocidas).

En el artículo “TrafficSense: monitoreo enriquecido de las carreteras y sus condiciones utilizando teléfonos inteligentes”[MPR08], desarrollado por P. Mohan *et al.* de los laboratorios de Microsoft Research, en este trabajo se consideran dos problemas: Por una parte, monitorear el tráfico, y por otra, detectar algunas condiciones de los caminos como baches y topes. De estos dos problemas, es de especial interés la detección de las características de los caminos, misma que resuelven utilizando heurísticas construidas a partir de observaciones de las señales sensadas, como variaciones repentinas en la señal al pasar por un bache. Los resultados que muestra este trabajo en cuanto a la detección de baches demuestran que la extracción de características del terreno usando acelerometría es un problema difícil, puesto que fué necesario crear dos heurísticas distintas, una para el caso de detección en situaciones de baja velocidad y otra para alta velocidad para mantener los falsos positivos por debajo del 10%. Este reporte técnico también propone un modelo de sensado colaborativo y oportunista.

La patrulla de los baches, o *The Pothole Patrol*, es un sistema de reconocimiento de anomalías severas en las carreteras, que emplea datos de acelerómetros dedicados y receptores GPS montados en siete taxis del área de Boston [EGH⁺08]. Aplicando filtros a la señal y utilizando agrupamiento según la localización de las incidencias, logran detectar y ubicar baches, mismos que son después reportados a las autoridades correspondientes. En este trabajo encontramos indicios del uso combinado de sensores

(acelerómetro y GPS). Es importante recalcar que a diferencia de la patrulla de los baches, este trabajo utiliza sensores de propósito general, además de que la interacción entre los sensores GPS y de aceleración tiene el objetivo de agrupar las muestras (según la posición reportada por el GPS) para luego efectuar el reconocimiento de eventos. El trabajo reporta una precisión del 92.4% en el reconocimiento de los baches, mismos resultados que son validados con el reporte de las calles afectadas por baches en los recorridos de los taxis.

Haciendo un breve resumen del trabajo relacionado con la tipificación de terrenos con acelerometría hasta la fecha, podemos decir que:

1. Se tienen buenos resultados. Los resultados son satisfactorios y alentadores para la búsqueda de nuevas aplicaciones de extracción de información contextual con acelerómetros.
2. Uso mayoritario de hardware dedicado. Aunque se expone una excepción (con el uso del iPhone), la mayor parte de los métodos propuestos requiere el uso de hardware dedicado, y colocado en lugares estratégicos como la cintura y las muñecas, lo que resulta demasiado invasivo para aplicaciones de uso cotidiano.

De cualquier forma, es importante recalcar la poca correspondencia entre los objetivos de los trabajos mencionados con el tema desarrollado en esta tesis, aunque no dejan de ser ilustrativos y un buen punto de partida para la investigación efectuada.

Finalmente, cabe hacer notar que aunque existe mucha actividad de investigación en cuanto a la generación automática de mapas y la explotación de la información contextual en ámbitos separados, y con resultados satisfactorios, encontramos la oportunidad de fusionar información de distintos sensores para mejorar la generación automática de mapas en términos del volumen de datos necesarios e infraestructura para su funcionamiento. De la misma manera (combinando la información de distintos sensores y explotando el contexto del usuario) podemos enriquecer la información presentada a un sistema GIS o de transporte inteligente.

Capítulo 4

Generación de mapas a partir de información de localización y acelerometría.

4.1. Introducción.

Durante este capítulo, se expondrá el mecanismo propuesto para generar mapas georeferenciados utilizando información de localización (datos GPS) y acelerometría. En la primera sección se habla del modelo de sistema que sustenta la operación del mecanismo, a continuación, se hace un análisis de la información que se puede obtener de los sensores por separado. Después se explica el mecanismo de detección de inconsistencia en trazos GPS y su corrección, finalizando con un ejemplo y una sección de resultados experimentales.

4.2. Modelo del sistema.

El modelo del sistema está representado por la figura 4.1, donde se puede apreciar las fases de recolección de datos, preprocesamiento, validación, corrección y trazado del mapa.

Para la aplicación del método descrito en esta sección y que propone esta investigación, se consideran las siguientes restricciones en el contexto donde se realiza el sensado.

1. Velocidad constante

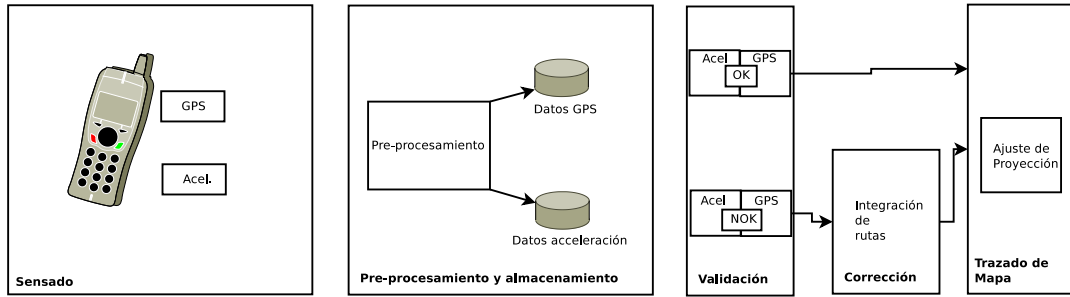


Figura 4.1: Modelo del sistema para la generación de mapas a partir de información de localización y acelerometría.

2. No se permite el movimiento en reversa.
3. Cambios graduales en la dirección de movimiento, con diferencias no mayores a 90 grados entre puntos consecutivos.
4. El vehículo tiende a moverse en mayor proporción “hacia adelante” respecto a cuanto se mueve “hacia los lados”. Este comportamiento es congruente con el movimiento que describe un automóvil al desplazarse en una calle.
5. Los ejes del acelerómetro están fijos durante el trayecto y alineados con el automóvil: el eje Z del acelerómetro coincide con el eje definido entre la parte frontal y posterior del automóvil, mientras que el eje X coincide con el eje definido entre los lados del vehículo. El eje Y del acelerómetro está comprendido entre el piso y el techo del automóvil. Esta alineación de ejes se ilustra en la figura 4.2 y la figura 4.3.

El sensado se hace en un recorrido, y el procesamiento de los datos se hace fuera de línea. A fin de diseñar un sistema de generación de mapas robusto, se realizó un análisis de la naturaleza y calidad de la información que pueden proporcionar los sensores con los que se propone trabajar en esta investigación. A continuación se muestra dicho análisis.

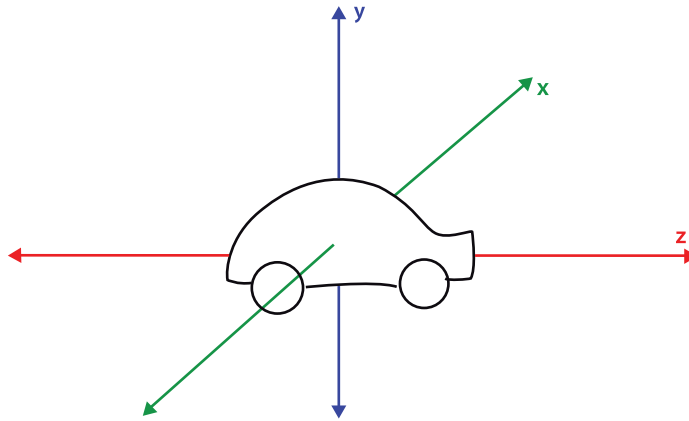


Figura 4.2: Alineación de los ejes del automóvil.

4.3. Análisis de la información proporcionada por los sensores

A continuación se describe y analiza la información contextual que se puede inferir a partir de datos obtenidos del aGPS y del acelerómetro. Se hará especial énfasis en la susceptibilidad al error por parte del sensor de posicionamiento y se señalará cómo es posible mejorar el desempeño del trazado de mapas usando datos del acelerómetro.

4.3.1. Análisis de la información proporcionada por el aGPS

El sistema GPS de aplicación civil utilizado por el sistema propuesto en esta investigación tiene una precisión de entre 10 y 20 metros [Axe05]. Desafortunadamente, esta precisión es difícil de conseguir porque requiere de algunas condiciones como la disponibilidad de al menos 4 satélites, red de datos y una línea de visión al cielo. Adicionalmente, a mayor velocidad, es de esperar la degradación de la precisión de los puntos reportados por el receptor.

Con la finalidad de hacer un análisis de la información que puede aportar un receptor aGPS en la tarea del trazado de calles de ciudades, utilizamos la aplicación *MyTracks* [?], hecha para dispositivos inteligentes con sistema operativo Android. La aplicación tiene la capacidad de grabar recorridos utilizando muestras de puntos GPS. Esta aplicación hace

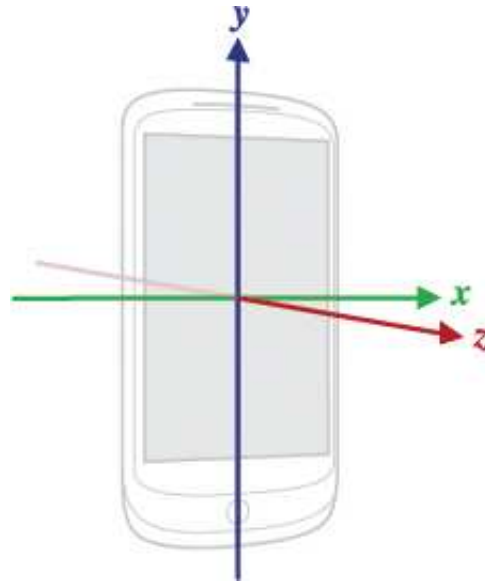


Figura 4.3: Alineación de los ejes del acelerómetro, según la documentación de Android.

disponible una API que nos permite revisar la información generada por el programa, y hacer uso de ella en nuestras aplicaciones.

La información que se puede obtener directamente del sistema GPS es:

1. Ubicación, representada en latitud, longitud y elevación respecto al nivel del mar.
2. Tiempo y fecha en formato Día-mes-año, hora:minuto:segundo.
3. Velocidad, en metros por segundo.
4. Rumbo, definido en grados desde el norte geográfico hacia el este.

Como dato adicional, el sistema *myTracks* reporta un estimado de la precisión en metros de la muestra. Para el caso del receptor GPS del teléfono utilizado, se reportan precisiones de hasta 4 metros.

Como se puede observar en la figura 4.4 y 4.5, y como se ha señalado anteriormente, el sistema *myTracks* se desempeña bien en terrenos abiertos, pero su propensión al error es mucho mayor cuando el sensado se hace en calles de ciudad. Para poder obtener un trazado confiable en la ciudad, es necesario poder distinguir los puntos que son resultado de un sensado erróneo y corregirlos.



Figura 4.4: Ejemplo de ruta trazada por *MyTracks* en terreno abierto.

Para los fines de esta investigación, es necesario obtener información del movimiento relativo del automóvil. Esto es, debemos ser capaces de determinar el cambio de dirección entre punto y punto reportado por el sistema GPS. Para lograrlo, se consideran los pares de puntos GPS y se determina el ángulo entre ellos, y después se compara el siguiente ángulo, en el siguiente par de muestras GPS. La diferencia entre estos ángulos determinará el cambio en la dirección durante el trayecto, y puede utilizarse después para verificar la consistencia de las mediciones usando información de otro sensor, como el acelerómetro. El razonamiento detrás de este proceso de determinación de cambio de dirección se ilustra en la figura 4.6. Según las restricciones del sistema, cuando se detecta un punto con un cambio de dirección mayor a 90 grados, el punto es marcado como inválido, puesto que este cambio de dirección implica un cambio repentino de movimiento. Todo este proceso mencionado se expone en los algoritmos 1, que detalla



Figura 4.5: Ruta trazada por *MyTracks* en zona urbana.

cómo se determina el cambio de dirección entre puntos GPS y el algoritmo 2 que es auxiliar del algoritmo 1.

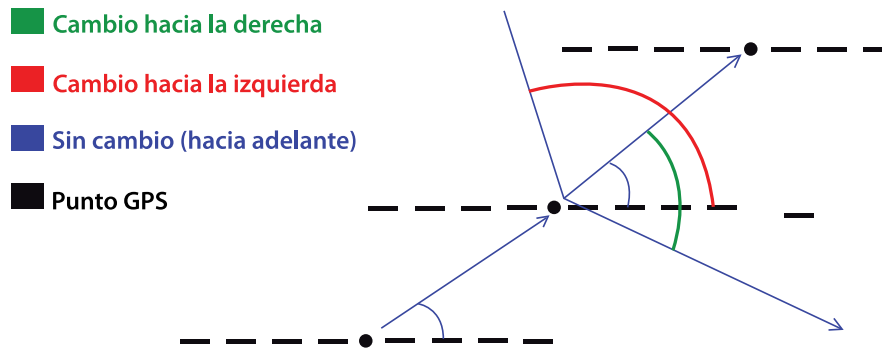


Figura 4.6: Detección de cambio de dirección del automóvil con muestras GPS.

En el algoritmo 1, la variable *puntosGPS* es una estructura que contiene información sobre la lista de *fixes* GPS que se está evaluando. Esta información es *numElementos*, que contiene el número de *fixes* del recorrido, *sistemaCoordenadas*, que contiene información sobre la proyección geográfica considerada por los puntos GPS, *posX* y *posY* que contienen las coordenadas del punto según la proyección geográfica en uso, \angle que

Algorithm 1 Detección de cambio de dirección entre puntos GPS

Require: *puntosGPS*
Require: *puntosGPS.sistemaCoordenadas = Mercatoruniversaltransversal*
Require: *puntosGPS.numElementos > 1*
 $\angle \leftarrow NULL$
for $i = 1$ **to** $i < puntosGPS.numElementos$ **do**
 $\Delta X' \leftarrow puntosGPS[i + 1].posX - puntosGPS[i].posX$
 $\Delta Y' \leftarrow puntosGPS[i + 1].posY - puntosGPS[i].posY$
 $\angle' \leftarrow tangenteInversa\left(\frac{\Delta Y'}{\Delta X'}\right)$
if $\angle == NULL$ **then**
 $puntosGPS[i].cambioDireccion \leftarrow desconocido$
else
 $puntosGPS[i].cambioDireccion \leftarrow obtenerNuevoRumbo(\Delta X', \Delta X, \Delta Y, \Delta Y', \angle, \angle')$
end if
 $\Delta X \leftarrow \Delta X'$
 $\Delta Y \leftarrow \Delta Y'$
 $\angle \leftarrow \angle'$
end for

almacena temporalmente el valor de los ángulos entre puntos GPS. Finalmente, tenemos *cambioDireccion*, que es la variable donde se guarda la codificación del cambio de dirección.

En el algoritmo 2, se utilizan las variables de entrada ΔX y ΔY que contienen las coordenadas del punto GPS que se está evaluando, de la misma forma que las variables $\Delta X'$ y $\Delta Y'$ que contienen las coordenadas del siguiente punto GPS. De manera similar, se requiere el par de ángulos \angle y \angle' que corresponden al ángulo entre el par de puntos anteriores y al par de puntos actual. La variable *tolerancia* es la cantidad de grados que podemos permitir cambiar entre mediciones antes de considerar un cambio de dirección. Esta variable puede ajustarse según la precisión de las mediciones. Los valores de retorno del algoritmo 2 están codificados como se muestra en la figura 4.7. Este algoritmo es llamado por el algoritmo 1 cada vez que se obtiene un par de puntos GPS entre los cuales se desea conocer el cambio de dirección.

Algorithm 2 Obtención del nuevo rumbo entre dos puntos GPS

Require: $\Delta X, \Delta X', \Delta Y, \Delta Y', \angle, \angle'$ **Require:** $|\Delta X| > 0, |\Delta X'| > 0, |\Delta Y| > 0, |\Delta Y'| > 0, |\angle| > 0, |\angle'| > 0$ *tolerancia* \leftarrow 20**if** $\Delta X.signo \neq \Delta X'.signo$ **and** $\Delta Y.signo \neq \Delta Y'.signo$ **then***return* -1 -1**end if***absDif* = $|\Delta X' - \Delta X|$ **if** *absDif* > 90 **then***return* -1**end if****if** *absDif* < *tolerancia* **then***return* 2**end if****if** $\Delta X' > \Delta X$ **then****if** *absDif* < 60 **then***return* 3**else***return* 4**end if****end if****if** $\Delta X' < \Delta X$ **then****if** *absDif* < 60 **then***return* 1**else***return* 0**end if****end if**

4.3.2. Análisis de la información proporcionada por el acelerómetro

El acelerómetro no tiene manera de ubicarse dentro de un marco absoluto de referencia. Para enfrentar esta situación, se le puede situar en un marco de referencia relativo (adelante, izquierda, derecha) donde se relaciona con el movimiento de un automóvil. Utilizando este sistema de referencia, podemos detectar el movimiento relativo del automóvil de manera confiable.

Aunque los sensores de los teléfonos inteligentes son en su mayor parte equipo de propósito general y sus mediciones tienden a ser ruidosas, la precisión que tienen es suficiente para detectar la dirección de movimiento del teléfono. En el caso del sensor utilizado en este trabajo, se determinó de manera empírica que su error está entre el 4 % y el 5 % al compararse sus medidas con una cantidad conocida.

El procedimiento para determinar la dirección de movimiento del móvil consiste en medir las fuerzas que afectaron al dispositivo durante su trayecto entre dos puntos y encontrar el ángulo entre las componentes longitudinal y transversal del movimiento. Para superar la limitación de la precisión del acelerómetro, se dividió la región de posible movimiento en 4 secciones, como se ilustra en la figura 4.7.

El ángulo resultante entre las componentes de movimiento puede asociarse con alguna de las categorías establecidas (0,1,2,3,4), y determinará el cambio de dirección *real* del automóvil. El algoritmo para determinar la dirección del móvil usando información del acelerómetro se presenta en el algoritmo 3, que muestra el procedimiento general para determinar el cambio de dirección con el acelerómetro y 4 que es auxiliar del algoritmo 3.

El algoritmo 3 utiliza las variables *muestrasAcelerometro*, que es una lista de todas las mediciones tomadas por el acelerómetro durante el trayecto. A su vez, contiene *numElementos*, que es el número total de mediciones tomadas, *time* que es la marca de tiempo en que la muestra fue tomada y *direccionAccel*, donde se guarda la dirección del automóvil ya codificada. También se utiliza *puntosGPS*, que es una lista de los puntos GPS registrados en el recorrido y también contiene los campos *tiempo*, que actúa como

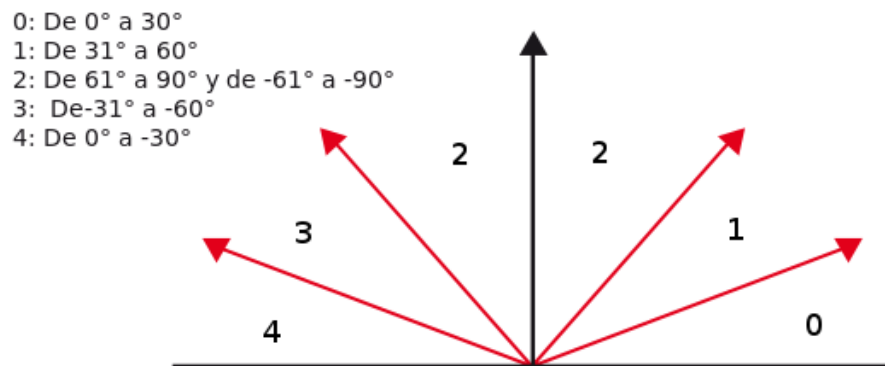


Figura 4.7: División del rango de movimiento en 4 secciones.

marca de tiempo de cada punto GPS y *numElementos* que es la cantidad de puntos GPS en el recorrido. Durante la ejecución del algoritmo también se utilizan las variables *mediaFuerzasX* y *mediaFuerzasY* para almacenar el promedio de las fuerzas en los ejes *x* e *y* entre dos puntos GPS y obtener el ángulo, que es pasado al algoritmo auxiliar 4.

De manera similar a lo que se hace con la dirección del GPS, la dirección obtenida del ángulo llamado *angle* obtenido por el acelerómetro en el algoritmo 4 codifica la dirección según la figura 4.7. Aunque la dirección de movimiento no pueda establecerse con una granularidad al nivel de grados, el complementar esta información con la información proporcionada es suficiente para detectar inconsistencias en los puntos del GPS, como se muestra en la siguiente sección.

Algorithm 3 Detección de cambio de dirección con un acelerómetro

Require: *muestrasAcelerometro*, *puntosGPS0*

Require: *muestrasAcelerometro.numElementos* > 0

Require: *puntosGPS.numElementos* > 1

Require: \forall *muestraAceleracion* \in *muestrasAcelerometro*,

Require: *puntoGPS[1].time* < *muestraAceleracion.time*

Require: *puntoGPS[puntosGPS.numElementos].time* > *muestraAceleracion.time*

inicio \leftarrow 1

fin \leftarrow 1

for *i* = 1 **to** *i* < *puntosGPS.numElementos* **do**

fin \leftarrow *ultimoPuntoEnRango(muestrasAcelerometro, puntosGPS)*

mediaFuerzasX \leftarrow *mediaDeRango(muestrasAcelerometro[inicio].x*
, muestrasAcelerometro[fin].x)

mediaFuerzasY \leftarrow *mediaDeRango(muestrasAcelerometro[inicio].y*
, muestrasAcelerometro[fin].y)

\angle \leftarrow *tangenteInversa*($\frac{\textit{mediaFuerzasY}}{\textit{mediaFuerzasX}}$)

puntosGPS[i].direccionAccel \leftarrow *obtenerRumbo*(\angle)

end for

Algorithm 4 Obtención del rumbo a partir del ángulo

Require: *angle*

if *angle* > 0 **and** *angle* <= 30 **then**

return 0

else

if *angle* > 30 **and** *angle* <= 60 **then**

return 1

else

if *angle* > 60 **and** *angle* <= 90 **or** *angle* > -90 **and** *angle* <= -60 **then**

return 2

else

if *angle* > -60 **and** *angle* <= -30 **then**

return 3

else

if *angle* > -30 **and** *angle* <= 0 **then**

return 4

end if

end if

end if

end if

end if

4.4. Detección de inconsistencia de trazos aGPS

Utilizando los algoritmos 1, 2, 3 y 4 es posible establecer una métrica común para el cambio de dirección reportada por el GPS y el acelerómetro. Esto se logra gracias a la codificación de dirección en cinco categorías (0,1,2,3,4)

Una vez que se ha establecido una métrica común entre el acelerómetro y el dispositivo aGPS, es posible validar los trazos de los recorridos del GPS y determinar si existe una inconsistencia entre la información reportada por el GPS y el acelerómetro. La estrategia utilizada para hacer la validación está relacionada con las técnicas de integración de rutas, usando información del desplazamiento que provee el acelerómetro.

Para detectar la inconsistencia entre la información del aGPS y el acelerómetro, se comparan directamente los cambios de dirección reportados por el aGPS y por el acelerómetro. Cuando la diferencia entre los ángulos ya ubicados en su sección es mayor a uno, se considera que el punto no es consistente, y se marca para su corrección. Esto se logra utilizando el algoritmo 5, donde las variables principales son *puntoGPS*, que representa cada punto GPS, y su campo *consistente* recibe el valor de *true* o *false*, que es revisado después para determinar si es necesario corregir el punto.

4.4.1. Algoritmo de detección de inconsistencia

Algorithm 5 Detección de inconsistencia en puntos GPS

Require: *puntosGPS.numElementos* > 0

for all *puntosGPS* in *puntosGPS* **do**

if $|puntoGPS.direccionAccel - puntoGPS.cambioDireccion| \leq 1$ **then**

puntoGPS.consistente \leftarrow *true*

else

puntoGPS.consistente \leftarrow *false*

end if

end for

Estos puntos inconsistentes pueden ser modificados para mejorar la precisión del trazo, como se muestra en la siguiente sección.

4.5. Corrección de trazos aGPS con acelerometría.

Se sabe que de un sensor inercial como un acelerómetro se puede obtener información de posición [And], así que una vez que se ha determinado que una medición del sistema GPS no es válida, es posible corregirla utilizando información del acelerómetro. En el caso de este trabajo, la información del acelerómetro se utiliza para determinar el tipo de trazo que se debe corregir (una recta o una curva) y para modificar la posición del trazo en el mapa.

4.5.1. Algoritmo de corrección de trazos en rectas

Utilizando el método de obtención de dirección expuesto con anterioridad, es posible distinguir trayectos en línea recta. Una vez determinada una recta, la corrección de los puntos GPS que no son válidos consiste en la interpolación de sus coordenadas con puntos que son válidos. El procedimiento se detalla en el algoritmo 6

Algorithm 6 Corrección de puntos GPS en rectas

Require: *segmentoPuntosGPS*

Ensure: *segmentoPuntosGPS.numElementos* > 2

Ensure: *segmentoPuntosGPS* es una recta

for all *puntosGPS* **in** *puntosGPS* **do**

if *puntoGPS.consistente* \leftarrow *false* **then**

puntoConfiableA \leftarrow *puntoConfiableAntes(puntoGPS)*

puntoConfiableB \leftarrow *puntoConfiableDespues(puntoGPS)*

puntoGPS \leftarrow *interpolaPuntos(puntoConfiableA, puntoConfiableB)*

end if

end for

En el algoritmo 6, se utiliza la variable *segmentoPuntosGPS*, que es un subconjunto de *puntosGPS*. Se considera que *segmentoPuntosGPS* contiene solamente puntos que han sido considerados como pertenecientes a una línea recta por el acelerómetro. Esto se puede hacer separando de la lista de puntos GPS aquellos puntos consecutivos donde su dirección, reportada por el acelerómetro, es 2.

Es importante señalar que la corrección se hace cuando existen puntos GPS confiables antes y después del punto GPS inválido.

4.5.2. Algoritmo de corrección de trazos en curvas

De manera similar al caso de las rectas, el acelerómetro puede detectar fácilmente el cambio de dirección que caracteriza a una curva. Para la corrección de los puntos GPS que no son consistentes con el cambio de dirección de la curva, se aplica un corrimiento de la dirección en favor de la dirección indicada por el acelerómetro. Los algoritmos 7 y 8 resumen el procedimiento.

Algorithm 7 Corrección de puntos GPS en curvas

Require: *segmentoPuntosGPS*

Ensure: *segmentoPuntosGPS.numElementos* > 2

Ensure: *segmentoPuntosGPS* es curva

for all *puntosGPS* in *puntosGPS* **do**

if *puntoGPS.consistente* \leftarrow *false* **then**

puntoConfiable \leftarrow *puntoConfiableAntes(puntoGPS)*

puntoGPS \leftarrow *muevePuntoGPS(puntoConfiable, PuntoGPS)*

end if

end for

De manera similar al algoritmo 6, el algoritmo 7 separa en *segmentoPuntosGPS* un subconjunto de *PuntosGPS*, pero considera únicamente los puntos consecutivos que pertenecen a una curva. Estos puntos pueden elegirse como todos los puntos reportados por el acelerómetro con un dirección 0,1,3 o 4. El algoritmo 7 revisará el estado de consistencia del punto GPS y utilizará a los algoritmos 9 y 10 para hacer la corrección, en caso de ser necesaria.

4.6. Ejemplo de funcionamiento del algoritmo

En esta sección se demuestra el funcionamiento del algoritmo según el modelo del sistema utilizando una lista de puntos reales. Se ejemplifica el caso de corregir una recta y una curva.

Algorithm 8 Movimiento de un punto GPS para corregir una curva

Require: *segmentoPuntosGPS*

Ensure: *segmentoPuntosGPS.numElementos* > 2

Ensure: *segmentoPuntosGPS* es una curva

```

for all puntoGPS in puntosGPS do
  cuadrante  $\leftarrow$  obtenerCuadrante(puntoGPS)
  cambio  $\leftarrow$  10
  if puntoGPS.cambioDireccion < puntoGPS.direccionAccel then
    nuevoAngulo  $\leftarrow$  puntoGPS.angulo + cambio
  else
    nuevoAngulo  $\leftarrow$  puntoGPS.angulo - cambio
  end if
  componentes = nuevasComponentes(puntoGPS, nuevoAngulo, cuadrante)
  puntoGPS.X  $\leftarrow$  puntoGPS.X + componentes.X
  puntoGPS.Y  $\leftarrow$  puntoGPS.Y + componentes.Y
end for

```

Corrección de rectas

En la figura 4.8 se muestran los puntos GPS registrados en una calle de la localidad de Cholula, en Puebla. Por brevedad, se muestra solamente el procedimiento para validar y corregir el punto número 6 del recorrido, que corresponde a la mitad del trayecto.



Figura 4.8: Trazo de recta a corregir dibujado en *Google Earth*. El ejemplo se centra en la mitad del recorrido.

Algorithm 9 Cálculo de las nuevas componentes de movimiento del GPS**Require:** $puntoGPS_A, puntoGPS_B$ **Require:** $nuevoAngulo, anteriorAngulo$ **Require:** $cuadranteAnterior$ $dx = puntoGPS_B.X - puntoGPS_A.X$ $dy = puntoGPS_B.Y - puntoGPS_A.Y$ **if** $nuevoAngulo > 0$ **then** **if** $dy > 0$ **then** $q = 1$ **else** $q = 3$ **end if****else** **if** $dy > 0$ **then** $q = 2$ **else** $q = 4$ **end if****end if****if** $qAaccl < qAgps$ **then** **if** $(q == 1 || q == 3)$ **then** $step = step * -1$ **end if****else** **if** $(q == 2 || q == 4)$ **then** $step = step * -1$ **end if****end if****if** $cuadranteAnterior == 1$ **then** **if** $nuevoAngulo < 0$ **then** **if** $nuevoangulo < 0$ **and** $nuevoangulo \geq -90$ **then** $xdir \leftarrow 1$ $ydir \leftarrow -1$ $nq \leftarrow 4$ **else** $xdir \leftarrow -1$ $ydir \leftarrow -1$ $nq \leftarrow 3$ **end if** **else** **if** $nuevoangulo > 0$ **and** $nuevoangulo \leq 90$ **then** $xdir \leftarrow 1$ $ydir \leftarrow 1$ $nq \leftarrow 1$ **else** $xdir \leftarrow -1$ $ydir \leftarrow 1$ $nq \leftarrow 2$ **end if** **end if****end if**

Algorithm 10 Cálculo de las nuevas componentes de movimiento del GPS (cont)

```

if cuadranteAnterior == 3 then
  if nuevoAngulo < 0 then
    if nuevoangulo < 0 and nuevoangulo ≥ -90 then
      xdir ← -1
      ydir ← 1
      nq ← 2
    else
      xdir ← 1
      ydir ← 1
      nq ← 1
    end if
  else
    if nuevoangulo > 0 and nuevoangulo ≤ 90 then
      xdir ← -1
      ydir ← -1
      nq ← 3
    else
      xdir ← 1
      ydir ← -1
      nq ← 4
    end if
  end if
end if
if cuadranteAnterior == 4 then
  if nuevoAngulo > 0 then
    if nuevoangulo > 0 and nuevoangulo ≤ 90 then
      xdir ← 1
      ydir ← 1
      nq ← 1
    else
      xdir ← -1
      ydir ← 1
      nq ← 2
    end if
  else
    if nuevoangulo < 0 and nuevoangulo ≥ -90 then
      xdir ← 1
      ydir ← -1
      nq ← 4
    else
      xdir ← -1
      ydir ← -1
      nq ← 3
    end if
  end if
end if
xcomp = |puntoGPS.Resultante * cosd(nuevoangulo)| * xdir
ycomp = |puntoGPS.Resultante * sind(nuevoangulo)| * ydir

```

Determinando la dirección usando el acelerómetro y el GPS

Para facilitar el seguimiento del ejemplo, la recta ilustrada en la figura 4.8 se grafica utilizando sus coordenadas UTM y se señala el punto a corregir en la figura 4.9 con color rojo.

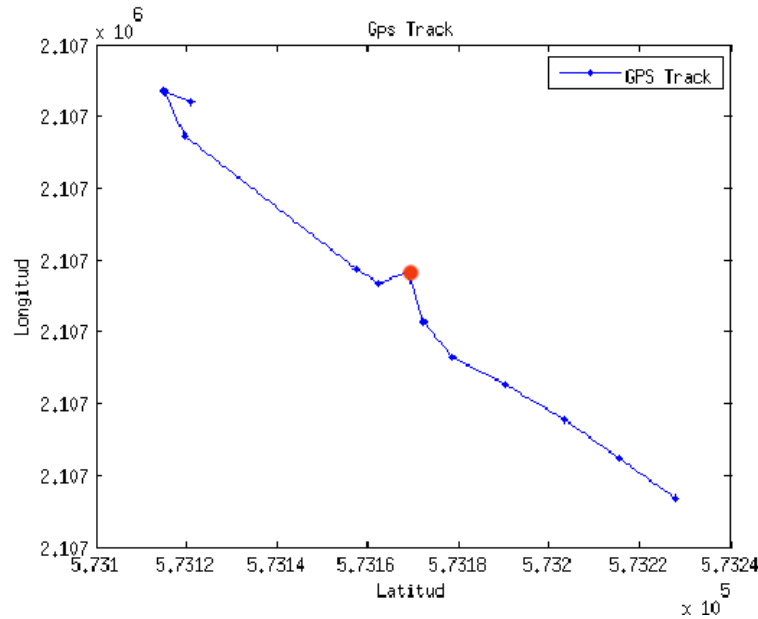


Figura 4.9: Trazo de recta a corregir el punto objetivo es el sexto en la secuencia, marcado en rojo, correspondiente a la figura 4.8.

Para calcular la dirección con los puntos del GPS, seguimos el algoritmo 1. Al llegar a la iteración $i = 6$ se obtienen las diferencias de distancia en las coordenadas X e Y , llamadas $\Delta X'$ y $\Delta Y'$. Estas diferencias se calculan respecto al siguiente punto $i = 7$. Estas componentes se utilizan para determinar el ángulo entre puntos GPS desde la horizontal, definida por el eje X . Cuando se tienen los ángulos $\Delta X'$ y $\Delta Y'$, y los ángulos de la medición anterior (ΔX y ΔY), se tienen los parámetros completos para la función que evalúa el cambio de rumbo.

En el caso del punto de la iteración ($i = 6$) respecto al punto de la iteración anterior ($i = 5$), la información que se pasa al algoritmo 2 es la siguiente: $angle = 13,58^\circ$, $angleX' = -63,99^\circ$, $\Delta X = 6,5m$, $\Delta X' = 3,39m$, $\Delta Y = 1,57m$, $\Delta Y' = -6,95m$. Con una medida de $tolerancia = 20^\circ$ descartamos que el trayecto haya continuado en línea

Cuadro 4.1: Comparación de direcciones detectadas por GPS y acelerómetro en una recta.

Muestra GPS	Rumbo GPS	Rumbo Acel.
1	-1	2
2	1	2
3	2	2
4	2	2
5	3	2
6	0	2
7	2	2
8	2	3
9	2	3
10	2	3
11	2	3
12	0	2

recta. Las condiciones de la función determinan que el movimiento es del tipo 0, que es como se codifica la “Izquierda relativa”.

Es importante notar que este algoritmo no intenta determinar el cambio de dirección entre el primer par de puntos GPS, porque es necesario haber encontrado un ángulo de un par anterior de puntos GPS para determinar el cambio en el siguiente par.

Para calcular la dirección con información del acelerómetro según el algoritmo 3, se deben seleccionar las muestras de aceleración que están comprendidas entre el par de puntos GPS a evaluar. Esto se hace utilizando una función que encuentra el primer registro de aceleración con tiempo mayor al primer punto GPS y el último registro de aceleración marcado con un tiempo menor al segundo punto GPS del recorrido. Una vez que se tienen las muestras que se deben utilizar, se promedian ($mediaFuerzasX = -2,13m/s^2$ y $mediaFuerzasY = 5,12m/s^2$) y se obtiene el ángulo entre ellas de manera similar al GPS. Este ángulo $angle = -68,80^\circ$ se evalúa en el algoritmo 4, siendo clasificado como 2, o “Norte relativo”. Estas mediciones son coherentes con la ruta trazada con el automóvil y serán usadas para corregir el trazo marcado en rojo.

Los resultados de la evaluación de la dirección tomada por el automóvil se muestra en la tabla 4.1:

Cuadro 4.2: Consistencia de los puntos en una recta

Muestra GPS	Rumbo GPS	Rumbo Acel.	Consistencia
1	-1	2	NOK
2	1	2	OK
3	2	2	OK
4	2	2	OK
5	3	2	OK
6	0	2	NOK
7	2	2	OK
8	2	3	OK
9	2	3	OK
10	2	3	OK
11	2	3	OK
12	0	2	OK

Determinando la consistencia entre mediciones GPS y del acelerómetro

El algoritmo 5 marcará como puntos inconsistentes aquellos donde la diferencia de la dirección relativa reportada por el acelerómetro y el GPS sea mayor a 1. El resultado se muestra en la tabla 4.2

Corrigiendo el punto GPS inconsistente.

Una vez marcados los puntos, el algoritmo 6 buscará un punto consistente antes y después del punto 6, siendo los puntos consistentes 5 y 7. Estos puntos se envían a la función de interpolación lineal, devolviendo como resultado el punto 6 corregido como $X = 573167,213801$ e $Y = 2106994,037940$. La corrección puede verse cuando se vuelven a graficar los puntos GPS, como se muestra en la figura 4.10

Corrección de curvas

De manera similar a la corrección de rectas, en la figura 4.11 se muestra una curva trazada por MyTracks. El ejemplo de funcionamiento del algoritmo se centra en la corrección de los puntos finales de la curva, como se muestra en la figura 4.12.

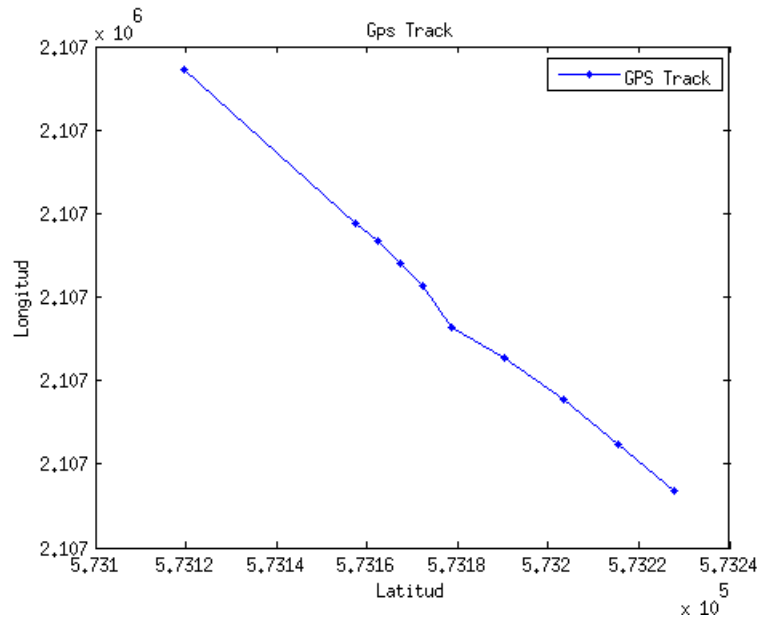


Figura 4.10: Trazo de recta a corregir el punto objetivo es el sexto en la secuencia, marcado en rojo.

Determinando la dirección del acelerómetro y GPS

El procedimiento es el mismo que en el caso de la recta, utilizando en el algoritmo 1. Pondremos atención en el punto $i = 16$ del recorrido, al que le corresponde un valor $\Delta X' = -7,64m$, $\Delta Y' = -9,43m$ y $angle' = 51^\circ$ así como $\Delta X = -7,95m$, $\Delta Y = -9,55m$ y $angle = 50,19^\circ$ que corresponden al punto anterior, $i = 15$. Esta información se pasa al algoritmo 2, que determina que el cambio de dirección es del tipo 2, lo que significa que el móvil se mantiene en línea recta desde el punto anterior.

De manera similar, para determinar la dirección del móvil con el acelerómetro, utilizamos el algoritmo 3 y 4. Después de encontrar las muestras de aceleración entre la muestra $i = 16$ y la muestra $i = 17$. Los promedios de las fuerzas en los ejes X e Y son $mediaFuerzasX = -37,03m/s^2$ y $mediaFuerzasY = 1,32m/s^2$ y el ángulo entre las componentes es $angle = -19,75^\circ$. La información del ángulo se evalúa con el algoritmo 4, y se determina que pertenece a la dirección 4, que significa un cambio de dirección hacia la izquierda relativa.



Figura 4.11: Trazo de curva a corregir. El ejemplo se centra en el final del recorrido.

En la tabla 4.3 se observan los resultados de la evaluación de la dirección de los puntos 15 en adelante.

Determinando la consistencia entre mediciones GPS y del aceler

Según el algoritmo 5, todos los puntos a partir del punto 16 (dibujados con rojo en la figura 4.12) son marcados como no consistentes porque la dirección reportada por el GPS difiere en más de uno con la medición del acelerómetro.

Corrigiendo el punto GPS inconsistente

Para la corrección de curvas, se utiliza el algoritmo 7, que solamente necesita ubicar un punto confiable antes del punto a corregir. En el caso del punto 16, el punto confiable que será usado en la corrección es el punto 15.

Cuadro 4.3: Comparación de direcciones detectadas por GPS y acelerómetro en una curva

Muestra GPS	Rumbo GPS	Rumbo Acel.
15	2	3
16	2	4
17	2	4
18	2	4
19	2	4
20	2	4
21	2	4
22	2	4

Cuadro 4.4: Comparación de direcciones detectadas por GPS y acelerómetro en una curva

Muestra GPS	Rumbo GPS	Rumbo Acel.	Consistencia
15	2	3	OK
16	2	4	NOK
17	2	4	NOK
18	2	4	NOK
19	2	4	NOK
20	2	4	NOK
21	2	4	NOK
22	2	4	NOK

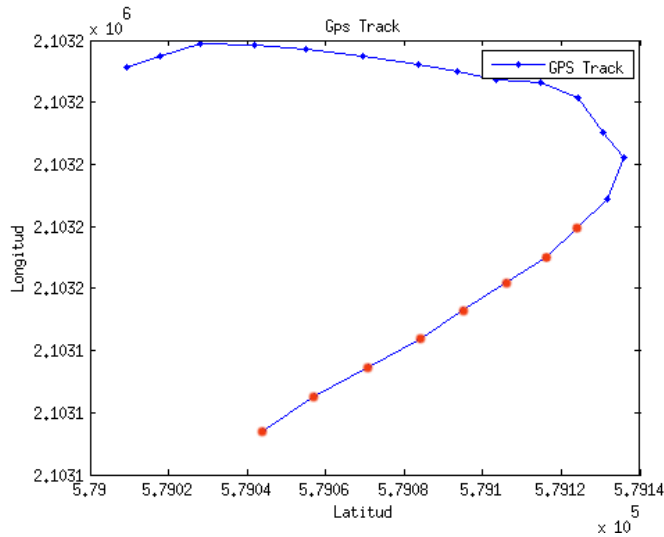


Figura 4.12: Trazo de curva a corregir. El ejemplo se centra en el final del recorrido.

La información de estos dos puntos se envía al algoritmo 8 que obtiene el cuadrante del plano cartesiano en el que se ubica el ángulo de movimiento. Para obtener el cuadrante, se analiza el cambio de dirección en las componentes del movimiento según la tabla ??, resultando en *cuadrante* = 3. A continuación, el ángulo entre el punto consistente y el punto inconsistente se modifica según el valor asignado a *cambio*, y que determinará cuánto se mueve el punto inconsistente de posición original. Para este punto, la corrección es hacia la izquierda relativa, y se agregan 10 grados al ángulo original.

Una vez reunida la información del cuadrante, el nuevo ángulo y el punto GPS a corregir, se utiliza el algoritmo 10 para determinar la dirección y magnitud de distancia con la que hay que mover el punto GPS. Con los valores *nuevoAngulo* = 61° y *cuadranteAnterior* = 3, se obtienen los valores *xdir* = -1 y *ydir* = -1. Esto significa que el punto describirá un movimiento en el sentido negativo tanto de la componente horizontal como vertical de el sistema de coordenadas UTM. La magnitud de las componentes se calcula utilizando el nuevo ángulo y las magnitud de desplazamiento total entre los puntos GPS. El resultado de las nuevas coordenadas se obtiene cuando se añaden a las coordenadas del punto consistente las nuevas magnitudes con su signo correspondiente.

Cuadro 4.5: Cuadrantes del movimiento de los puntos GPS. El símbolo + representa valores positivos, y el símbolo - representa valores negativos.

Cuadrante	Componentes	
	X	Y
1	+	+
2	-	+
3	-	-
4	-	+

Como resultado de este proceso, se obtienen las nuevas coordenadas del punto 16, con $GPS.utm_x = 579117,978035$ y $GPS.utm_y = 2103168,761828$. El resultado de iterar este proceso por los siguientes puntos se ilustra en la figura 4.13.

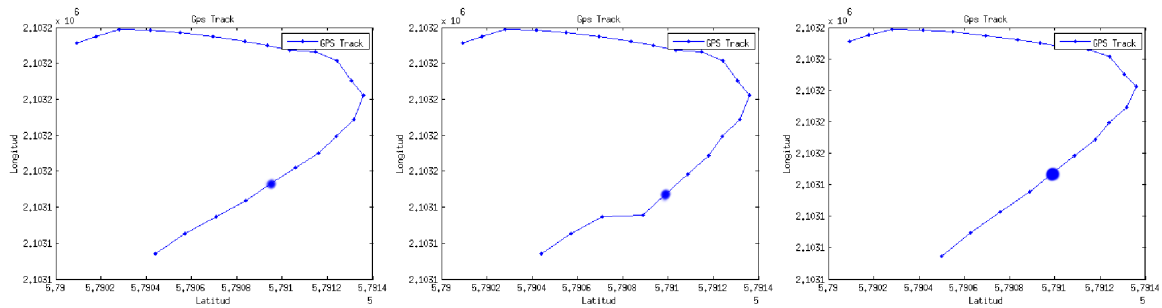


Figura 4.13: Secuencia de modificación de puntos GPS inconsistentes.

Es importante notar que este algoritmo mantiene la misma distancia entre los puntos, y solamente modifica el ángulo respecto a la horizontal del GPS.

4.7. Resultados experimentales

En esta sección se demuestra el funcionamiento de los algoritmos de trazado de mapas, de acuerdo con el modelo de sistema expuesto con anterioridad. La implementación se hace en matlab y android, poniéndose a disposición del lector el código y el conjunto de datos que se usaron en este experimento.

En las siguientes secciones se muestran los resultados de la corrección de dos curvas y dos rectas para el caso de la generación de mapas a partir de información de localización y acelerometría.

4.7.1. Configuración del experimento

En esta sección se describe el equipo utilizado y su configuración para recoger la información de posición y aceleración.

Hardware y software

Para la recolección de datos se utilizó un teléfono inteligente *Sony Ericsson Xperia Arc*. A continuación se muestran las características relevantes para los experimentos, pudiéndose encontrar la lista completa en [Son11].

- CPU: Qualcomm MSM8255 @ 1Ghz
- RAM: 512 MB
- Acelerómetro triaxial
- aGPS
- Sistema operativo Android Ice Cream Sandwich (4.0.4)

Para el procesamiento de los datos se utilizó una laptop *Dell XPS 15*, cuyas características relevantes son:

- CPU: Intel i3 @ 2.6Ghz
- RAM: 4 GB
- Sistema operativo: Ubuntu Linux 12.04
- Entorno de desarrollo: Eclipse Índigo (3.7.2) / Android Development Toolkit 18 y Matlab 2011a

Vehículo

Para la recolección de muestras en los experimentos, se utilizó un auto modelo Chevy C2, de la General Motors, tomando en cuenta la siguiente configuración:

1. Velocidad constante de 60 kmph del vehículo en zonas de tráfico fluido.
2. Velocidad constante de 25 kmph del vehículo en zonas de tráfico intenso o con gran flujo peatonal.
3. Se consideran trayectos cortos, de aproximadamente 200 metros.

Configuración del acelerómetro

En el caso del acelerómetro, el sensado es controlado por software, y no existe una manera efectiva de controlar la velocidad de muestreo del sensor. Esta situación se debe a que Android presenta una API universal para el sensado con acelerómetro. Considerando esta limitación, se eligió usar siempre la velocidad máxima de muestreo permitida por el sensor, a través de la directiva *SENSOR_DELAY_FASTEST*. Como resultado de esta configuración, se pudieron obtener muestras con una separación promedio de 8 milisegundos.

Configuración del aGPS

En el caso del aGPS, el sistema reporta los puntos GPS basado en un criterio de disponibilidad de información, no de frecuencia de muestreo y es compatible con este experimento.

Trazado y corrección de mapas con muestras de GPS y aceleración

El experimento se realizó en las inmediaciones de Ciudad Judicial y La Vista, en los límites de la Ciudad de Puebla y Cholula, así como en la zona de Ciudad Universitaria. Ambas localidades en el estado de Puebla, como se muestra en los mapas de las figuras 4.14 , 4.15 y 4.16.

Recolección de datos

Para la recolección de datos en la zona del periférico (Ciudad Judicial) se viajó a una velocidad de aproximadamente 60kmph. En cuanto a la zona del centro de Cholula,

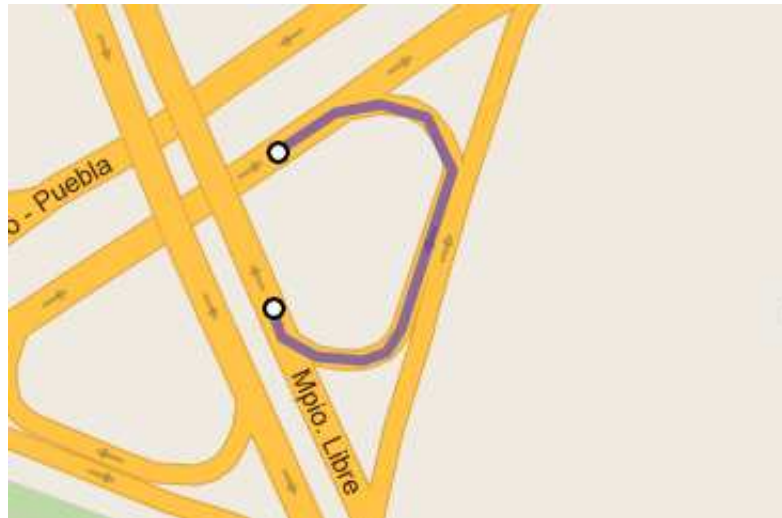


Figura 4.14: Recorrido en La Vista. Dibujado con GPSP prune

la velocidad de tránsito fue de aproximadamente 25kmph, de acuerdo a las normas de tránsito local y el comportamiento de la mayoría de los vehículos.

Durante el recorrido, el dispositivo móvil se fija en el automóvil de tal manera que los ejes del acelerómetro están de acuerdo a las figuras 4.2 y 4.3.

Después de realizar el recorrido, los datos son guardados como archivos de texto separados por comas para su procesamiento fuera de línea. En el archivo, por cada muestra tomada por el acelerómetro se tiene una línea con la estampa de tiempo en formato UNIX y la aceleración en X, Y y Z.

Preprocesamiento de datos

Para el caso del trazo de caminos, no es necesario contar con alguna frecuencia de muestreo específica, puesto que solamente se busca recolectar la mayor cantidad de muestras entre puntos GPS para hacer un promedio de fuerzas que afectan al móvil.

La información de tiempo proporcionada por *MyTracks* está codificada en el formato *año:mes:diaThora:minuto:segundo.milisegundoZ*, donde T es la separación entre fecha y hora y Z representa la zona horaria, por ejemplo 20120614T18:28:47.000Z. Esta información debe convertirse a un formato que sea fácilmente manipulable, con el fin de

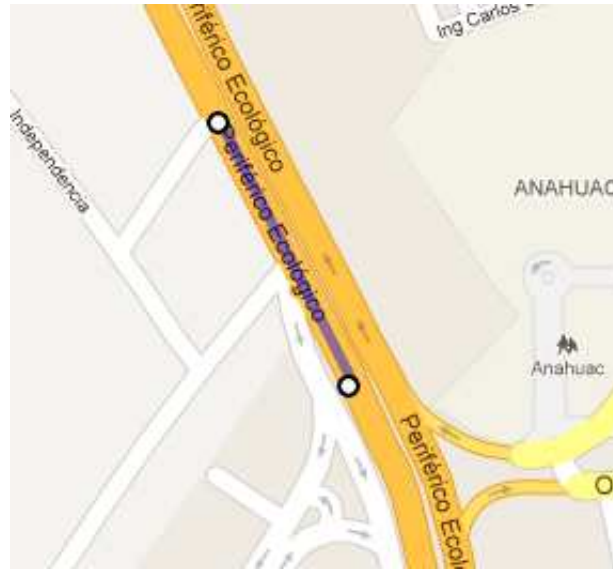


Figura 4.15: Recorrido en ciudad Judicial. Dibujado con GPSPRUNE

encontrar las mediciones del acelerómetro que corresponden al periodo de tiempo transcurrido entre dos mediciones GPS. Para lograr esto, ambas fechas son representadas en tiempo UNIX, que es un conteo de los segundos transcurridos desde el primero de enero de 1970 hasta la fecha. Por ser una representación con números enteros, las operaciones de comparación de tiempo son sencillas.

Rectas

Recta consistente

En la primera recta, ilustrada en la figura 4.15, podemos apreciar que las mediciones GPS son correctas. La tabla 4.6 demuestra que no hay inconsistencia entre las mediciones tomadas por el GPS y el acelerómetro. Debido a este resultado, los puntos GPS no son modificados y quedan como en la figura 4.17

Recta inconsistente

En el segundo ejemplo, que se muestra en la figura 4.18, se puede notar cómo varias muestras del GPS son erróneas. La inconsistencia de los puntos puede verificarse en



Figura 4.16: Recorrido en Ciudad Universitaria. Dibujado con GPSPruve

la tabla 4.7. A diferencia de la detección de inconsistencias en curvas, se considera como punto inconsistente todo aquel donde el valor de la dirección reportada por el acelerómetro y el GPS sean distintas.

El resultado de la aplicación del algoritmo se muestra en la figura 4.19

Curvas

Curva consistente

De manera similar a como fueron presentados los resultados de la corrección de rectas, se muestran los resultados de la operación de los algoritmos en una curva que

Cuadro 4.6: Comparación de direcciones detectadas por GPS y acelerómetro en una recta sin inconsistencias.

Muestra GPS	Rumbo GPS	Rumbo Acel.
1	-1	2
2	1	2
3	2	2
4	2	2
5	2	2
6	2	2
7	2	2
8	2	2
9	2	2
10	2	2
11	2	2
12	2	2
13	2	2
14	2	2
15	2	2
16	2	2
17	2	2
18	2	2
19	2	2
20	2	2

no presenta inconsistencia en los datos reportados por los sensores además de una curva que si muestra inconsistencias.

La primera curva se ilustra en la figura 4.20 , y corresponde al recorrido en la figura 4.16. Como demuestra la evaluación de consistencia en la tabla 4.8, al no haber una diferencia mayor a uno en las direcciones reportadas por el acelerómetro y el GPS, la curva no se corrige. Al igual que en ejemplos anteriores, el primer punto no se evalúa, porque el algoritmo no puede detectar el cambio de dirección sin un punto GPS previo.

Curva inconsistente

La segunda curva evaluada se muestra en la figura 4.21, tomada del recorrido que se ilustra en la figura 4.14. Esta curva muestra inconsistencias en algunas de sus mediciones, como puede observarse en la tabla de la figura 4.9. El resultado de la corrección puede observarse en la figura 4.22.



Figura 4.17: Trazo de recta sin puntos de GPS inconsistentes. Dibujado con GPSPRUNE

La segunda curva evaluada se muestra en la figura 4.21, tomada del recorrido que se ilustra en la figura 4.14. Esta curva muestra inconsistencias en algunas de sus mediciones, como puede observarse en la tabla de la figura 4.9. El resultado de la corrección puede observarse en la figura 4.22.

4.8. Análisis de resultados

Observando los resultados del proceso de detección y corrección de muestras GPS obtenidos en los experimentos descritos, se puede observar que:

1. En condiciones habituales, las muestras del GPS presentan pocos errores. Es a mayores velocidades y en zonas urbanas con edificios que bloquean la vista al cielo cuando la calidad de las muestras se degrada.



Figura 4.18: Trazo de recta a corregir con puntos GPS inconsistentes. Dibujado con GPSPruve

2. Aunque el acelerómetro puede proporcionar información para determinar si un cambio de dirección es correcto o no, no tiene manera de corregir la posición inicial de un muestra GPS.
3. El resultado de los trazos del GPS validados con el acelerómetro resultan en una mejor aproximación de la trayectoria descrita por el automóvil. Esta trayectoria no necesariamente coincide con la forma de la carretera, pero elimina los *outliers* y es una buena aproximación para un mapa inicial.

También puede notarse que la el desplazamiento de los puntos corregidos respecto a los puntos no consistentes es poco en la mayoría de los casos de corrección de curvas. Esto sucede porque aunque la detección de la inconsistencia se logra satisfactoriamente, aún debe considerarse cuál será el factor de corrección para la dirección y no puede adoptarse una estrategia de corrección agresiva. En otras palabras, aún debe considerarse

Cuadro 4.7: Comparación de direcciones detectadas por GPS y acelerómetro en una recta con inconsistencias.

Muestra GPS	Rumbo GPS	Rumbo Acel.
1	2	2
2	2	2
3	2	2
4	2	2
5	2	2
6	2	2
7	2	2
8	-1	2
9	2	2
10	1	2
11	2	2
12	2	2
13	-1	2
14	2	2
15	2	2
16	2	2
17	2	3
18	2	2
19	2	2
20	-1	2
21	2	2
22	2	2
23	2	2
24	2	2

qué tan severa es la desviación de dirección. Durante la serie de experimentos expuestos en la sección anterior, se corrigen las direcciones por 5 o 10 grados. El caso es distinto cuando se trata de recorridos en línea recta, puesto que la estrategia de corrección es una interpolación lineal.

4.9. Conclusiones

Aunque la detección y corrección de inconsistencias en la ubicación de los puntos GPS es satisfactoria por la eliminación de *outliers*, la corrección de los mismos puntos puede mejorarse estableciendo una métrica del desacuerdo entre los sensores de ubicación y aceleración.



Figura 4.19: Corrección de una recta con inconsistencias. Dibujado con GPSPPrune

La mejora de este método de corrección significaría no solamente obtener mejores trazos iniciales para un mapa de calles, sino la posibilidad de adaptar el método a herramientas de asistencia de navegación en la ciudad.

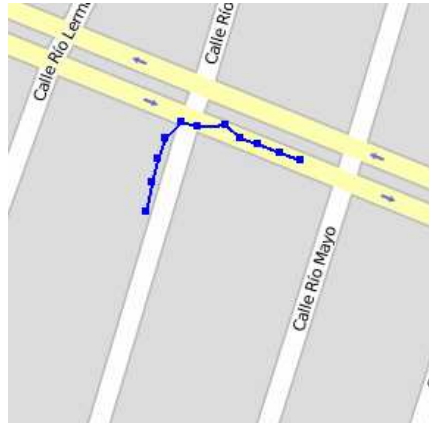


Figura 4.20: Curva sin inconsistencias entre dirección del acelerómetro y GPS. Dibujado con GPSPRUNE

Cuadro 4.8: Comparación de direcciones detectadas por GPS y acelerómetro en una curva sin inconsistencias.

Muestra GPS	Rumbo GPS	Rumbo Acel.
1	-1	2
2	2	2
3	2	2
4	2	2
5	1	1
6	2	1
7	1	2
8	2	2
9	2	2
10	2	2



Figura 4.21: Trazo de una curva con inconsistencias.



Figura 4.22: Corrección de una curva con inconsistencias. Dibujado con GPSPPrune

Cuadro 4.9: Comparación de direcciones detectadas por GPS y acelerómetro en una curva con inconsistencias.

Muestra GPS	Rumbo GPS	Rumbo Acel.
1	-1	2
2	2	2
3	2	1
4	2	2
5	2	4
6	1	0
7	2	0
8	1	0
9	2	1
10	2	0
11	2	1
12	2	2
13	2	2
14	2	0
15	2	0
16	2	4
17	2	0
18	2	1
19	2	0
20	0	0
21	2	0
22	2	0
23	1	2
24	2	2
25	0	2

Capítulo 5

Tipificación del terreno.

5.1. Introducción.

Durante este capítulo, se expondrá el mecanismo propuesto para tipificar el terreno utilizando información de acelerometría y una red neuronal para clasificar. En la primera sección se habla del modelo de sistema que sustenta la operación del mecanismo, y a continuación se hace un análisis de la información que se puede obtener del sensor de aceleración. A continuación, se explica cómo se determinan las clases para la tipificación de terrenos, además de la construcción del clasificador. El capítulo finaliza con una breve demostración del clasificador y el reporte de los resultados experimentales.

5.2. Modelo del sistema

El modelo del sistema está representado por la figura 4.1, donde se puede apreciar las fases de recolección de datos, preprocesamiento, entrenamiento y selección de una red neuronal para construir un clasificador.

Para esta sección, el modelo del sistema mantiene las mismas restricciones que el modelo descrito en el capítulo 4, salvo un cambio en cuanto a la velocidad del automóvil:

1. Velocidad constante según cada tipo de terreno.
2. No se permite el movimiento en reversa.
3. Cambios graduales en la dirección de movimiento, con diferencias no mayores a 90 grados entre puntos consecutivos.

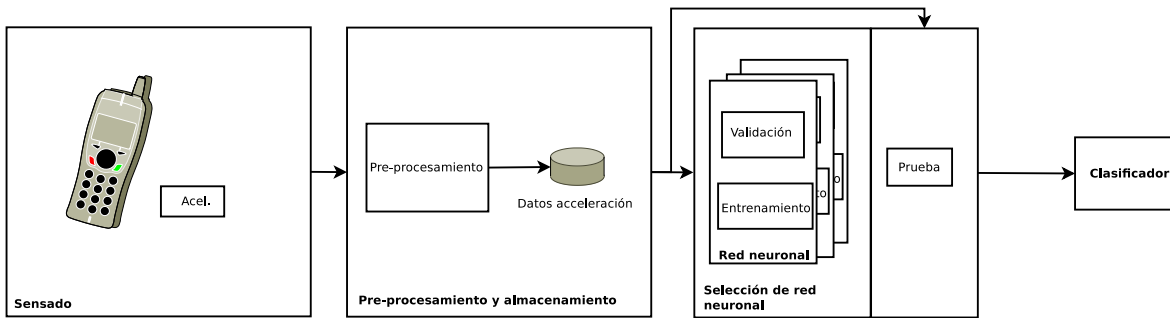


Figura 5.1: Modelo del sistema para la generación de mapas a partir de información de localización y acelerometría.

4. El vehículo tiende a moverse en mayor proporción “hacia adelante” respecto a cuanto se mueve “hacia los lados”. Este comportamiento es congruente con el movimiento que describe un automóvil al desplazarse en una calle.
5. Los ejes del acelerómetro están fijos durante el trayecto y alineados con el automóvil: el eje X del acelerómetro coincide con el eje definido entre la parte frontal y posterior del automóvil, mientras que el eje Z coincide con el eje definido entre los lados del vehículo. El eje Y del acelerómetro está comprendido entre el piso y el techo del automóvil. Esta alineación de ejes se ilustra en la figura 4.2 y la figura 4.3.

5.3. Análisis de la información proporcionada por el acelerómetro

Los acelerómetros, al ser instrumentos que miden la fuerza específica o aceleración inercial, son especialmente útiles para capturar características del movimiento que está relacionado con la fuerza detectada. En el caso de un acelerómetro triaxial, como el que se utiliza en este proyecto, cuando se le ubica en el marco de referencia de un automóvil según el modelo del sistema, podemos identificar:

1. Eje X : información sobre la sacudida en el plano horizontal del vehículo. Las fuerzas aplicadas a este eje son el resultado de cambios de dirección del automóvil. Dichos

cambios de dirección pueden asociarse con los intentos del conductor por evitar obstáculos o irregularidades del terreno en caminos accidentados. Las fuerzas en este eje también se ven afectadas por los cambios de velocidad, aunque en menor medida.

2. Eje Z: información sobre los cambios de velocidad en el automóvil. El uso del acelerador y el freno afectan directamente a las mediciones en este eje y pueden asociarse también con los intentos del conductor por evitar obstáculos. También está influenciado por el movimiento laterar del automóvil, aunque en mucho menor grado que el eje X. El eje X y Z son considerados como el plano de movimiento del automóvil.
3. Eje Y: las irregularidades del terreno se manifiestan en las fuerzas aplicadas al eje Y del acelerómetro, que coincide con el eje vertical del automóvil. Un terreno uniforme inducirá muy poca variación en las mediciones del instrumento, mientras que un terreno accidentado inducirá una gran variación en las mediciones del acelerómetro.

Con el fin de ilustrar el comportamiento listado y sus efectos en las mediciones, se presenta un cuadro comparativo en la figura 5.2, que representan las señales registradas en los eje X,Y y Z del acelerómetro. En estas figuras se aprecia una aceleración repentina, el registro de un terreno accidentado y finalmente, un cambio repentino de dirección.

En la primera fila de imágenes, que representa la aceleración repentina del automóvil, se puede observar un incremento constante en los valores de aceleración del eje Z, resaltado en color rojo. En la segunda fila, que representa el paso por un empedrado, se puede observar gran varianza en el eje Y, además de que los valores de la aceleración se encuentran en valores poco habituales (llegando a los $16m/s^2$) respecto a los valores alcanzados en otras actividades. La señal del eje Y se muestra en color amarillo. En cuanto a la tercera fila, que representa el cambio repentino de dirección del vehículo, es fácil observar cómo el cambio de valores de la señal (de positivo a negativo, y viceversa) en el eje X y coloreada en verde es marcadamente distinto al de los otros dos ejes.

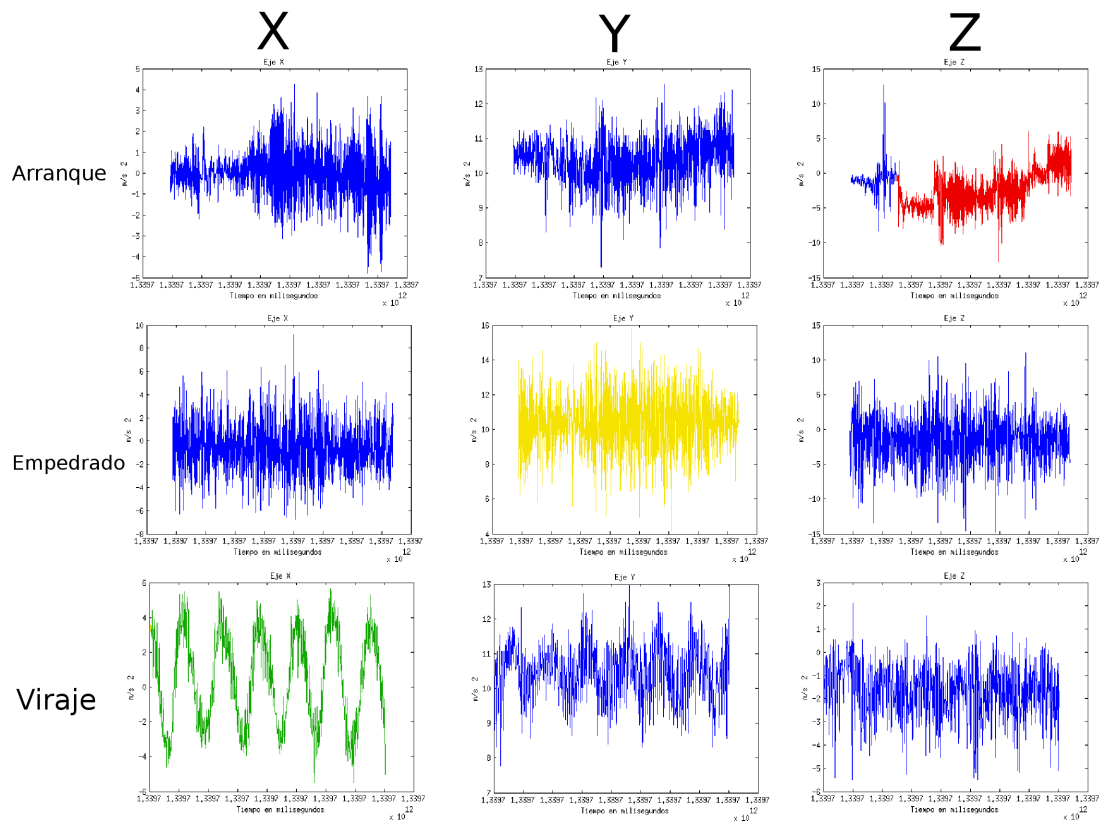


Figura 5.2: Comparación de la respuesta en los 3 ejes del acelerómetro ante movimientos del automóvil.

También es importante considerar que las mediciones del acelerómetro son ruidosas, y que además de necesitar una etapa de postprocesamiento, es conveniente aplicar transformaciones a la señal obtenida para poder definir características útiles en el contexto de un clasificador.

5.4. Determinación de las características

Una vez que se ha determinado que es posible extraer información sobre las condiciones del terreno que afectan la inercia del automóvil, es necesario encontrar un conjunto de características que puedan ser presentadas a un clasificador.

Como se ha mencionado en la sección anterior, la señal de la cual se extraerán las características debe ser sometida a un preprocesamiento, que en este caso consiste en una etapa de normalización y de descomposición con transformada wavelet.

Normalización

Considerando f_x , f_y y f_z como las componentes de la señal en los ejes X , Y y Z respectivamente, obtenemos la señal normalizada \hat{f}_r mediante las ecuaciones (5.1) , (5.2) , (5.3) [Pé12].

$$f_r = \sqrt{f_x^2 + f_y^2 + f_z^2} \quad (5.1)$$

$$\bar{f}_r = f_r - \text{promedio}(f_r) \quad (5.2)$$

$$\hat{f}_r = f_r - \bar{f}_r \quad (5.3)$$

División de la señal en ventanas de tiempo

Con el fin de obtener segmentos de la señal de los cuales obtener características que puedan ser utilizadas en un clasificador, se divide el total de la señal en ventanas con algún número determinado de muestras o tiempo. En el caso de este trabajo, se utilizan ventanas de 250 muestras, traslapadas en 125 muestras, o el 50% del tamaño de la ventana. Esta configuración se utiliza por haber demostrado facilitar la extracción de características en trabajos pasados [RDML] y se ilustra en la figura 2.6 del capítulo de fundamentos teóricos.

Descomposición de la señal con la transformada Wavelet

Ya contando con la señal normalizada y dividida en ventanas de tiempo, se procede a utilizar la transformada wavelet de la familia Daubechies. Esta transformada divide

a cada ventana en cuatro regiones según su frecuencia, con un avance logarítmico. A cada región corresponde un conjunto de coeficientes, de los cuales se obtiene su media y varianza. Estos cuatro grupos de medias y varianzas conforman un vector de características de ocho posiciones, que serán utilizados para el entrenamiento del clasificador. Es importante señalar que este procedimiento fué tomado del trabajo realizado en el laboratorio CPU del INAOE [Pé12].

La elección de usar la media y la varianza de los coeficientes fue motivada por el trabajo relacionado en cuanto procesamiento de señales para obtener características contextuales a partir de datos de acelerometría [FDFC10].

5.5. Determinación de las clases

Este trabajo establece 5 clases de terreno:

1. A: reposo
2. B: Pavimento
3. C: Adoquín
4. D: Empedrado
5. D: Terracería

Estas clases se eligieron según la calidad del terreno, comenzando con el menos accidentado e irregular (el reposo) y llegando hasta el mas irregular y accidentado (la terracería). Es importante aclarar que la clase de terracería comprende solamente el terreno no tratado, puesto que la terracería puede ser tanto un camino de tierra aplanado (que es prácticamente indistinguible del pavimento) como un camino rural, que suele ser muy accidentado e irregular.

5.6. Construcción de un clasificador

Para resolver el problema de clasificación de la señales, se ha elegido una red perceptrón multicapa, por su capacidad de resolver problemas de clasificación que no son linealmente separables y su resistencia al ruido en los datos de entrada.

En cuanto a su configuración, se eligió utilizar una capa oculta y probar con el número de neuronas en ella. Esta decisión se ha tomado apoyándose en el teorema de aproximación universal, que establece que *una sola capa oculta es suficiente para una red perceptrón multicapa para calcular una aproximación ϵ a un conjunto de entrenamiento representado por las entradas x_1, \dots, x_m y una salida deseada $f(x_1, \dots, x_m)$* [Hay98].

Para evaluar las distintas configuraciones de la capa oculta en la red, se elaboró un conjunto de entrenamiento, prueba y validación. Los tamaños de capa oculta que se probaron están en el rango de 20 a 250 neuronas. El proceso de evaluación se describe a continuación.

Evaluación del desempeño de las redes MLP

Esta evaluación se realizó utilizando un modelo de 3 dobleces (3fold validation). El conjunto de señales recopiladas por el acelerómetro y sometidas al preprocesamiento se divide en un grupo de entrenamiento y otro de validación. A su vez, el grupo de entrenamiento se divide en tres secciones que se utilizarán para entrenar y probar cada configuración de la red en tres ocasiones. Para hacerlo, se combinan dos de los tres subgrupos de vectores de características en un grupo de entrenamiento y un grupo de validación que se usará durante una de las tres iteraciones de prueba. En cada iteración se altera el orden de los grupos a combinar, y de esta forma podemos obtener un promedio y una varianza del porcentaje de reconocimiento de la red que está siendo evaluada.

Al final del proceso, podemos comparar los promedios y las varianzas de todas las redes y elegir la que haya tenido el mejor desempeño, para hacer la prueba de generalización con los datos nunca antes presentados al clasificador (grupo de validación). Los resultados obtenidos se presentan en la matriz de confusión en la tabla 5.1. El resultado

de generalización es del 80 % para la red elegida, la cual tiene 185 neuronas en la capa oculta.

Cuadro 5.1: Matriz de confusión de la red neuronal con mejor desempeño.

	A	B	C	D	E
A	20	0	0	0	0
B	0	20	0	0	0
C	0	2	11	2	5
D	0	1	0	18	1
E	0	5	2	2	11
Generalización: 80%					

5.7. Ejemplo de funcionamiento del clasificador

En esta sección se presentan las características de dos señales correctamente clasificadas, pertenecientes a la clase de terracería y pavimento respectivamente.

Terracería

La señal normalizada puede observarse en la figura 5.3. Su transformación wavelet se muestra en la figura 5.4.

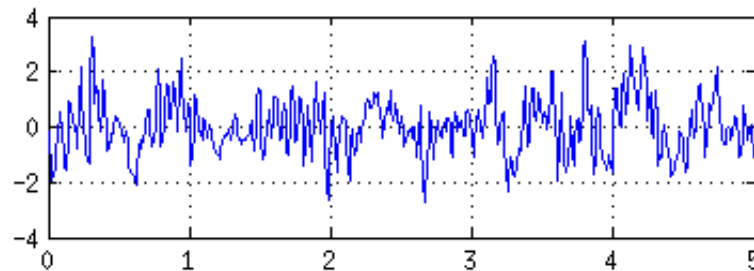


Figura 5.3: Señal de aceleración en los tres ejes del acelerómetro en terracería.

El vector de características \bar{x} de la señal, luego de aplicar la transformación wavelet en tres niveles y obtener los coeficientes de cada nivel se muestra en (5.4). El vector de características es reconocido por el clasificador como perteneciente a la clase de terracería.

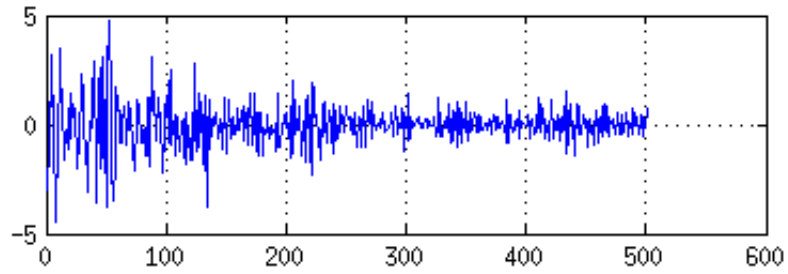


Figura 5.4: Transformación wavelet de la señal de terracería.

$$\bar{x} = [1,583, 1,989, 0,376, 0,501, 0,742, 0,941, 0,974, 1,185] \quad (5.4)$$

Pavimento

De la misma forma, puede observarse la señal normalizada en la figura 5.5. Su transformación wavelet se muestra en la figura 5.6 y el vector de características que se extrae de los coeficientes se muestra en (5.5).

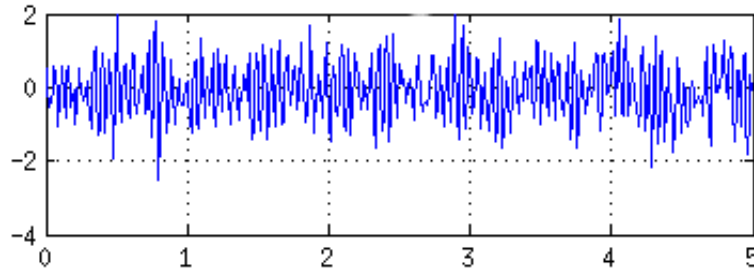


Figura 5.5: Señal de aceleración en los tres ejes del acelerómetro en pavimento.

$$\bar{x} = [0,617, 0,721, 0,525, 0,624, 0,906, 1,076, 0,435, 0,540] \quad (5.5)$$

5.8. Resultados experimentales

En esta sección se demuestra el funcionamiento del clasificador usado en la tipificación de terrenos, de acuerdo con el modelo de sistema expuesto con anterioridad. La

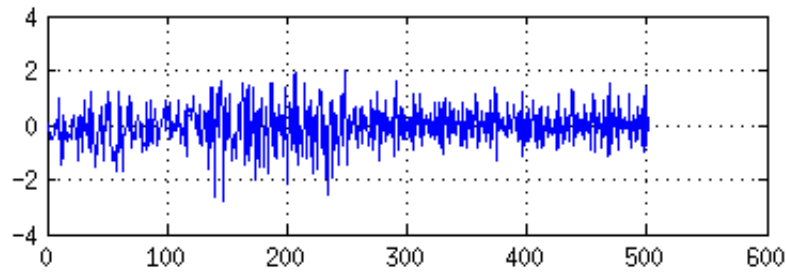


Figura 5.6: Transformación wavelet de la señal de pavimento.

implementación se hace en Matlab 2011a y Android 4.0.4, poniéndose a disposición del lector el código y el conjunto de datos que se usaron en este experimento.

Se explica el proceso mediante el cual se recogieron las muestras con las que se entrenó al clasificador y con las que se evaluó su desempeño.

5.8.1. Configuración del experimento

En esta sección se describe el equipo utilizado y su configuración para recoger la información de posición y aceleración. La configuración es muy similar a la descrita en el capítulo 4.

5.8.2. Hardware y software

Para la recolección de datos se utilizó el mismo equipo descrito en el capítulo 4: un teléfono inteligente *Sony Ericsson Xperia Arc* y una computadora *Dell XPS 15*.

5.8.3. Vehículo

Para la recolección de muestras en los experimentos, se utilizó un auto modelo Chevy C2, de la General Motors, tomando en cuenta la siguiente configuración:

1. La velocidad del vehículo:
 - a) Reposo: No aplica.

- b) Pavimento: 60 - 80 kmph
- c) Adoquín: 20 - 40 kmph
- d) Empedrado: 0 - 10 kmph
- e) Terracería : 0 - 10kmph

2. Se consideran trayectos cortos, de aproximadamente 200 metros.

5.8.4. Configuración del acelerómetro

En el caso del acelerómetro, el sensado es controlado por software, y no existe una manera efectiva de controlar la velocidad de muestreo del sensor. Esta situación se debe a que Android presenta una API universal para el sensado con acelerómetro. Considerando esta limitación, se eligió usar siempre la velocidad máxima de muestreo permitida por el sensor, a través de la directiva *SENSOR_DELAY_FASTEST*. Como resultado de esta configuración, se pudieron obtener muestras con una separación promedio de 8 milisegundos.

5.8.5. Recolección de datos

La recolección de muestras para el empedrado se realizó en las inmediaciones del INAOE, en Santa María Tonanzintla (ver figura 5.7). Las muestras de terracería se tomaron de la zona cercana a las Pirámides de Cholula, como se muestra en la figura 5.8. Las muestras correspondientes al adoquín se tomaron de la Avenida Juárez, en la ciudad de Puebla 5.9. Para el caso de las muestras de reposo, no se hace ningún recorrido, dado que las muestras se toman cuando el automóvil no está en movimiento. En el caso de las muestras que se tomaron en pavimento, se utilizó el mismo recorrido en Ciudad Judicial que se muestra en la figura 4.15.

Los datos se recolectan con una frecuencia de muestreo de 100 Hz, recolectándose 20,000 muestras por recorrido. Estos datos se almacenan en archivos de texto para su procesamiento fuera de línea. El dispositivo se monta en el automóvil de tal manera que sus ejes queden alineados.



Figura 5.7: Recorrido en empedrado. Dibujado en GPSPRUNE



Figura 5.8: Recorrido en una zona de terracería. Dibujado en GPSPRUNE

5.8.6. Preprocesamiento de datos

Para el experimento de tipificación de terreno, es importante contar con una frecuencia de muestreo constante, por lo que se aplicó una interpolación lineal a los datos con el fin de tener un periodo promedio de 10 milisegundos entre muestras, lo que significa una frecuencia de muestreo de 100hz.

Además de la etapa de interpolación de las muestras, los datos se agrupan en ventanas de 250 muestras, con un traslape del 50%. Cada ventana es tratada por la transformada wavelet Daubechies 1 (ver capítulo 2). De los coeficientes de cada ventana se obtiene



Figura 5.9: Recorrido en adoquín.

la media y varianza en tres niveles de descomposición, mismos que son utilizados como vector de características [Pé12].

5.8.7. Clasificación

Los datos preprocesados y divididos en ventanas de tiempo son agrupados de acuerdo a lo descrito en el capítulo de tipificación de terreno.

Las características de la red perceptrón multicapa utilizada para clasificar son las siguientes:

- Capas ocultas: 1
- Neuronas en la capa oculta: 185
- Función de activación: Sigmoide, descrita en la ecuación (2.5)
- Método de aprendizaje: traingdx, del toolbox de matlab 2011a.

Esta configuración reporta un 80 % de generalización evaluados sobre recorridos con 20,000 muestras registradas.

5.9. Análisis de resultados

Al analizar la matriz de confusión (ver tabla 5.1), podemos observar que gran parte del error se debe a que la clase *C* (adoquín) y la clase *E* (terracería) se confunden

entre sí. Esto puede deberse principalmente a a que las características presentadas a la red neuronal contiene información contradictoria, que impide un mejor aprendizaje al sistema.

El preprocesamiento de los datos también es importante. El tamaño de la ventana de muestras que se presenta al clasificador, así como la frecuencia con que las muestras son tomadas son factores de impacto en los resultados de la clasificación, como demuestran los trabajos de [HS05] En este experimento se utiliza un acercamiento conservador, de tal manera que se utiliza una sola configuración de tamaño de ventana (250 muestras) y frecuencia (100 Hz) durante todo el experimento

5.10. Conclusiones

El resultado de la tipificación de terrenos es satisfactorio, en el sentido de que se logró un 80% de precisión en la clasificación y de que no hay hasta la fecha algún trabajo que trate con ese problema.

El uso de técnicas apropiadas de preprocesamiento es esencial para lograr un buen desempeño del clasificador. Desafortunadamente, determinar la mejor técnica y parámetros para un problema en particular es una tarea que requiere una gran inversión de tiempo. De manera similar, la búsqueda de características apropiada para este problema aún se mantiene abierta, dando a lugar posibles mejoras a la solución de este problema en el futuro. Es importante mencionar, que aunque la tipificación de terreno y el trazado de mapas se han considerado por separado, cabe la posibilidad de estudiar la relación entre ambos problemas. En otras palabras se propone el uso de la información del tipo de terreno para asistir la generación de mapas. Como ejemplo, sería posible obtener un estimado del cambio de distancia estimado entre dos puntos GPS dependiendo de la velocidad que se puede desarrollar habitualmente en un tipo de terreno.

Capítulo 6

Conclusiones y trabajo futuro

En este trabajo de investigación se propone un método para trazar calles georeferenciadas a partir de muestras de puntos GPS y de acelerometría. La relevancia de esta aportación reside en la capacidad del algoritmo de combinar la información de dos tipos de sensores distintos para ofrecer una calidad de trazo superior a la que se obtendría con un radio GPS o aGPS. Aunque las técnicas de integración de rutas no son nuevas, y este algoritmo puede catalogarse como tal, su aplicación sólo se ha mostrado con equipo altamente especializado y preciso, a diferencia del equipo utilizado aquí.

Adicionalmente, las técnicas de corrección empleadas en este trabajo son muy simples. Se propone como trabajo futuro en la generación de mapas georeferenciados el uso de técnicas mas avanzadas, como el uso de filtros Kalman para mejorar la corrección de los trazos GPS [And]. También se propone establecer una medida del desacuerdo entre los sensores involucrados en la determinación de coherencia de los trazos, para mejorar la corrección automatizada.

Por la parte de la tipificación de terrenos, fue posible extraer la información contextual sobre el estado del terreno para establecer una escala de calidad del camino, logrando clasificar en 5 clases de caminos de acuerdo a tal escala. El método propuesto es de importancia al no existir hasta la fecha un método que enfrente la clasificación de terrenos en caminos y porque hace posible obtener información sobre el estado del camino para su uso en aplicaciones de cómputo ubicuo y particularmente sistemas concientes del entorno. Como trabajo futuro se propone la investigación de técnicas de preprocesamiento mas adecuadas para la actividad de detección de características del camino, así como

la investigación de sistemas de aprendizaje automático que pudieran facilitar la clasificación del terreno, como los mapas auto-organizativos [Hay98]. Se propone también el estudio detallado de las características más significativas para extraer de las señales del acelerómetro tomando en cuenta el problema de la tipificación de terrenos. El contar con características especialmente definidas para este problema podría mejorar mucho el reconocimiento en un sistema de aprendizaje automático.

Una propuesta de trabajo futuro que incumbe tanto a la tipificación del terreno como la generación de mapas georeferenciados es la investigación en el modelo distribuido de dichas tareas. Durante este trabajo se considera un modelo centralizado y con un procesamiento fuera de línea, y se considera que para una aplicación a gran escala es necesario distribuir la carga del proceso en la mayor cantidad de dispositivos posible. Aún más a favor de un modelo descentralizado es el hecho de que contar con una gran cantidad de muestras, que pueden ser recogidas por varios dispositivos móviles, ayuda a disipar el efecto negativo de la poca precisión de los sensores.

Bibliografía

- [And] Angus P Andrews. *Global positioning systems, inertial navigation and integration*.
- [ANN10] Gabriel Agamennoni, Juan I Nieto, and Eduardo M Nebot. Robust and Accurate Road Map Inference. *IEEE International Conference on Robotics and Automation*, pages 3946–3953, 2010.
- [Axe05] Axel Küpper (Ludwig Maximilian University). *Location-based Services Fundamentals and operation*. Wiley, 2005.
- [CK] Guanling Chen and David Kotz. A Survey of Context-Aware Mobile Computing Research. *Time*, pages 1–16.
- [CN01] H. Choset and K. Nagatani. Topological simultaneous localization and mapping (SLAM): toward exact localization without explicit localization. *IEEE Transactions on Robotics and Automation*, 17(2):125–137, April 2001.
- [Cos00] A C F Coster. Classification of basic daily movements using a triaxial accelerometer. *Frontiers of Medical & Biological Engineering*, 10(4):369–372, December 2000.
- [DaBH06] J.J. Davies, a.R. Beresford, and a. Hopper. Scalable, Distributed, Real-Time Map Generation. *IEEE Pervasive Computing*, 5(4):47–54, October 2006.

- [EGH⁺08] Jakob Eriksson, Lewis Girod, Bret Hull, Ryan Newton, Samuel Madden, and Hari Balakrishnan. The Pothole Patrol : Using a Mobile Sensor Network for Road Surface Monitoring. 2008.
- [F. 09] M.Ferreira F. Lima. Mining spatial data from GPS traces for automatic road network extraction. *International Symposium on Mobile Mapping Technology*, 2009.
- [FDFC10] Davide Figo, Pedro C. Diniz, Diogo R. Ferreira, and João M. P. Cardoso. Preprocessing techniques for context recognition from accelerometer data. *Personal and Ubiquitous Computing*, 14(7):645–662, March 2010.
- [Fou] Foursquare. <https://foursquare.com/>.
- [Har07] Robert K. Harle. Spatial Indexing for Location-Aware Systems. *2007 Fourth Annual International Conference on Mobile and Ubiquitous Systems: Networking & Services (MobiQuitous)*, pages 1–8, 2007.
- [Hay98] Simon Haykin. *Neural Networks: A Comprehensive Foundation (2nd Edition)*. Prentice Hall, 2 edition, July 1998.
- [HH94] Andy Harter and Andy Hopper. A distributed location system for the active office - IEEE Network. *Ieee Network*, pages 62–70, 1994.
- [HS05] Tâm Huynh and Bernt Schiele. Analyzing Features for Activity Recognition. (october), 2005.
- [INE00] Proyecto carta topográfica escala 1:1'000,000. pages 1–61, 2000.
- [Lat] Google Latitude. <http://latitude.google.com>.
- [LMD⁺07] Hui Liu, Student Member, Houshang Darabi, Pat Banerjee, and Jing Liu. Survey of Wireless Indoor Positioning Techniques and Systems. *37(6):1067–1080*, 2007.

- [LML⁺10] Nicholas D Lane, Emiliano Miluzzo, Hong Lu, Daniel Peebles, Tanzeem Choudhury, Andrew T Campbell, and Dartmouth College. Ad Hoc and Sensor Networks, a Survey of Mobile Phone Sensing. *IEEE Communications Magazine*, (September):140–150, 2010.
- [MCS⁺07] Rossana Muscillo, Silvia Conforto, Maurizio Schmid, Paolo Caselli, and Tommaso D’Alessio. Classification of motor activities through derivative dynamic time warping applied on accelerometer data. *Conference proceedings : ... Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Conference*, 2007:4930–3, January 2007.
- [Mit99] D. Mitrovic. Driving event recognition by hidden Markov models. *4th International Conference on Telecommunications in Modern Satellite, Cable and Broadcasting Services. TELSIKS’99 (Cat. No.99EX365)*, 1:110–113, 1999.
- [MPR08] Prashanth Mohan, Venkata N Padmanabhan, and Ramachandran Ramjee. TrafficSense : Rich Monitoring of Road and Traffic Conditions using Mobile Smartphones. 2008.
- [MT] Practical Machine and Learning Tools. *Data mining, practical machine learning tools and techniques*. 2nd edition.
- [NSY09] Ben Nham, Kanya Siangliulue, and Serena Yeung. Predicting Mode of Transport from iPhone Accelerometer Data. *stanfordedu*, 2009.
- [Opp] Oppenheim. *Discrete-time signal processing*. Prentice Hall.
- [Pau87] Scott A. Wood. Paul R. Cooper, Daniel E. Friedman. The automatic generation of digital terrain models from satellite images by stereo. *Acta Astronautica*, 15(3):171–180, 1987.

- [Pé12] Victor Ernesto Alonso Pérez. Clasificación de tipos de terreno usando la transformada wavelet discreta. May 2012.
- [RDML] Nishkam Ravi, Nikhil Dandekar, Preetham Mysore, and Michael L Littman. Activity Recognition from Accelerometer Data. *Neural Networks*.
- [RGWNSM07] a. Roxin, J. Gaber, M. Wack, and a. Nait-Sidi-Moh. Survey of Wireless Geolocation Techniques. *2007 IEEE Globecom Workshops*, pages 1–9, November 2007.
- [RLUC] Mike Rizzo, Peter F Linington, Ian A Utting, and Kent Ct. Integration of Location Services in the Open Distributed Office. pages 1–14.
- [Sha98] C.E. Shannon. Communication In The Presence Of Noise. *Proceedings of the IEEE*, 86(2):447–457, February 1998.
- [SMSL09] Wenhuan Shi, Student Member, Shuhan Shen, and Yuncai Liu. Automatic Generation of Road Network Map from Massive GPS Vehicle Trajectories. pages 48–53, 2009.
- [Son11] Sony Ericsson. LT15i, LT15a. Technical Report August, 2011.
- [TPB06] Rudolph Triebel, Patrick Pfaff, and Wolfram Burgard. Multi-Level Surface Maps for Outdoor Terrain Mapping and Loop Closing. *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2276–2282, October 2006.
- [Tra92] Discrete Wavelet Transforms. Discrete Wavelet Transforms : Theory and Implementation Tim Edwards. (September 1991):1–26, 1992.