UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

POSGRADO EN CIENCIA E INGENIERÍA DE LA COMPUTACIÓN

"ESTIMACIÓN DEL MOVIMIENTO PROPIO A PARTIR DE UNA SERIE DE IMÁGENES"

T E S I S

QUE PARA OBTENER EL GRADO DE:

MAESTRA EN CIENCIAS (COMPUTACIÓN)

P R E S E N T A:

ALICIA MONTSERRAT ALVARADO GONZÁLEZ

DIRECCIÓN: "DR. YANN FRAUEL"
CODIRECCIÓN: "DR. FRANCISCO ESCOLANO"

México, D.F. **2007.**

Agradecimientos

Humberto, mi gran amor, mi mejor amigo. Espero nunca lograr hacer que lo que siento por ti quepa en una palabra. Ni dejar de tener la sensación de que lo que siento es tan grande que me desgarra por estarse desbordando. No puedo contar las batallas que hemos librado juntos tantos años. No tengo palabras para agradecerte tanto amor, respeto, apoyo, libertad, fortaleza, amistad… Es tan poco el espacio y tanto que decir. Sobre este tiempo, gracias por proveerme de lo necesario para facilitarme la vida, por darme la libertad para viajar, aprender y desarrollarme en todos los aspectos, y, sobre todas las cosas, gracias por entender.

Papás, mil gracias por darme las bases que me han permitido seguir. Por educarme para no dejarme vencer por lo que, a veces, parece imposible lograr, y para gritarle al mundo que nadie va a decirme hasta donde puedo llegar, solo yo misma. Sobre todo, gracias por enseñarme a valorar todo. Y sí, hay más de un camino, gracias a ustedes existe. Tenemos mucho que vivir juntos, quédense a mi lado, tengo mucho que aprender de ustedes aún.

Hermano, el mundo tiene magia, te juro que existe. Solo debes tener tus antenas paradas para captarla. El universo tiene mucho que decir, sus mensajes pueden ser descifrados. Atento. Escucha. Te indican tu camino. Y cuando eso sucede, ya no hay más dudas de hacia donde continuar.

M. en C. Wendy Aguilar, yo insisto amiga, yo llegué aquí para conocerte a ti. Gracias por ser mi traductora, mi maestra, mi terapeuta y mi mejor amiga. Sin ti, este camino hubiera sido mucho más difícil. Tu corazón tiene tanta bondad y tu cerebro tanta inteligencia, que aquel que pueda estar cerca de ti, tiene muchas bendiciones. Hicimos buen equipo en muchas cosas y hemos sido grandes complementos en otras. Ojala que el camino nos mantenga juntas mucho tiempo.

Rodrigo García, gracias por todas esas tardes en compañía. Tu amistad fue fundamental para mantenerme en el camino. Y ya sabes no?

Dr. Yann Frauel, gracias por abrirme las puertas cuando nadie quería tomar el proyecto. Por tenerme tanta paciencia con mis necedades, por ser tan buen guía y por darme tanta libertad para explorar.

Dr. Francisco Escolano, gracias por recibirme en su grupo de investigación en la Universidad de Alicante, España. Su guía fue fundamental para desarrollar este proyecto.

Dra. Elena Martínez, Dr. Edgar Garduño, Dr. Boris Escalante y Dr. Fernando Arámbula, muchas gracias por haberse tomado el tiempo para revisar mi tesis y hacerme sus mejores comentarios. Me hicieron sufrir pero quedó mucho mejor.

Dr. Rafael Pérez y Pérez, eres uno de los motores más importantes en mi vida profesional. Gracias por motivarme tanto y por creer en mí. Tu entusiasmo contagia. Esa es tu cualidad más especial.

Alberto, sin ti, no hubiera logrado entrar al posgrado, ya no digamos salido. Muchas gracias por creer en mí y por enseñarme.

Gibrán, Crevel, David Esparza, Paty y Jerry, gracias por compartir los proyectos en los que hicimos equipo y por esforzarse tanto para fueran de los mejores.

Oswaldo, Ramiro, Gauss y David Flores, gracias por colorear mis días en el IIMAS.

Lulú y Diana, gracias por hacerse cargo de todo, sin ustedes, esto hubiera sido un desastre.

CONACyT, gracias por otorgarme la beca para poder llevar a cabo mis estudios de posgrado.

Contenido

Índice de figuras

Índice de cuadros

Resumen

Para poder recuperar la rotación instantánea del observador y la dirección de translación mientras se mueve a través del ambiente (movimiento propio), se aplicó la estimación mediante la percepción del entorno. Una de sus etapas es la correspondencia entre puntos. Si ésta no es solucionada adecuadamente, la estimación no será precisa.

Por este motivo, en esta tesis se implementan algoritmos que permiten encontrar el número de correspondencias correctas, necesario para llevar a cabo la estimación del movimiento. Dichos algoritmos se basan en la Geometría Epipolar (GE) y el *"Thin-Plate spline"* (TPS).

El objetivo del TPS es encontrar una transformación no rígida entre los conjuntos de puntos correspondientes de dos imágenes. Como una de sus desventajas es que la solución es muy costosa, debido a que durante su cálculo se debe invertir una matriz cuadrada del tamaño del número de puntos correspondientes, se implementó el algoritmo RANSAC-TPS. El cual selecciona un subconjunto de puntos correspondientes, aplicando el RANSAC de forma incremental, a partir del cual se obtienen los coeficientes de la transformación, para luego, aplicarlos al total de los puntos de la primera imagen. Esto da como resultado unos puntos que deben ser aproximados a los de la segunda imagen mediante la minimización de la energía de la deformación, que busca la transformación óptima entre ellos. Con lo cual se obtiene no solo la transformación no rígida, sino puntos correspondientes. Sin embargo, se creó el algoritmo ISRANSAC-TPS, con el que se disminuye la aleatoriedad de la selección de las muestras y, con esto, se obtienen mejores resultados que el RANSAC-TPS, pero por su costo computacional, no es útil para aplicaciones en tiempo real.

Luego, con el objetivo de obtener más y mejores correspondencias, se aplicó la geometría epipolar a los algoritmos anteriores. La mezcla de ambas técnicas dio origen al algoritmo GE-TPS, el cual permite obtener muy buenas correspondencias, sin embargo, en algunos casos, no se obtienen las necesarias. Por lo tanto, se implementó el algoritmo GE-TPS-GE-CT. Éste obtiene una matriz fundamental que representa mejor la geometría de todos los puntos y, con ella, se filtran los datos de entrada, de esta manera se incrementan las correspondencias y se conservan las transformaciones.

Por otro lado, para resolver el problema de la estimación del movimiento propio se implementaron algoritmos basados en la matriz esencial y algoritmos basados en el método directo. Se hicieron experimentos con datos controlados y se encontró que los mejores resultados se obtienen con el algoritmo robusto.

Introducción

La visón es considerada como uno de los sentidos de mayor uso en la vida de las personas [Audette, R. et al., 2000], por lo tanto, los humanos somos seres visuales. Si bien el conjunto de los sentidos nos permiten relacionarnos con el entorno, se estima que cerca del 80 por ciento de la información de que disponemos, la obtenemos a través del sentido de la vista.

Las tareas diarias como viajar, encontrar direcciones y leer señales y documentos, son muy difíciles de realizar para personas con ceguera o debilidad visual.

De acuerdo con los tabuladores temáticos sobre la población con discapacidad del Censo General de Población y Vivienda 2000, reportados por el INEGI, en México, de una población de aproximadamente 97.4 millones de personas, 1.7 millones son discapacitadas y, de éstas, cuatrocientas mil (es decir, el 0.48% de la población total), tienen discapacidades visuales. Lo cual la coloca en el segundo lugar luego de la discapacidad motora.

De esta población, el 47.7% no recibe ingresos, el 29.21% es población ocupada; de los que son mayores de 4 años, el 33% no tienen instrucción, el grado promedio de escolaridad es tercero de primaria, el 13.71% la termina, el 6.22% termina la secundaria, el 4.85% termina el nivel medio superior o estudios técnicos y sólo el 3.5% tienen instrucción superior.

Por otro lado, la Organización Mundial de la Salud reporta que en el 2002, de un total de aproximadamente 6.200 millones de personas en el mundo, 161 millones (es decir, el 2.59% de la población total) tienen alguna discapacidad visual, de las cuales, 37 millones de ellas son ciegas y 124 millones son débiles visuales. El 3.9% de éstas nace con ésta discapacidad y el resto la adquieren por enfermedades, accidentes u otras.

En algunos países ha tomado tanta importancia atender las necesidades de este grupo minoritario, que han instituido organismos o centros de investigación especializados en ello. Por ejemplo, en España, la Organización

Nacional de Ciegos Españoles estimula la investigación y el desarrollo tecnológico a través del establecimiento de nexos de colaboración estrechos con universidades, empresas y entidades públicas y privadas, mediante diferentes estrategias de apoyo a la investigación. Su financiamiento ha permitido que se diseñen y desarrollen productos de tecnología para el acceso a la información y autonomía personal. Por una parte, se desarrollan productos de software: convertidores de textos de tinta a braille, reconocedor óptico de textos en tinta, magnificadores y lectores de voz de pantallas de computadoras, diccionarios y enciclopedias electrónicas, juegos de computadoras, etc. Y, por otra, aparatos y recursos materiales como: anotadores electrónicos, radio lupa, impresoras y monitores/líneas braille, bastones, avisadores sonoros y táctiles, etc.

Por otro lado, en el Centro de Investigación en Ingeniería de Rehabilitación, localizado en San Francisco California, E. U. A., el Instituto de Investigación sobre el Ojo Smith-Kettlewell desarrolla nuevas tecnologías y métodos para entender, tasar y rehabilitar ciegos y débiles visuales. Su población está compuesta de ciegos, débiles visuales y ciegos con sordera. El principal financiamiento del instituto es en gran parte a través del Instituto Nacional de Discapacidades e Investigación en Rehabilitación. Uno de los proyectos del Instituto son las aplicaciones de visión computacional para personas ciegas, del cual se encarga, por un lado, el laboratorio de Visión Computacional liderado por el investigador James M. Coughlan, y por otro, el Grupo Sendero, liderado por Mike May. Entre las tareas que tratan de resolver están:

1. Analizar intersecciones de tráfico (ej. encontrar un cruce peatonal cercano que provea una adecuada orientación).
2. Proveer información del terreno para usuarios ciegos en sillas de ruedas (usando cámaras estéreo para construir mapas 3D que permitan ubicar curvas, rampas y obstáculos).
3. Encontrar y leer textos en ambientes desordenados (como señalamientos en las calles).
4. Adaptar las tareas anteriores para que funcionen con las cámaras de teléfonos celulares (como encontrar y leer textos y códigos de barras).
5. Herramientas para viajar y encontrar el camino.

Otros grupos de investigación también han llegado a trabajar en proyectos de esta índole. Incluso, debido al creciente número de investigadores, en visión computacional, que se han interesado en aplicaciones para personas con ceguera o debilidad visual, en el 2005 se inauguró el primer Taller IEEE en aplicaciones de Visión Computacional para las Personas Ciegas (CVAVI por sus siglas en inglés), como parte del congreso en Visión Computacional y Reconocimiento de Patrones (CVPR por sus siglas en inglés) que se llevó a cabo en San Diego, California, E. U. A. En el que se juntaron investigadores de visión computacional y expertos en rehabilitación y tecnología asistida para ciegos. Los principales temas que se abordaron fueron:

1. Herramientas para viajar y encontrar el camino.
2. Interfaces visuales, auditivas y táctiles.
3. Acceso a la información.

El desarrollo de herramientas asistidas para la comunidad ciega, requiere un conocimiento informado de varios factores humanos relevantes, así como temas de tecnología, entre los que se pueden mencionar los siguientes:

1. Problemas actuales que impiden la calidad en la vida diaria de estos usuarios.
2. La falta de conocimiento de la existencia de primeros auxilios asistidos.
3. Un entendimiento realista de las posibilidades ofrecidas por las interfaces disponibles (visuales, auditivas, táctiles).
4. Una aproximación a nivel de sistemas para el diseño de algoritmos y el "*hardware*", que toma en consideración factores prácticos tales como el tamaño, la velocidad y el costo.

La inquietud de hacer este proyecto surge de la necesidad de ayudar a invidentes a vencer una de las barreras más significativas para ellos: navegar libremente por un ambiente desconocido. Esto implica responder dos preguntas: 1) ¿Dónde estoy con respecto al resto del mundo? y 2) ¿Cómo llegar a un destino dado?

Para resolver este problema, en algunos países se ha implementado la tecnología de Señales Parlantes. La cual usa transmisores infrarrojos, ubicados en lugares específicos, que emiten rayos invisibles a unos receptores portátiles, los cuales decodifican estos rayos en mensajes de voz. Estos indican si hay cruces peatonales, paradas de autobús, o teléfonos públicos. A pesar de ser bien aceptada, tiene desventajas, como el costo de instalación y el mantenimiento.

También se han desarrollado herramientas útiles haciendo uso del GPS, el cual ofrece una precisión de hasta 3 metros. Los sistemas basados en esta tecnología proveen localizaciones relativamente buenas en exteriores, sin embargo, es casi inservible en interiores. Para resolver el problema en este tipo de ambientes, se han propuesto varias alternativas mediante la visión computacional, que hace uso de imágenes obtenidas con cámaras económicas, como las que tienen los teléfonos celulares o en general cualquier cámara CCD, que son de uso común y proveen un amplio rango de visibilidad dependiendo de la resolución. De hecho, 1.3 mega píxel de una cámara de un celular tiene una visión de 20/80 lo cual es ¼ de la resolución humana de 20/20 [Coughlan, J. et al., 2006].

El problema de la detección, descripción y comprensión del movimiento, son de las tareas más difíciles de resolver en esta área, e implican un gran reto. Ya que, en un bajo nivel, el movimiento 3D debe ser analizado con base en la apariencia 2D y en la evolución de las características que son observables en las imágenes. En un alto nivel, los campos de movimiento, previamente derivados, deben ser capaces de distinguir entre objetos que tienen distintos movimientos, estimar los parámetros del movimiento, etc. [Demirdjian, D. and Horaud, R., 2000].

El objetivo de esta tesis es contribuir en la solución de una parte de esta problemática, al estimar el movimiento de una persona a partir de una secuencia de imágenes consecutivas, tomadas con una cámara monocular, obteniendo puntos característicos correspondientes entre ellas. Esto con base en el conocimiento de que los humanos

tenemos dos sistemas de navegación basados en la integración de rutas y en puntos característicos visuales precisos [Foo, P.S. et al., 2004].

La estimación del movimiento es un problema que, como se verá en el capítulo 1, ha sido ampliamente investigado y se han generado una gran cantidad de alternativas para resolverlo. Una de las aproximaciones, que se desarrollará en este trabajo, es la llamada estimación mediante la percepción del entorno, cuyo problema es que una de sus etapas, la correspondencia entre puntos, no ha sido solucionada satisfactoriamente, y esto impide que la estimación sea precisa.

Por lo tanto, la principal aportación de esta tesis es la implementación de algoritmos que permitan encontrar el número de correspondencias correctas, necesario para llevar a cabo la estimación del movimiento. Dichos algoritmos se basan en la Geometría Epipolar (GE) y el *"Thin-Plate spline"* (TPS), las cuales han sido ampliamente utilizadas en varias áreas del conocimiento (i.e. ingeniería, gráficos por computadora, análisis de imágenes), entre ellas, la visión computacional. Ambas técnicas, así como la notación necesaria para entender el resto de la tesis, serán explicadas en el capítulo 2.

La mezcla de ambas técnicas dio origen al algoritmo llamado GE-TPS, el cual permite obtener muy buenas correspondencias, sin embargo, en algunos casos, no se obtienen las necesarias. Por lo tanto, se implementó un algoritmo diseñado para incrementarlas, llamado GE-TPS-GE-CT. En el capítulo 3 se explican dichos algoritmos, y se presentan una serie de experimentos para verificar su funcionamiento.

Finalmente, en el capítulo 4, se presentan los resultados de la implementación de cuatro algoritmos, probados con datos controlados, que permiten estimar el movimiento.

PARTE 1
FUNDAMENTOS

Capítulo 1

Estimación del movimiento: Generalidades

En un robot móvil (del cual se hablará en esta tesis) se pueden identificar subsistemas de percepción, planificación, control de movimientos y locomoción, que interaccionan entre sí mientras navega, los cuales le brindan autonomía. El sistema de percepción le permite al robot adaptarse a situaciones cambiantes del entorno y a reaccionar ante posibles eventos inesperados. La información del entorno es proporcionada mediante un sistema sensorial. Esta información debe permitir al robot realizar tres tareas fundamentales: estimar su posición, orientación y movimiento; generar o actualizar el mapa del entorno; y detectar los posibles obstáculos. Las cuales requieren distintos tipos de información dependiendo de la velocidad de respuesta que se espere de ellas. Así, mientras que para la estimación de la posición y la construcción (o actualización) del mapa del entorno se tienen en cuenta, sobre todo, características como la precisión, la resolución espacial, el alcance, etc., en la detección de obstáculos, el tiempo entre observaciones normalmente debe ser mucho menor que en las tareas anteriores, resultando vital el disponer de la información ya procesada lo más rápidamente posible. En este caso, las características anteriores no son primordiales. En algunos robots, la estimación de la posición y la actualización del mapa recaen sobre cámaras de video ó escáneres láser, mientras que la detección de obstáculos se realiza mediante sónares.

Existe una estrecha relación entre el problema de la estimación de la posición y el de la construcción del mapa del entorno, siendo necesaria la localización precisa del robot para poder resolver ambos. Para que un robot pueda llevar a cabo tareas como la generación de trayectorias o la evasión de obstáculos, se requiere que este sea capaz de determinar su localización (posición y orientación) con respecto a un sistema de referencia absoluto. De forma general, determinar la localización de un robot equivale a encontrar las componentes de translación $\left(t_x, t_y, t_z\right)$ y de rotación $\left(\theta_x, \theta_y, \theta_z\right)$ del sistema de coordenadas del robot, con respecto a un sistema absoluto.

La estimación de la posición o del movimiento de un robot ha recibido una considerable atención por parte de numerosos investigadores, quienes han propuesto una gran variedad de técnicas. La mayoría de los robots están

provistos de codificadores en los ejes de movimiento, lo que permite estimar en cada instante su ubicación. Sin embargo, esta estimación no resulta suficientemente precisa para la mayoría de las aplicaciones. El motivo se debe, fundamentalmente, al hecho de que los errores se van acumulando durante la navegación, lo que origina que la región de incertidumbre asociada a la posición y orientación del robot vaya creciendo conforme éste se mueve. De esta forma, cada vez que ésta supere unos determinados límites, el robot necesitará de algún sistema de posicionamiento que reduzca la incertidumbre. Estos límites vienen impuestos por el tipo de escenario, la naturaleza de la tarea a realizar y la precisión requerida en los movimientos. Asimismo, la precisión alcanzada con el sistema de posicionamiento dependerá de la técnica y del sensor utilizados.

De acuerdo con [González, J. and Ollero, A., 1996], las principales técnicas de estimación de la posición o del movimiento se pueden categorizar en dos grupos principales: estimadores explícitos y estimadores basados en la percepción del entorno.

1.1 Estimadores de movimiento explícitos

Los estimadores explícitos proporcionan la posición y la orientación del robot a partir de medidas más o menos directas del sensor, sin que ello exija una interpretación del entorno. Los dos grupos que se consideran dentro de estos son [Burgard, W. et al., 1996]: los sistemas de posicionamiento relativo (como el sistema odométrico o el sistema de navegación inercial) y los sistemas de posicionamiento absoluto (como los sistemas de localización mediante estaciones de transmisión, terrestres u orbitales, y los sistemas de navegación mediante balizas).

La estimación de la posición y la orientación usando sistemas de posicionamiento relativo se basa en medidas internas del robot. Consiste en integrar la trayectoria recorrida por éste a partir de una serie de mediciones, tales como: vueltas dadas por las ruedas, velocidades, aceleraciones y cambios de orientaciones, obtenidas a partir de sensores integrados en el vehículo (como los codificadores, giróscopos, acelerómetros ó tacómetros) y sin ningún tipo de información exterior. Estas mediciones localizan al robot con respecto a su punto de partida, integrando cada uno de sus movimientos. Pueden distinguirse dos grupos fundamentales (en función de la información que se utilice):

a) Sistemas odométricos. Estiman la posición y la orientación de un vehículo a partir del número de vueltas (las cuales se cuentan con codificadores ópticos de elevada precisión) dadas por sus ruedas. La gran ventaja de la odometría reside en su simplicidad y bajo costo. Sin embargo, además de necesitar una frecuente calibración (como consecuencia del desgaste y pérdida de presión de las ruedas, ó del desajuste de los ejes), esta técnica es vulnerable a considerables imprecisiones causadas fundamentalmente por el deslizamiento de las ruedas e irregularidades en el suelo. Otro hecho que afecta a la estimación son las variaciones en la carga transportada por el vehículo. Este se puede evitar diseñando un modelo que, a partir del peso y distribución de la carga, permita corregir las desviaciones introducidas.

b) Sistemas de navegación inercial. Estiman la posición y orientación del vehículo empleando medidas de aceleración y ángulos de orientación. Las aceleraciones se miden con el acelerómetro, cuyos errores, aunque sean pequeños, repercuten notablemente en la posición estimada (debido a la doble integral con la que se calcula la posición a partir de la aceleración). Los ángulos de orientación se miden con giróscopos, los cuales pueden ser mecánicos (masa giratoria) u ópticos (de anillo láser ó de fibra óptica), aunque también se pueden medir con brújulas. Estos sistemas, en la práctica, son mucho más fiables y precisos que los sistemas basados en odometría, debido a que no se ven afectados por los problemas derivados de la interacción del vehículo con el suelo y a que pueden corregir los efectos de las ondulaciones e irregularidades del terreno. Aunque, como contrapartida, son más frágiles y caros.

Ambos sistemas son fáciles de implementar, sin embargo, los errores se van acumulando y la incertidumbre crece proporcionalmente al espacio recorrido. Por lo que, generalmente, se complementan con algún otro sistema de posicionamiento absoluto que reduzca periódicamente dicha incertidumbre (generalmente basados en la percepción del entorno).

La estimación de la posición y la orientación usando sistemas de posicionamiento absoluto permiten localizar a un robot respecto a un sistema de coordenadas fijo o global y basan su funcionamiento en señales externas. Un ejemplo de este tipo de sistemas son las estaciones de transmisión, las cuales están configuradas, por un lado, de modo que el receptor esté montado sobre el vehículo y, por otro, de modo que el emisor esté ubicado en una posición conocida del entorno. Las estaciones de transmisión pueden dividirse en:

a) Terrestres. Este tipo de sistemas se usan en aplicaciones marítimas, aeronáuticas, de robots móviles para exteriores y de vehículos autónomos que navegan en todo tipo de terreno. La ventaja de esta técnica es que proporciona la localización absoluta del vehículo, en un área suficientemente grande, sin requerir estructuración alguna del entorno. La configuración de estos sistemas está basada en un receptor a bordo del vehículo y en un conjunto de estaciones transmisoras de ondas de radio.

b) Orbitales. Se consideran dos diferentes grupos: sistemas de posicionamiento mediante estaciones fijas y sistemas de posicionamiento mediante estaciones móviles. Los primeros utilizan, fundamentalmente, señales de radio de media y alta frecuencia, que son emitidas por estaciones terrestres fijas. Los sistemas de estaciones móviles operados desde satélites son los de mayor interés para los robots móviles. El término genérico estándar para los sistemas de navegación satelital es Sistema Satelital de Navegación Global ("*Global Navigation Satellite System*", GNSS), el cual es un conjunto de satélites que transmite rangos de señales utilizados para el posicionamiento y la localización en cualquier parte del globo terrestre (ya sea por mar, tierra o aire) [University of Newcastle,2007d]. Los GNSS más desarrollados hasta el momento son:

- El Sistema de Posicionamiento Global ("*Global Positioning System"*, GPS) [NAVCEN: United States Coast Guard's Navigation Cente, 2007c], el cual fue diseñado por el Departamento de Defensa de los Estados Unidos. Consta de 30 satélites, en 6 órbitas casi circulares, a una altitud de 20.200 kilómetros (4 satélites por órbita). Cada uno de ellos transmite dos señales de radio (en alta frecuencia) en las que se codifica cierto tipo de información (tal como el instante en el que la señal fue transmitida o la información orbital).

  El receptor calcula (por triangulación) la longitud, la latitud y la altura del vehículo, de forma instantánea y continua, utilizando al menos 3 satélites. También puede determinar la velocidad a partir del desplazamiento en frecuencias mediante el efecto Doppler. Este tipo de sistemas de posicionamiento es también conocido como Radio-Navegación.

  La precisión del GPS depende de numerosos factores, algunos comunes a otros sistemas de posicionamiento mediante radio (prestaciones del receptor, inestabilidades en el recorrido de la transmisión, o la posición relativa del receptor con respecto a las estaciones), y otros, específicos del GPS (alteración de la velocidad de propagación de la señal de radio por la ionosfera, o errores en el posicionamiento orbital). Esta precisión puede llegar a ser de 10 a 20 metros para vehículos en movimiento, y algo inferiores para medidas estacionarias. Las medidas pueden llegar a ser más precisas con la utilización simultánea de un mayor número de satélites. La estimación precisa de la posición mediante GPS ha estado restringida tradicionalmente a aplicaciones militares. Sin embargo, en los últimos años, el empleo del GPS diferencial, basado en la corrección del error utilizando una estación base de coordenadas conocidas, está permitiendo incrementar de forma importante la precisión hasta un par de metros.

- El Sistema Satelital de Navegación Global (*"GLObal NAvigation Satellite System",* GLONASS) [Andrews Space & Technology, 2007a], el cual fue desarrollado por la Unión Soviética y, actualmente, es administrado por el Gobierno de la Federación Rusa, por las Fuerzas Espaciales Rusas y operado por la Coordinación Científica del Centro de Información del Ministerio de Defensa de la Federación Rusa. Es equivalente al GPS. Está conformado por 24 satélites (de los cuales 21 están activos y 3 son de repuesto), en 3 planos orbitales con 8 satélites cada uno, a una altitud de 19.000 kilómetros. Cada satélite transmite dos tipos de señales a distintas frecuencias, las cuales pueden ser recibidas por usuarios en cualquier parte de la Tierra para identificar su posición y su velocidad en tiempo real.

- El Sistema de Posicionamiento Galileo (*"Galileo Positioning System"*, Galileo) [European Space Agency, 2007b], desarrollado por la Unión Europea y la Agencia Espacial Europea. Es una constelación de 30 satélites y estaciones terrestres, que proveen información de la posición de los usuarios del sistema, triangulando de la misma forma que lo hace el GPS. Estará ubicado a una altura de 23.616 kilómetros.

Actualmente está funcionando parcialmente y se espera que esté en funcionamiento en su totalidad para el 2008.

Otro ejemplo de este tipo de sistemas es la estimación mediante balizas, la cual permite determinar la posición del vehículo, en un entorno restringido, colocando un determinado número de balizas en una ubicación conocida. La posición se estima de una forma más o menos directa, en base al principio de triangulación, ya sea a partir de medidas de distancias, de ángulos ó combinaciones de los dos.

Existen distintas configuraciones para implantar físicamente este tipo de sistemas. Una opción es adaptarle al vehículo un receptor giratorio que busque señales emitidas por las balizas. Un ejemplo de este tipo de configuración consiste en una serie de balizas de luz infrarroja y un dispositivo óptico giratorio abordo, capaz de detectar este tipo de luz. Conocida la velocidad de giro del receptor óptico, el sistema determina los ángulos entre balizas consecutivas a partir del tiempo que transcurre entre las detecciones de éstas. La posición del robot se estima con base en estos ángulos mediante relaciones trigonométricas. Mientras que la orientación se obtiene directamente midiendo el ángulo entre cualquiera de las balizas del entorno y otra, colocada abordo con tal fin. Otra opción es marcar con señales luminosas (i.e. códigos de barras) un entorno, y dotar al vehículo de una o varias cámaras CCD que exploren el entorno en busca de ellas.

Hay que tomar en cuenta que la precisión y la fiabilidad de este tipo de estimación dependen, fundamentalmente, del tipo de señal utilizada (infrarrojos, láser, radio, ultrasonidos, etc.), de las características del sensor y del número de balizas utilizadas en la triangulación. En general, la estimación mediante balizas está considerada como una de las más precisas. Las principales desventajas radican en la necesidad, tanto de configurar apropiadamente el entorno de trabajo, como de garantizar que un suficiente número de estas señales quede en todo momento libre de oclusiones y dentro del campo visual del sensor. También deben tenerse en cuenta los problemas que pueden originar las condiciones ambientales de iluminación y ruido (i.e., acústico, electromagnético). Todo ello impide la utilización de esta técnica en entornos muy dinámicos o no-estructurados.

1.2 Estimadores basados en la percepción del entorno

Una alternativa a los estimadores explícitos son los estimadores basados en la percepción del entorno. Éstos utilizan sensores que suministran información sobre el entorno (marcas naturales, puntos de interés, entorno completo percibido, etc.), para que el robot pueda inferir su localización de forma autónoma. Esto se logra haciendo una correspondencia entre la información (obtenida con los sensores) y los datos previamente conocidos acerca del entorno, los cuales pueden provenir de un modelo conocido del entorno, o de anteriores observaciones (ver figura 1.1).

Las técnicas desarrolladas en esta tesis, para resolver el problema de la estimación del movimiento propio, se fundamentan en este tipo de estimadores. Esto debido a que es un paso esencial en muchos problemas de visión computacional [Tian, T. et al., 1996]. Tal como, la utilización de una cámara para la estimación de la posición y la orientación de un objeto con respecto a un marco de referencia, a la posición de otro objeto, o a la posición de él mismo en un tiempo previo.

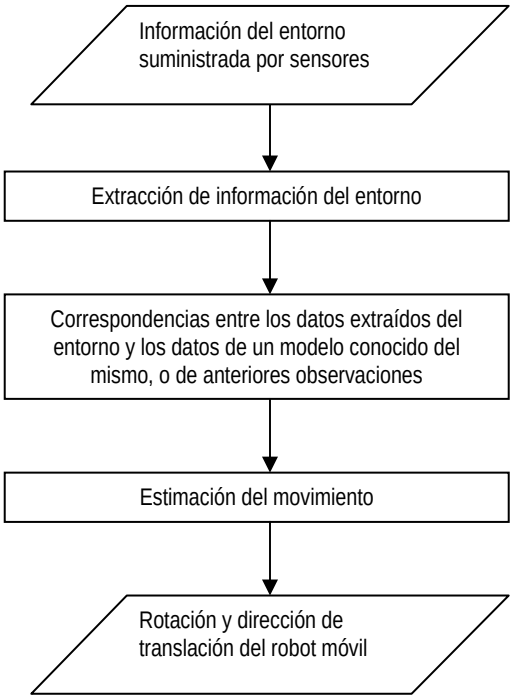


Figura 1.1 Esquema general de la estimación del movimiento propio basado en la percepción del entorno.

1.2.1 Sensores

El sistema sensorial puede operar con base en distintos tipos de sensores, los cuales pueden agruparse en: activos y pasivos.

a) Los sensores activos son aquellos que emiten algún tipo de energía al medio (por ejemplo: luz infrarroja, ultrasonidos, luz láser y ondas de radio) y proporcionan directamente medidas de distancias del entorno.

El principio de funcionamiento más utilizado es el denominado tiempo de vuelo, el cual consistente en medir el tiempo transcurrido entre la emisión de una señal y la recepción del eco correspondiente. A partir de esta medida y, teniendo en cuenta la velocidad de propagación de la señal, es posible estimar la distancia a la que el objeto ha sido interceptado. Uno de los principales inconvenientes de estos sensores surge cuando se

encuentran más de un robot en la misma área de trabajo, pudiendo producirse interferencias entre sus sistemas sensoriales. Los sensores activos más utilizados son los sónares y los sensores láser:

- Los sónares, también conocidos como sensores de ultrasonido, emiten un pulso que se propaga por el aire a la velocidad del sonido. Las dos configuraciones más utilizadas en robots móviles son: el anillo de sónares y el dispositivo de rastreo radial.

  Las principales ventajas de los sónares son su bajo costo y su simplicidad. Sin embargo, son muy vulnerables a reflexiones especulares (es decir, que un objeto parezca brillante o liso) y al medio; presentan una muy pobre resolución angular (sobre 30 grados) y una escasa fiabilidad para ángulos de incidencia mayores de 15 grados. Estas limitaciones obligan al desarrollo de técnicas probabilísticas, como las celdas de ocupación, las cuales permiten la integración de miles de lecturas conforme el robot navega, mientras representan el espacio en una matriz de celdas; cada una de las cuales viene caracterizada por una estimación de la probabilidad de que esté ocupada por algún objeto del entorno. Estas estimaciones son obtenidas a partir del modelo de incertidumbre de los sónares utilizados (generalmente distribuidos en forma de anillo). La estimación de la posición se realiza mediante una correlación a varios niveles de resolución entre el mapa global y el mapa local, obtenido desde la posición a estimar. Sin embargo, no es precisa ni fiable.

- Los sensores láser, a pesar de que tienen un costo elevado, están cobrando cada vez mayor importancia en el campo de los robots móviles, gracias a su precisión y fiabilidad. Utilizando uno o dos espejos convenientemente sincronizados, es posible direccionar el rayo de luz láser, configurándose así un sensor de rastreo bidimensional o tridimensional, respectivamente. La luz láser utilizada opera en las proximidades del infrarrojo, lo que hace que estos sensores sean prácticamente insensibles a las condiciones de iluminación del entorno. Esto debido a que la radiación infrarroja es de mayor longitud de onda que la luz visible, consecuentemente tiene menor frecuencia que ésta, lo que evita que la luz visible interfiera con la luz láser. Asimismo, la longitud de onda de la señal, la reducida divergencia del haz y las características propias de la luz láser, permiten obtener un eco, incluso para superficies con un bajo coeficiente de reflectividad (del orden del 10%).

  Con base en la manera en que la señal láser es transmitida, el funcionamiento de estos sensores puede clasificarse en tres grupos: i) detección de pulsos, en el que la distancia se estima midiendo el tiempo de vuelo entre la emisión y la detección de un pulso láser; ii) modulación de amplitud, en el que se transmite una señal láser modulada en amplitud, obteniendo la distancia a partir de la diferencia de fases entre la onda emitida y la onda recibida; iii) modulación de frecuencia, donde la distancia es proporcionada por el desplazamiento en frecuencia de la onda recibida con respecto a la emitida. Muchos de los sensores láser (fundamentalmente los 3D, modulados en amplitud), proporcionar un mapa de distancias al entorno,

aportan información adicional sobre las características (i.e., rugosidad, textura, color) de las superficies interceptadas.

Las principales desventajas que presentan estos sensores están relacionadas con las partes mecánicas que incorporan (i.e. vibraciones y posicionamiento de espejos) y con su alto costo. A pesar de ello, los escáneres láser son los sensores que ofrecen más ventajas para aplicaciones en robots móviles (por ejemplo, la precisión, la fiabilidad o el tiempo de procesamiento).

b) Los sensores pasivos captan la energía ya existente en el medio. Para aplicaciones de robots móviles es necesario tener en cuenta algunas características del sensor, tales como la inmunidad a las variaciones de las condiciones ambientales, la robustez ante vibraciones y otros efectos del medio y del vehículo, el tamaño, el consumo de energía, el desgaste, la seguridad de su funcionamiento, la resolución, la precisión o el alcance.

Las cámaras de tipo "*pinhole*" (palabra inglesa comúnmente utilizada en la visión computacional para referirse a un modelo de cámara específico) y las cámaras omnidireccionales son de este tipo de sensores. Ambas perciben el entorno a través de la cantidad de luz que les llega, procedente directamente de fuentes luminosas o a través de reflexiones en los objetos del entorno:

- Las cámaras de tipo "*pinhole*" son utilizadas en percepción debido a las ventajas que presentan, como la velocidad en la adquisición de la información, el bajo consumo de energía, o su bajo costo. Sin embargo, la fuerte dependencia de las condiciones ambientales de iluminación, así como la excesiva complejidad computacional requerida a la hora de obtener información tridimensional, hacen difícil su utilización en muchas de las aplicaciones de robots móviles (en especial en escenarios de exteriores).

  La estimación del movimiento propio es difícil de calcular con estas cámaras [Vassallo, R. et al., 2002], aún cuando la información acerca del movimiento del vehículo se presenta en las imágenes, ya que se vuelve muy sensible al ruido y a la orientación de la cámara. Esto se debe a que pueden ser modeladas como la proyección en perspectiva plana (es decir, la proyección en perspectiva de la estructura 3D en un plano de la imagen). Es muy común que la dirección de translación quede fuera de su limitado campo de visión, lo cual es un reto para muchos métodos. Adicionalmente, los diferentes movimientos de la cámara pueden producir campos de movimientos similares en la imagen [Daniilidis, K. and Nagel, H., 1993]. Un ejemplo es el caso de una cámara mirando sobre el eje z, donde la rotación sobre el eje x produce un efecto similar a una translación en la dirección del eje y.

- Las cámaras omnidireccionales tienen un campo de visión alargado que simula la retina del observador, la cual es esférica. Con un campo de vista esférico, tanto el Foco de Expansión ("*Focus Of Expansion*", FOE) como el Foco de Contracción ("*Focus Of Contraction"*, FOC) son visibles en la imagen, lo que mejora la estimación del movimiento propio, haciéndolo más sencillo de calcular [Vassallo, R. et al., 2002].

En [Baker, S. and Nayar, S., 1998] y [Baker, S. and Nayar, S., 1999] se han desarrollado diferentes sistemas omnidireccionales y, gracias a que el campo de visión alargado mejora significativamente la robustez del seguimiento o localización de un robot, han sido usadas en muchas aplicaciones de navegación de robots autónomos ([Winters, N. et al., 2000], [Matsumoto, Y. et al., 2000]), incluyendo la estimación del movimiento propio [Lee, J.W. et al., 2000].

1.2.2 Extracción de información del entorno

En esta tesis se utiliza una cámara de tipo "*pinhole*" para captar el entorno, debido, principalmente, a que son más fáciles de adquirir y son más económicas que las omnidireccionales. De las imágenes (que se obtienen con ella) se extraen puntos sobresalientes, que son los puntos de la imagen caracterizados por ser distintivos, repetibles y localizados. Éstos se llaman puntos de interés [Pratt, W., 2001] y se utilizan como base para muchos algoritmos en visión computacional. Por esta razón, una propiedad deseable en los métodos que las detectan (llamados detectores de puntos de interés), es que exista un alto porcentaje de puntos que estén simultáneamente presentes en dos imágenes. Ya que, si esto sucede, mayor cantidad de puntos pueden ser potencialmente correlacionados y, por lo tanto, se tendrán mejores resultados en las correspondencias.

La detección de puntos de interés es una operación de procesamiento de imágenes de bajo nivel. Es, usualmente, desarrollada como la primera operación en la imagen, en la que se examina cada píxel para ver si hay una característica presente.

Una vez que los elementos de interés son encontrados, se tiene que codificar su apariencia local, de tal forma que permita la búsqueda de elementos similares. Esta codificación se conoce como descriptor ó vector de características.

Existe un gran número de posibles descriptores que enfatizan diferentes propiedades en la imagen, como las intensidades de los píxeles, el color, la textura o los contornos, por mencionar algunos. A continuación se muestran algunos descriptores basados en imágenes con valores de gris, que han sido utilizados, satisfactoriamente, en aplicaciones de correspondencias y reconocimiento de una misma escena u objetos observados bajo diferentes puntos de vista:

a) Descriptores basados en la distribución. Utilizan histogramas para representar diferentes características de apariencia o forma. Un descriptor simple es la distribución de las intensidades de los píxeles representados con un histograma, pero existen otro tipo de descriptores más complejos como los siguientes:

- En [Johnson, A. and Hebert, M., 1997] desarrollaron una representación de superficies, en donde la forma de éstas se describen como una colección densa de puntos orientados. A partir de un solo punto base, construido a partir de un punto orientado, la posición de otros puntos en la superficie puede ser descrita por dos parámetros. La acumulación de estos parámetros, para muchos puntos en la superficie, resultan en una imagen en cada punto orientado. A través de la correlación de imágenes, se establece la correspondencia de puntos entre superficies. Juntando los puntos orientados y las imágenes asociadas, se genera la representación de la superficie (llamada "*spin-image*").

- En [Zabih, R. and Woodfill, J., 1994] desarrollaron una aproximación robusta a cambios de iluminación. Aplican un histograma de relaciones recíprocas y ordenadas entre intensidades de píxel. Las relaciones binarias entre intensidades de varios píxeles vecinos son codificadas por cadenas binarias y una distribución de todas las posibles combinaciones es representada por histogramas. Este descriptor es útil para representar texturas, pero se requiere un gran número de dimensiones para construir un descriptor confiable.

- En [Lowe, D., 2004] desarrollan un descriptor basado en histogramas de orientación del gradiente, llamado Transformación de Características Invariante a Escala (*"Scale-Invariant Feature Transform"*, SIFT), el cual es un algoritmo de visión computacional para extraer características distintivas de la imagen. Se utiliza en algoritmos que realizan tareas como la correspondencia entre diferentes vistas de un objeto en una escena, el reconocimiento de objetos, la correspondencia estéreo, por mencionar algunas. Las características son invariantes a la rotación y a la escala de la imagen y parcialmente invariantes al cambio de iluminación y al punto de vista de la cámara 3D (ver cuadro 1.1).

- En [Mikolajczyk, K. and Schmid, C., 2005] implementaron una versión de SIFT, para hacerlo más robusto y distintivo, llamada Histograma de Gradiente de Orientación-Localización (*"Gradient location and Orientation Histogram"*, GLOH).

- El Histograma Geométrico y el Contexto de la Forma [Mikolajczyk, K. and Schmid, C., 2005] implementan la misma idea y son muy similares al descriptor SIFT. Ambos métodos calculan el histograma describiendo la distribución de los contornos en una región.

b) Entre las técnicas que describen el contenido de la frecuencia de una imagen, se pueden mencionar las siguientes:

- La transformada de Fourier. Descompone el contenido de una imagen en las funciones base (cosenos). Sin embargo, en esta representación, las relaciones espaciales entre puntos no son explícitas y las funciones base son infinitas, por lo que es difícil de adaptar a una aproximación local.

- Las transformadas de ondoletas. Dividen una función dada en diferentes componentes de frecuencia y estudian cada componente con una resolución que corresponde a su escala. Está localizada en espacio y frecuencia, mientras que la transformada de Fourier estándar solo se localiza en frecuencia. Otra ventaja es que las transformadas de ondoletas discretas son computacionalmente menos complejas que la transformada de Fourier rápida.

c) En los descriptores diferenciales, un conjunto de derivadas de una imagen, calculadas dado un orden, aproximan a un punto del vecindario. Se pueden mencionar las siguientes:

- En [Florack, L. et al., 1991] derivaron las diferenciales invariantes, las cuales combinan los componentes de las derivadas locales para obtener invarianza en la rotación.

- En muchas tareas de visión y procesamiento de imágenes se requiere usar un mismo filtro orientado, rotado a diferentes ángulos, bajo un control adaptativo, o se requiere usar la respuesta del filtro en varias orientaciones. En [Baird, H.S., 1985] se presenta una arquitectura para sintetizar filtros de orientaciones arbitrarias, a partir de combinaciones lineales de filtros base, lo cual permite que se pueda dirigir la orientación del filtro, usando, por ejemplo, la segunda derivada Gaussiana, y determinar analíticamente la salida del filtro como una función de orientación.

Para extraer las características de las imágenes de los algoritmos, que serán presentados en los siguientes capítulos, se seleccionaron dos tipos de extractores; para probar que tan sensibles son al ruido, cuantas correspondencias iniciales soportan, etc.:

a) Extractor basado en gradiente. Primero se hace una convolución entre la imagen (en escala de grises) y un filtro Prewitt (para detectar los bordes). Después, cada píxel de la imagen resultante se compara con sus vecinos de un radio determinado; para encontrar aquellos que tengan el cambio de intensidad luminosa más sobresaliente.

b) Descriptor SIFT. La elección de este detector se basa, por un lado, en los resultados presentados en el artículo [Mikolajczyk, K. and Schmid, C., 2005], en el que se hizo una evaluación experimental de algunos de los descriptores de regiones mencionados anteriormente. En la mayoría de los casos, GLOH obtuvo el mejor resultado, fuertemente seguido de SIFT. Sin embargo, se utiliza este último debido a que el ejecutable está disponible para su uso. Por otro lado, SIFT ya fue aplicado en un proyecto realizado en [Se, S. et al., 2002] para la localización de un robot.

El extractor basado en gradiente no es muy bueno (y menos comparado con SIFT), pero ese es el objetivo. De esta manera, los algoritmos se podrán probar con distintas entradas.

Cuadro 1.1 Algoritmo de SIFT

Los principales pasos para generar el conjunto de características de la imagen con SIFT son:

1. Detección extrema del espacio-escala. En el primer paso, se hace una búsqueda sobre todas las escalas y localizaciones de la imagen; para identificar cuales puntos de interés son repetibles bajo diferentes vistas del mismo objeto. Esto se logra mediante el uso de una función continua de escala conocida como espacio-escala, $L(x, y, \sigma)$, que es producida por la convolución de una Gaussiana de escala variable, $G(x, y, \sigma)$, con una imagen de entrada, $I(x, y)$:

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y) \tag{1.1}$$

donde $*$ es la operación de convolución en x y y , y

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2} \tag{1.2}$$

Para detectar eficientemente la localización de puntos de interés estables en el espacio escala, se utiliza una función espacio-escala extrema, formada por la convolución de una diferencia de Gaussianas con la imagen, $D(x, y, \sigma)$. Dicha función se calcula con la diferencia de dos escalas cercanas separadas por un factor constante multiplicativo, k :

$$\begin{aligned} D(x, y, \sigma) &= \big(G(x, y, k\sigma) - G(x, y, \sigma) \big) * I(x, y) \\ &= L(x, y, k\sigma) - L(x, y, \sigma) \end{aligned} \tag{1.3}$$

Para detectar el máximo y mínimo local de dicha función, cada punto muestreado se compara con sus ocho vecinos en la imagen actual y con sus nueve vecinos de la escala superior e inferior.

2. Localización del punto de interés. En el segundo paso, en cada ubicación candidata se ajusta un modelo detallado para determinar su ubicación y escala. Los puntos de interés son seleccionados con base en mediciones de su estabilidad.

3. Asignación de orientación. En el tercer paso, se busca obtener una orientación consistente, para cada punto de interés, basada en las propiedades locales de la imagen. El descriptor del punto de interés puede ser representado con relación a su orientación y, por lo tanto, puede ser invariante a la rotación de la imagen. Para cada imagen muestreada, $L(x, y)$, la magnitud del gradiente $m(x, y)$, y la orientación, $\theta(x, y)$, se precalculan utilizando diferencias de píxel:

$$m(x,y) = \sqrt{\left(L\left(x+1,y\right)-L\left(x-1,y\right)\right)^2+\left(L\left(x,y+1\right)-L\left(x,y-1\right)\right)^2} \qquad (1.4)$$

$$\theta(x,y) = \tan^{-1}\left(\left(L\left(x,y+1\right)-L\left(x,y-1\right)\right)/\left(L(x+1,y)-L(x-1,y)\right)\right) \qquad (1.5)$$

Se forma un histograma de orientación con las orientaciones de gradiente de los puntos muestreados alrededor del punto de interés. Los picos en el histograma de orientación corresponden a direcciones dominantes de gradientes locales. Se detecta el pico más alto en el histograma, y luego, cualquier otro pico local (es decir, el pico que esté dentro del 80% del pico más alto) también es usado para crear un punto de interés con esa orientación. Por lo tanto, para ubicaciones con picos múltiples de magnitudes similares, habrá múltiples puntos de interés creados con la misma ubicación y escala pero con diferentes orientaciones.

4. Descriptor del punto de interés. El último paso es calcular un descriptor para la región local de la imagen, que sea altamente distintivo e invariante a cambios de iluminación y puntos de vista 3D. Para lo cual, se muestrean las magnitudes y las orientaciones del gradiente de la imagen (ver figura 1.2.a) alrededor de la ubicación del punto de interés. Se usa la escala del punto de interés para seleccionar el nivel de suavizado Gaussiano adecuado a la imagen. Con el objetivo de obtener una invarianza en la orientación, se rotan las coordenadas del descriptor y las orientaciones del gradiente, con relación a la orientación del punto de interés. Se utiliza una función de peso Gaussiana (con σ igual a la mitad del ancho de la ventana del descriptor), para asignarle un peso a la magnitud de cada punto muestreado. El propósito de esta ventana Gaussiana (ilustrada con una ventana circular en la figura 1.2.a) es, tanto evitar cambios repentinos en el descriptor, con pequeños cambios en la posición de la ventana, como darle menos énfasis a los gradientes que están lejos del centro del descriptor.

El descriptor del punto de interés permite crear histogramas de orientación sobre regiones de muestreo de 4×4, para cambios significativos en las posiciones del gradiente. La figura 1.2.b muestra un arreglo de histogramas de orientación de 2×2, con ocho direcciones cada uno. La longitud de cada flecha corresponde a la magnitud de ese histograma de entrada. Con esto se logra el objetivo de permitir cambios de posición local más largos.

El descriptor está formado a partir de un vector que contiene los valores de todas las entradas de los histogramas de orientación, correspondientes a las longitudes de las flechas de la figura 1.2.b.

Los mejores resultados se obtienen con un arreglo de histogramas de orientación de 4×4, con 8 cuadros de orientaciones cada uno. Por lo que, cada punto de interés está formado por un vector de características de $4\times4\times8=128$ elementos.

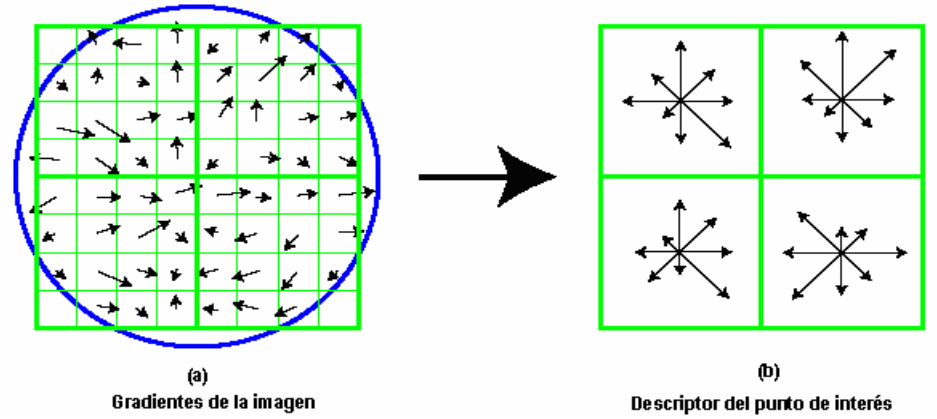


Figura 1.2 Cálculo del descriptor del punto de interés. Imagen tomada de [Lowe, D., 2004].

El vector de características se normaliza, para reducir los efectos de cambios de iluminación. Tal que, un cambio en el contraste de la imagen (en el que cada valor de los píxeles es multiplicado por una constante) será cancelado por el vector de normalización, y, un cambio en el brillo (en el que una constante es agregada a cada píxel de la imagen) no afectará los valores del gradiente.

1.2.3 Correspondencia de datos

Una vez que se localizan los puntos de interés de distintas imágenes (por ejemplo, de una misma escena, o de una escena y de un modelo o mapa conocido a priori), se debe hacer una correspondencia entre ellos.

El problema de la correspondencia es de gran interés en el campo del análisis de las imágenes, de la visualización y del proceso digital de imágenes, por mencionar algunos. Los puntos de interés correspondientes entre imágenes constituyen un paso fundamental en muchas aplicaciones de visión computacional, como recobrar escenas 3D, detectar objetos móviles, o sintetizar las vistas de una cámara. Para resolver este problema, se pueden seguir dos estrategias alternativas: correspondencia icónica y correspondencia basada en la extracción de características.

En la primera, el procesamiento y las operaciones son a nivel de píxeles ó elementos de la imagen. Se trabaja directamente con los dos conjuntos de datos, buscando la correspondencia directa de sus elementos. Es menos popular que la correspondencia basada en características, sin embargo, esta técnica ha sido empleada con éxito en aplicaciones diversas como la estimación de profundidad ([Lucas, B., 1985], [Matthies, L. et al., 1989]) o la correspondencia entre mapas ([Elfes, A., 1990], [Gonzalez, J., 1996], [Szeliski, R., 1988]). Una variante normalmente considerada como icónica es aquella que hace corresponder los datos directamente proporcionados por el sensor con un modelo o mapa formado por características ([Cox, I., 1991], [González, J. et al., 1995]).

Por el contrario, en la extracción de características se extraen un conjunto de características de cada uno de los dos conjuntos de interés, para luego buscar los pares de correspondencia entre elementos de ambos conjuntos de características. Este procedimiento corresponde al enfoque típicamente empleado en los problemas de reconocimiento de objetos, aunque también ha sido ampliamente utilizado en el campo de los robots móviles para la construcción de mapas y la estimación de la posición.

Ambos enfoques del problema presentan ventajas y desventajas [Shaffer, G. et al., 1992]. La correspondencia basada en extracción de características requiere de una cierta estructuración subyacente en los datos, que posibilite la identificación de objetos o formas determinadas (i.e., segmentos, círculos, polígonos). Además, en determinados casos, el proceso de extracción puede llegar a ser bastante costoso desde el punto de vista computacional. El número de combinaciones a explorar en esta metodología es considerablemente menor que en la metodología icónica, si bien, en este caso, la máxima complejidad es conocida de antemano (en principio todos con todos).

En este trabajo, se decidió utilizar la correspondencia basada en la extracción de características, debido a que son robustas a cambios de iluminación, puntos de vista y oclusiones.

Entre las aplicaciones que utilizan la extracción de características se pueden mencionar las siguientes:

- El prototipo de un sistema de visión estéreo portable [Stein, F. and Medioni, G., 1995], el cual utiliza dos cámaras que se conectan a un puerto USB ("*Universal Serial Bus*"), a un sistema de retroalimentación táctil y a una computadora portable. Se aplica un método de correspondencia píxel a píxel utilizando la restricción de la geometría epipolar. Trabaja en tiempo real, ya que limita el campo de visión sólo lo necesario para la navegación.

- La etapa de estimación del movimiento propio desarrollado en [Saez, J.M. et al., 2005] utiliza una extracción de características sencilla. Para obtener las correspondencias entre ellas, se miden las similitudes entre las apariencias locales en el vecindario de sus proyecciones respectivas. Con el objetivo de asegurar la calidad de la correspondencia, se rechazan aquellas con baja similitud, bajos distintivos y unidireccionalmente, después de calcular las mejores asociaciones para todos los puntos, se procede a identificar y remover potenciales oclusiones. Finalmente se lleva a cabo la estimación de la transformación.

- El sistema de visión DROID [Harris, C. et al., 1992], el cual utiliza filtros de Kalman para el seguimiento de características y, a partir de esto, se determina tanto el movimiento de la cámara como la posición 3D de las características.

Los algoritmos para obtener las correspondencias entre puntos de interés tienen el objetivo de reducir el número de correspondencias incorrectas, mientras se preservan las correctas. La presencia de correspondencias incorrectas significa que los puntos de interés resultantes no pueden hacer correspondencias exactas; o que los puntos de interés en un conjunto de puntos, no tienen correspondencia con los puntos del otro grupo.

Sin embargo, debido a que una búsqueda, que intente hacer corresponder todos los posibles puntos, lleva a una explosión combinatoria, se han propuesto algunos métodos para poder reducir el espacio de búsqueda. Existen tres tipos principales:

a) Métodos basados en características densas. En [Metaxas, D. et al., 1997], [Sclaroff, S. and Pentl, A., 1995], [Tagare, H.D. et al., 1995], [Szeliski, R. and Lavallée, S., 1996] y [Feldmar, J. and Ayache, N., 1996], las características extraídas de las imágenes son ajustadas a curvas, líneas, o superficies, con lo que el espacio de correspondencias puede ser drásticamente reducido, haciendo el problema de correspondencia mucho más fácil. El problema con estos métodos es que solo trabajan bien cuando las curvas que hay que corresponder son razonablemente suaves. Además, se necesita una buena extracción de características para que funcione bien. La mayoría de estos métodos no pueden manejar múltiples curvas o curvas con oclusiones parciales.

b) Métodos que trabajan con puntos más dispersos.

- Se asignan atributos de movimiento a un conjunto de puntos, de acuerdo a las transformaciones no rígidas que sufran. Dichos atributos son usados para distinguir los puntos y determinar sus correspondencias ([Scott, G. and Longuet-Higgins, H.C., 1991], [Shapiro, L.S. and Brady, J.M., 1992] y [Cootes, T.F. et al., 1995]).

- La aproximación de correspondencia modal descrita en [Sclaroff, S. and Pentl, A., 1995], utiliza una matriz, que es construida a partir de distancias Gaussianas entre los puntos de interés de un conjunto de puntos. Se obtienen vectores propios a partir de las llamadas ecuaciones desemparejadas de equilibrio dinámico, las cuales son definidas usando las matrices, mencionadas anteriormente. La correspondencia es calculada comparando cada participación relativa del punto, en los vectores. Una limitación importante de estos algoritmos es que no toleran correspondencias incorrectas. Ya que pueden deformar los modos, invalidando las correspondencias resultantes.

c) Métodos de correspondencias de grafos con pesos.

- En [Shapiro, L.G. and Haralick, R.M., 1981] utilizan las relaciones espaciales entre los puntos de cada conjunto para limitar las búsquedas de las correspondencias. Esta interrelación entre los puntos se toma en cuenta en [Cross, A.D.J. and Hancock, E.R., 1998] para construir una representación gráfica de la triangulación Delaunay. Utilizan un algoritmo llamado "*Expectation – Maximization"* (EM) para solucionar el problema de la optimización de la correspondencia de grafos. Sin embargo, los mapeos espaciales son restringidos a ser afines o proyectivos.

- En [Lohmann, G. and von Cramon, D.Y., 2000] presentan un algoritmo basado en grafos para realizar la correspondencia estructural de características locales. El algoritmo consiste en la transformación simultánea de dos grafos, uno para los puntos característicos de la imagen del modelo y otro para los de la escena.

- Los sistemas propuestos en [Lebegue, X. and Aggarwal, J., 1993] y [Tardos, J., 1990], extraen de una imagen segmentos característicos en las tres direcciones del espacio. Un procedimiento alternativo consiste en extraer de la imagen aristas verticales para luego establecer la correspondencia entre las direcciones de las aristas observadas y las esquinas que aparecen en el mapa bidimensional del entorno [Krotkov, E., 1989], [Muñoz, A. and González, J., 1996], [Sugihara, K., 1988]. El problema se formula como una búsqueda en un árbol de interpretaciones, donde cada nodo es una posible interpretación de las aristas observadas, es decir, cada nodo representa un conjunto de posibles correspondencias para las direcciones observadas.

Los resultados obtenidos con estas técnicas dependen de la calidad de la imagen, del número de aristas detectadas, de la exactitud del mapa utilizado y, sobre todo, del tipo de escenario. El rango de aplicación de estos estimadores queda limitado a entornos fuertemente estructurados. Sin embargo, también se han propuesto estimadores para robots que navegan en escenarios abiertos. La idea consiste en posicionar una o varias cámaras de manera que puedan ver el horizonte. A partir de la forma y posición del perfil de éste dentro de la imagen, y suponiendo que se conoce el mapa topográfico de la zona (por ejemplo mediante curvas de nivel), puede estimarse la localización del vehículo estableciendo la correspondencia entre ambos ([Stein, F. and Medioni, G., 1995], [Sutherland, K. and Thompson, W., 1994], [Szeliski, R., 1988]).

Por otro lado, en el caso, de secuencias de imágenes tomadas con cámaras monoculares, donde no se tiene la información 3D (que se obtiene de las cámaras estéreo), el movimiento relativo entre píxeles en imágenes consecutivas, está sometido a transformaciones rígidas y no rígidas. Esto también ocurre en muchas otras aplicaciones del mundo real. Tales transformaciones son difíciles de obtener, debido al ruido producido por el proceso de adquisición de las imágenes y a la extracción de las características, por lo que se han propuesto algoritmos para lograrlo. Entre los que se pueden mencionar los siguientes:

- En [Stockman, G., 1987] plantean la transformada de Hough, donde la transformación del parámetro espacial es dividido en pequeñas secciones, y cada sección representa una cierta configuración de los parámetros de la transformación. Los puntos votan por todas las secciones y la que tenga más votos es elegida.

- En [Huttenlocher, D.P. et al., 1993] presentan algoritmos para calcular la distancia Hausdorff, entre todas las posibles posiciones relativas de un modelo y una imagen. Se enfocan en los casos en los que se permite que el modelo y la imagen se trasladen con respecto una de la otra, también consideran el caso más general de movimiento rígido.

- En [Ullman, S., 1989] se utiliza el método de alineación, el cual está dividido en dos partes. La primera determina la transformación rígida en el espacio, que es necesaria para alinear el objeto visto, con posibles modelos de objetos. Este paso puede generarse con mínima información, como la orientación dominante del objeto, o un pequeño número de puntos característicos correspondientes entre el objeto y el modelo. El segundo paso, determina el modelo que mejor corresponda con el objeto visto. En esta etapa, la búsqueda es sobre todos los posibles modelos de objetos, no sobre todas sus posibles vistas, dado que la transformación ha sido únicamente determinada en el paso de alineación. Este método también propone descripciones abstractas, pero a diferencia de los métodos con descripción estructural, utiliza descripciones pictóricas, en lugar de descripciones estructurales simbólicas.

Sin embargo, hay casos en los que se requiere que se obtenga tanto la correspondencia como la transformación entre los puntos. Por ejemplo: las correspondencias de patrones con caracteres escritos a mano, la generación de interpolaciones suaves entre cuadros intermedios y cuadros clave en animaciones de cartones, el seguimiento del movimiento del cuerpo humano, el recobrar el movimiento dinámico del corazón en los análisis de imágenes cardiacas, por mencionar algunas. Todas estas actividades involucran encontrar la transformación óptima entre formas u objetos cercanamente relacionados.

Este es un problema difícil, dado que la apariencia de la proyección de las superficies en las imágenes varía dependiendo de varios fenómenos, como la posición de la cámara, la deformación de las superficies o la iluminación. Para solucionarlo se han propuesto muchos métodos, sin embargo, debido a lo complejo que es, no lo atacan por completo, solo lo simplifican. Entre los que se encuentran:

- El algoritmo llamado "*Iterative Closest Point"* (ICP) se utiliza en [Besl, P.J. and McKay, N.D., 1992] para encontrar el punto más cercano de una entidad geométrica a un determinado punto. Dado un conjunto inicial de rotaciones y translaciones, para una clase particular de objetos (con un cierto nivel de complejidad de la forma), se puede minimizar globalmente la métrica de la distancia de mínimos cuadrados sobre los seis grados de libertad, probando cada correspondencia inicial. Por ejemplo, teniendo un modelo de una forma y datos censados, que representen la mayor porción del modelo de la misma, éstos pueden ser correspondientes probando una traslación inicial y un conjunto relativamente pequeño de rotaciones. Este método se puede utilizar para encontrar la equivalencia entre las formas de diferentes representaciones geométricas, así como para estimar el movimiento entre conjuntos de puntos, donde las correspondencias son desconocidas.

- En [Cross, A.D.J. and Hancock, E.R., 1998] unifican las tareas de estimar las transformaciones geométricas e identificar los puntos correspondientes. Para lo cual construyen un modelo mixto sobre una gráfica bipartita, que representa la correspondencia de los puntos. Utilizan un algoritmo EM para optimizar. Las probabilidades de la correspondencia estructural contribuyen a la función de probabilidad esperada, usada para estimar los parámetros de transformación con máxima probabilidad. Estas probabilidades miden la consistencia de la correspondencia de los vecinos en un grafo.

- En [Wells, W., 1997] se utilizan un par de aproximaciones estadísticas para el reconocimiento de objetos basado en modelos con distribuciones normales de características. En la primera, la búsqueda de soluciones puede ser ordenada como una búsqueda combinatoria en el espacio de correspondencias, o como una búsqueda sobre el espacio de posiciones. En la segunda, se tiene una función suavizada de la posición que es usada para la búsqueda. Utilizan EM para la optimización, el cual alterna entre el refinamiento de la posición y la reestimación de las probabilidades de las correspondencias, hasta converger.

- En [Hinton, G.E. et al., 1992], dígitos escritos a mano son modelados como "*splines"*[1], las cuales son funciones polinomiales que pueden tener una forma localmente muy simple, pero al mismo tiempo son globalmente muy flexibles y suaves (ver apéndice 1). Las imágenes reales pueden ser reconocidas encontrando el modelo del dígito, que más se parezca a los datos generados con las "*splines"*. Para cada modelo del dígito, se utiliza un algoritmo de correspondencia elástico para minimizar una función de energía, que incluye tanto la energía de la deformación del modelo del dígito, como la probabilidad de que el modelo pueda generar unos puntos de control en la imagen. El modelo con la menor energía total, gana.

- El algoritmo mostrado en [Chui, H. and Rangarajan, A., 2000] explota en puntos lo que en [Saez, J.M. et al., 2005] se explota para grafos. Es un algoritmo tipo EM que calcula la correspondencia y la transformación no rígida (descompuesta en parte rígida y parte no rígida) siguiendo el método de los "*Thin-Plate splines"* (TPS, ver capítulo 2).

- En [Gold, S. et al., 1995], el problema de correspondencias de puntos se modela como un problema de optimización de una conjunción de mínimos cuadrados y una asignación lineal, para resolverlo se crearon nuevos algoritmos llamados "*Deterministic Annealing*" y "*Softassign*". En estos algoritmos se formulan funciones de energía, caracterizadas por el uso de una matriz de correspondencia que denota explícitamente la asignación entre un conjunto de puntos y otro. La matriz de correspondencia es una matriz de ceros y unos, donde los unos indican que un punto en un conjunto está asociado a un punto en el otro conjunto. Las variables de la matriz de correspondencia deben satisfacer las restricciones de la matriz de asignación (i.e. las filas y las columnas deben sumar a lo más, uno y las entradas deben ser cero o uno). Para ello, a cada par de puntos se les asigna una función de probabilidad de correspondencia basada en una función Gaussiana de la distancia entre ellos "*Softassign*" y, el "*Deterministic Annealing*" controla lo difuso de lo generado en el "*Softassign*", agregando un término de entropía.

- En [Chui, H. and Rangarajan, A., 2003] se plantea un algoritmo para resolver las correspondencias no rígidas entre puntos, llamado TPS-RPM ("*Robust Point Matching*", RPM), que utiliza, por un lado, el TPS para la parametrización del mapeo espacial no rígido y, por otro, el Asignado Suave para las correspondencias. Es relativamente robusto cuando se considera un número reducido de correspondencias incorrectas. La robustez también ha sido considerada en [Rohr, K. et al., 2001]. Una aproximación (y más eficiente estimación) del TPS es discutido en [Donato, G. and Belongie, S., 2002]. También se ha aplicado en combinación con el ICP [Brown, B. and Rusinkiewicz, S., 2004] para la alineación no rígida de acoplamientos 3D.

[1] La palabra inglesa *"spline*" es comúnmente utilizada en varias áreas, entre ellas, la visión computacional.

En esta tesis, para resolver el problema de la correspondencia y la transformación, se proponen algunos algoritmos (basados en TPS y Geometría Epipolar, ver capítulo 2) que se detallan en el capítulo 3.

1.2.4 Estimación del movimiento

El problema de la detección, descripción y entendimiento del movimiento, a partir de datos visuales, son de los problemas más difíciles en visión computacional. Esto se debe a que existen dos tareas principales que deben ser resueltas: i) La estimación del movimiento del sensor visual (llamado movimiento propio), con respecto a alguna referencia estática y al movimiento asociado con los objetos observados; y ii) la discriminación de la segmentación del movimiento, donde, los campos de movimiento 2D, previamente derivados, deben ser interpretados en términos de objetos rígidos, articulados o deformables, discriminando entre objetos que tienen distintos movimientos.

Las técnicas propuestas para resolver estas tareas, caen en dos categorías: análisis de una secuencia de imágenes y análisis del movimiento estéreo. Solo serán mencionadas las primeras, debido a que el problema que se pretende resolver, es la estimación del movimiento a partir de una secuencia de imágenes obtenidas con una cámara monocular.

De acuerdo con [Armangue, X. et al., 2003], las aproximaciones a la estimación del movimiento a partir de una secuencia de imágenes, pueden ser clasificadas en métodos discretos y métodos diferenciales, dependiendo de si se utiliza un conjunto de puntos correspondientes o el flujo óptico.

Cuando se comparan los métodos diferenciales con los discretos, la ecuación epipolar discreta incorpora una sola matriz, mientras que la ecuación epipolar diferencial incorpora dos matrices. Estas matrices codifican la información acerca de las velocidades lineales y angulares de la cámara.

En [Tian, T. et al., 1996] y [Haralick, R., 1988] hacen un comparativo de algunos algoritmos diferenciales para calcular la estimación del movimiento. Aquí solo serán mencionados algunos:

- En [Bruss, A. and Horn, B.K.P., 1983] aplicaron una manipulación algebraica simple para eliminar la profundidad y obtener restricciones bilineales, en la velocidad de traslación y la velocidad de rotación, por cada píxel de la imagen. En [MacLean, W. et al., 1994] derivaron las mismas restricciones bilineales (mencionadas en el punto anterior). Estimaron la translación minimizando la restricción no lineal sobre las velocidades de toda la imagen.

- En [Longuet- Higgins, H.C., 1984] propusieron un método basado en la paralaje del movimiento, en el que si dos puntos 3D tienen la misma ubicación en la imagen, pero tienen diferentes profundidades, entonces la diferencia de vectores estará orientada a través del FOE, el cual se localiza a partir de la diferencia de

vectores de flujo locales. El problema con este algoritmo es que es particularmente difícil medir los vectores de flujo cerca de los límites de la oclusión.

- En [Tomasi, C. and Shi, J., 1993] se desarrolló un método que usa información de la paralaje del movimiento en una forma distinta a las anteriores. Su método estima la translación a partir de deformaciones de la imagen, definida como el cambio en la distancia angular entre pares de puntos de la imagen mientras se mueve la cámara.

- En [Prazdny,K., 1980] se propone un algoritmo que en lugar de calcular primero la translación como los algoritmos anteriores, calcula primero la rotación. A partir de una tripleta de puntos de la imagen, la siguiente restricción en los parámetros de rotación (independientes de la translación y la profundidad) fue derivada por manipulación algebraica.

- En [Kanatani, K., 1993] se reformuló la restricción epipolar del tiempo instantáneo en términos de parámetros esenciales y flujo giratorio (una versión rotada del vector de velocidad). También se propuso otro algoritmo, en el que se encontró que la estimación de mínimos cuadrados de translación son sistemáticamente sesgados. Se analizó el sesgo estadístico, usando un modelo simple de ruido Gaussiano, después, se propuso un método (llamado método de renormalización) que elimina los sesgos compensando automáticamente para el ruido desconocido.

En [Torr, P. and Murray, D.W., 1997] se hace un comparativo de algunos algoritmos discretos para calcular la estimación del movimiento. Entre los que se pueden mencionar:

- El desarrollado en [Longuet-Higgins, H.C., 1981] donde presentaron un algoritmo para calcular la estructura y el movimiento a partir de un pequeño grupo de puntos correspondientes en dos imágenes. Plantearon una solución cerrada al problema usando ocho puntos 3D y dos cuadros, con base en la suposición de que el problema de correspondencia había sido resuelto.

- Otros han extendido la aproximación de los parámetros esenciales a líneas ([Faugeras, O., 1993], [Spetsakis, M. and Aloimonos, J., 1990]), desarrollado análisis más detallados del error [Weng, J. et al., 1992], y desarrollado técnicas de mínimos cuadrados no lineales para el problema de dos imágenes [Weng, J. et al., 1989].

- En [Saez, J.M. et al., 2005] utilizan un plano retinal estándar. El objetivo es desarrollar tanto la acción de estimación como el de correspondencia. Se tienen nubes de puntos 3D restringidas, y dado que cada punto en la primera nube corresponde a un punto en la segunda, la acción óptima es la que lleve a la transformación (rotación y traslación) que maximice el grado de alineación entre ambas nubes, esto es,

aquella que minimice una función de energía cuadrática. Para lo cual, se desarrolló un gradiente conjugado descendiente con un paso adaptativo a través de acciones incrementales.

- En [Fischler, M. and Bolles, R., 1981] proponen utilizar una retina esférica en lugar de un plano retinal estándar, ya que se dieron cuenta de que los parámetros del movimiento del observador son más fáciles de estimar. En trabajos previos a este se encontró como proyectar vectores de velocidad de la imagen en la superficie de una retina esférica (unitaria). La transformación entre los vectores de la imagen planar y los vectores esféricos se alcanzan con una matriz Jacobiana que es específica para el uso de sistemas omnidireccionales. Así, cada vez que un sistema de imágenes de ángulos amplios es usado, se debe derivar una nueva ecuación para la Jacobiana. Para superar esta limitación, en este artículo se propuso una expresión general para la Jacobiana que puede ser usada por muchas cámaras omnidireccionales diferentes, simplemente mandando dos parámetros. Para lograrlo se basaron en un modelo de formación de la imagen general, válido para todas las cámaras con un solo centro de proyección. También discuten el problema de escoger entre dos modelos de estimación de movimiento propio. Un modelo considera solamente movimiento rotacional, mientras que el segundo aplica simultáneamente traslación y rotación. El criterio de selección se basa en la desviación media y estándar de los residuos entre el campo del movimiento reconstruido y el observado. Obteniendo buenos resultados.

- En [Debrunner, C.H. and Ahuja, N.A., 1990] proveen expresiones de forma cerrada para formas y movimiento asumiendo que el movimiento es constante sobre la secuencia. Las soluciones incrementales para movimientos múltiples se calculan tomando ventaja de la redundancia de las mediciones.

- En [Cui, N. et al., 1990] utilizan una técnica de estimación óptima (mínimos cuadrados no lineales) entre cada par de imágenes, y un filtro de Kalman extendido para acumular información en el tiempo.

- En [Tomasi, C. and Kanade, T., 1990] utilizan un método de factorización, el cual extrae la forma y el movimiento de una secuencia de imágenes, sin calcular la profundidad del centro de la cámara. Sus aproximaciones formularon la forma a partir del problema de movimiento en coordenadas centradas del objeto, a diferencia de las formulaciones más convencionales centradas en la cámara.

Debido a que en este trabajo se utiliza un conjunto de puntos correspondientes, se aplicarán algoritmos discretos para resolver la estimación del movimiento.

Se pueden tener cuatro tipos de problemas, para estimar el movimiento, dependiendo de los datos iniciales:

1. Los conjuntos de datos consisten en puntos bidimensionales en un espacio bidimensional. Estos datos surgen cuando objetos planos en 3D, son vistos bajo la proyección en perspectiva, con el ángulo de vista siendo el mismo que la superficie normal del objeto visto.

2. Los conjuntos de datos consisten en un conjunto de puntos 3D en un espacio 3D. Este conjunto de datos surgen cuando los objetos en 3D, son vistos con un sensor buscador de rango.

3. Se tiene un conjunto de datos consistente en proyecciones en perspectiva 2D de puntos 3D, y otro conjunto de datos consistente en un conjunto de puntos 3D. Este caso se conoce como problema de orientación absoluta.

4. Se tiene un conjunto de puntos 3D, proyectados en perspectiva 2D, los cuales tienen un movimiento rígido (rotación y traslación) desconocido, reflejado en una segunda vista en proyección en perspectiva 2D, del mismo conjunto de puntos en 3D. Dicho de otro modo, se toman dos vistas del mismo objeto, que se pueden considerar como el movimiento rígido desconocido de un cuerpo de la primera vista a la segunda. Ambas vistas son proyecciones en perspectiva 2D. Este caso se conoce como problema de orientación relativa.

En [Haralick, R. et al., 1989] se muestran una serie de algoritmos para resolverlos. Sin embargo, debido a que en este trabajo se pretende resolver, solamente, el problema de orientación relativa, a continuación se muestra una clasificación de los algoritmos que resuelven este problema:

a) Algoritmos de la matriz esencial. Son algoritmos lineales que requieren por lo menos ocho puntos, aunque se han hecho extensiones de estos algoritmos cuando solo hay cinco o siete puntos disponibles.

b) Algoritmos directos. Estos algoritmos encuentran la rotación y la traslación directamente y requieren la minimización de funciones de costo no lineales.

c) Algoritmos de subespacio. Estos algoritmos son válidos para rotaciones pequeñas entre dos imágenes. Se requiere de la minimización de una función de costo bidimensional.

En el capítulo 4 se describirán, a detalle, los algoritmos implementados en este trabajo, que se ubican dentro de las dos primeras clasificaciones.

Capítulo 2

Fundamentos de la GE y del TPS

Como se mencionó en el capítulo anterior, en esta tesis se proponen algunos algoritmos (los cuales se explicarán en los capítulos 3 y 4) para resolver el problema de la correspondencia y la transformación entre los puntos extraídos de un par de imágenes. Dichos algoritmos están basados en los métodos de la Geometría Epipolar y el "*Thin-Plate spline"*, por lo cual, en este capítulo se explican a detalle.

2.1 Notación

A continuación se muestra la notación relevante para comprender las ecuaciones presentadas en esta tesis.

A	Vector.
n	Escalar.
$f()$	Función.
$\mathbf{M}$	Matriz.
$\mathbf{I}$	Matriz identidad.
$A^T B$	Producto interno entre dos vectores, también conocido como producto escalar o producto punto.
$A \times A$	Producto cruz entre dos vectores.
$\mathbf{M}^+$	Pseudoinversa de $\mathbf{M}$.
$[A]^\times$	Matriz antisimétrica de A, usada para calcular un producto cruz.
$\mathbf{0}$	Vector cero.
$\mathbf{M}^{-1}$	Matriz inversa de $\mathbf{M}$.
$\mathbf{M}^T$	Transpuesta de $\mathbf{M}$.

$\|A\|$	Norma del vector A.
$\det(\mathbf{M})$	Determinante de $\mathbf{M}$.
$\mathcal{R}$	Conjunto de puntos.

2.2 Geometría Epipolar

La Geometría Epipolar (GE) ha sido ampliamente utilizada en el campo de la visión computacional, debido, entre otras cosas, a que se pueden obtener correspondencias entre pares de imágenes y, a partir de ellas, se puede calcular la calibración de las cámaras, o viceversa.

Por ejemplo, en [Wexler, Y. et al., 2003] la GE se usa en la calibración automática de los sistemas de las cámaras estéreo. Esto se logra con un método que permite el aprendizaje de las formas de las curvas epipolares, a través de la adquisición de las correspondencias de múltiples pares de imágenes (vía cámaras estéreo) con configuración fija. Este método tiene dos beneficios secundarios: primero, permite la estimación de la geometría epipolar cuando el modelo paramétrico es desconocido (i.e. con material archivado). Segundo, aún cuando el modelo paramétrico sea conocido, el algoritmo no paramétrico puede ser un paso de preproceso valioso, ya que puede minimizar el costo computacional. Otro uso es en la estimación del movimiento de la cámara estéreo [Goshen, L. et al., 2003], aplicado al movimiento de un robot móvil [Armangue, X. et al., 2003] o la medición de la profundidad para sistemas de visión estereoscópica para invidentes [Audette, R. et al., 2000]. También ha sido la base para el cálculo de la geometría de tres vistas (llamada tensor trilineal), que es usada para describir la posición de la cámara a partir de una secuencia de imágenes, ya que produce un conjunto de correspondencias confiable y preciso ([Roth, G. and Whitehead, A., 2000] y [Kimura, M. and Saito, H., 2002]). Por otro lado, la GE es el modelo geométrico detrás de la reconstrucción de escenas del mundo real a través de imágenes, ya que, modela la posición relativa y la orientación del plano de la imagen. Para cada punto en una imagen, define una línea en la otra imagen, donde podría estar su punto correspondiente. Esto puede ser usado para restringir la búsqueda en la correspondencia densa [Megyesi, Z. and Chetverikov, D., 2003].

2.2.1 Geometría de la cámara tipo "pinhole"

La cámara tipo "*pinhole*" es un sistema compuesto por dos planos (ó pantallas). En el primero, llamado plano principal, existe un agujero pequeño (centro de proyección o centro óptico) a través del cual pasan algunos de los rayos de luz emitidos o reflejados por el objeto observado, que, al impactarse en la película, forman una imagen invertida del mismo en la segunda pantalla, también conocida como plano de la imagen o plano de la retina (π). Esta imagen se forma a partir de una operación llamada proyección en perspectiva (ver figura 2.1).

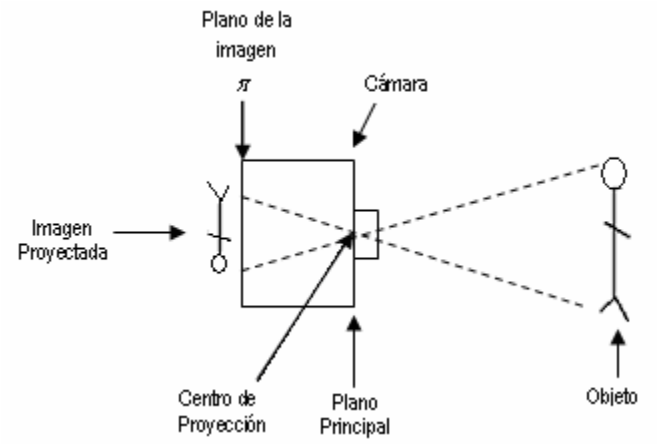


Figura 2.1 Cámara tipo "*pinhole*".

En este tipo de proyección, cuanto más lejos está un objeto del observador, en este caso la cámara, más reducida es su proyección. Esto permite al observador tener una idea de profundidad. Es una indicación de qué porciones de la imagen corresponden a las partes del objeto que están más cerca o más lejos del observador.

Si el centro de proyección está en el sistema de coordenadas estándar de la cámara, $O = (0,0,0)^T$, y un punto en el objeto está en $P = (x, y, z)^T$ (en coordenadas homogéneas $P = (x, y, z, 1)^T$), entonces, el rayo proyectante será la recta que pasa por estos dos puntos (ver figura 2.2).

El eje óptico se define como la línea que pasa por el centro de proyección y es perpendicular al plano de la imagen. Para simplificar, el sistema de coordenadas de la cámara coincide con el del mundo exterior. La distancia focal, *f*, mide la distancia desde el plano de la imagen al centro de proyección. El punto proyectado, tiene coordenadas homogéneas $\overline{P} = (\overline{x}, \overline{y}, 1)^T$, será el punto de intersección del rayo proyectante con el plano de la imagen (ver figura 2.3).

El eje de la coordenada *z* es la línea que pasa por el eje óptico. De lo anterior se pueden obtener las siguientes ecuaciones:

$$\overline{x} = \frac{-fx}{z} \tag{2.1}$$

$$\overline{y} = \frac{-fy}{z} \tag{2.2}$$

Para poder operar con álgebra matricial (y sea más sencillo de trabajar en computadora), se puede reformular la ecuación anterior en coordenadas homogéneas como:

$$z\begin{pmatrix}\bar{x}\\ \bar{y}\\ -f\end{pmatrix}=\begin{pmatrix}-fx\\ -fy\\ z\end{pmatrix}=\begin{pmatrix}-f & 0 & 0 & 0\\ 0 & -f & 0 & 0\\ 0 & 0 & 1 & 0\end{pmatrix}\begin{pmatrix}x\\ y\\ z\\ 1\end{pmatrix} \tag{2.3}$$

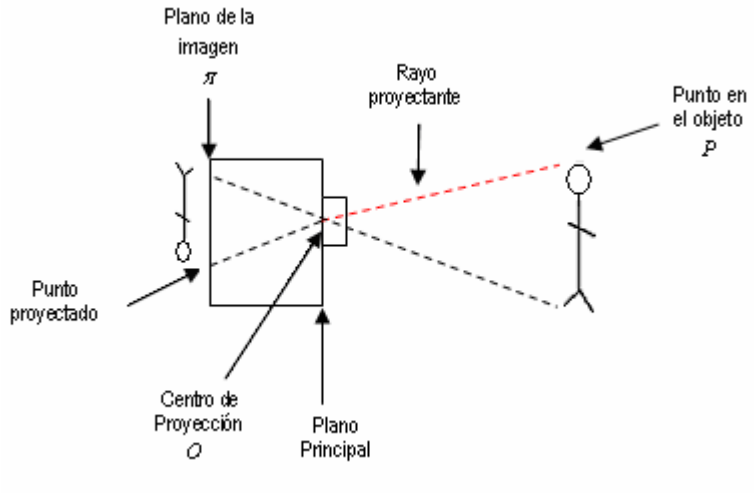


Figura 2.2 Rayo proyectante.

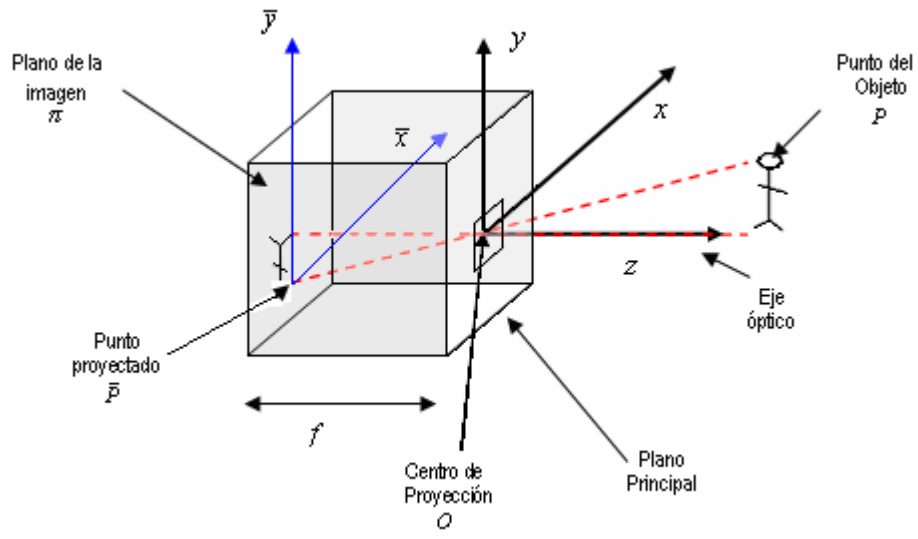


Figura 2.3 Geometría de la cámara tipo "*pinhole*".

Donde $\mathbf{M}$ es la matriz de proyección

$$\begin{pmatrix} -f & 0 & 0 & 0 \\ 0 & -f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \tag{2.4}$$

Así, la ecuación (2.3) queda como:

$$z\overline{P} = \mathbf{M}P \tag{2.5}$$

Sin embargo, como no afecta multiplicar un esclara por coordenadas homogéneas, la ecuación (2.5) queda como:

$$\overline{P} = \mathbf{M}P \tag{2.6}$$

La fórmula matemática de la matriz de proyección $\mathbf{M}$ es más complicada cuando los sistemas de coordenadas no coinciden, ya que es necesario calibrar los parámetros de cámara.

2.2.2 Análisis de la Geometría Epipolar

Supóngase que se tiene un punto P del espacio, que se proyecta en el punto $\overline{P}$ de la primera imagen, y el punto $\overline{P}'$ de la segunda imagen. Entonces, por aplicación directa de la ecuación (2.6) a este escenario, se tiene el sistema de ecuaciones matriciales:

$$\overline{P} = \mathbf{M}P \tag{2.7}$$

$$\overline{P}' = \mathbf{M}'P \tag{2.8}$$

Donde, $\mathbf{M}$ es la matriz de proyección de la primera imagen, $\mathbf{M}'$ es la matriz de proyección de la segunda imagen, P son las coordenadas homogéneas del punto en el espacio, $\overline{P}$ son las coordenadas homogéneas de la proyección del punto P en la primera imagen, $\overline{P}'$ son las coordenadas homogéneas de la proyección del punto P en la segunda imagen.

Se define la visión estéreo como aquella en la que se emplea más de una imagen para obtener una idea de tridimensionalidad. Según el número de imágenes que sean empleadas, se hablará de visión bifocal – dos imágenes –, trifocal – tres imágenes –, quadrifocal – cuatro imágenes – o N-focal – N imágenes –. En este caso, se empleará el sistema bifocal.

La geometría estéreo, denominada **geometría epipolar** (figura 2.4), es, esencialmente, la geometría de la intersección de los planos de la imagen, tomando como línea base el eje que une los centros de las cámaras. Se compone de dos cámaras cuyos centros de proyección son O y O', y sus planos de la imagen (π, π') están en coordenadas normalizadas. Las distancias focales se nombran f y f'. Cada cámara identifica un sistema de referencia 3D, el origen del cual se sitúa en el centro de proyección de la cámara, y el eje z con el eje óptico. Los vectores $P = (x, y, z)^T$ y $P' = (x', y', z')^T$ se refieren al mismo punto 3D, P, como vectores en los sistemas de referencia de la cámara izquierda y derecha respectivamente.

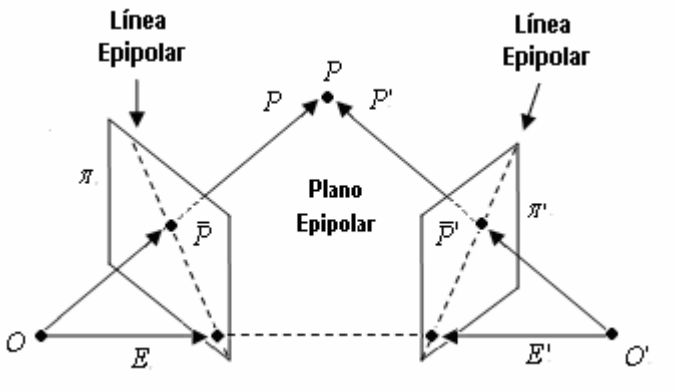


Figura 2.4 Geometría epipolar.

Los vectores $\bar{P} = (\bar{x}, \bar{y}, 1)^T$ y $\bar{P}' = (\bar{x}', \bar{y}', 1)^T$ definen las proyecciones del punto P en la imagen izquierda y derecha respectivamente y están expresados en su correspondiente sistema de referencia.

Los sistemas de referencia de las cámaras izquierda y derecha están relacionados a través de los **parámetros extrínsecos**. Estos definen una **transformación rígida**, en el espacio 3D, definida por un vector de traslación $T = O' - O$ y una matriz de rotación $\mathbf{R}$. Dado un punto P en el espacio, la relación entre P y P' es por tanto

$$P' = R(P - T) \qquad (2.9)$$

El nombre de geometría epipolar se debe a que los puntos, en los cuales la recta (que une los centros de proyección de las cámaras) corta a los planos de proyección, se llaman **epipolos**. La notación para estos puntos es E y E', el epipolo izquierdo y derecho respectivamente. Por construcción, ambos epipolos representan la proyección, en su correspondiente plano de la imagen del centro de proyección de la otra cámara. En el caso de que uno de los planos de la imagen sea paralelo a la recta que une los centros de proyección, el epipolo de ese plano estará situado en el infinito.

La importancia práctica de la geometría epipolar parte del hecho de que el plano identificado por P, O y O' (llamado **plano epipolar**) interseca cada imagen en una línea, llamada **línea epipolar** (Figura 2.5).

Considérese la tripleta P, $\overline{P}$ y $\overline{P}'$. Dado $\overline{P}$, P puede caer en cualquier punto del rayo definido por O y $\overline{P}$. Pero, dado que la imagen de este rayo, en la imagen derecha, es la línea epipolar a través del punto correspondiente $\overline{P}'$, dicho punto debe estar sobre la línea epipolar. Esta correspondencia establece una aplicación entre puntos de la imagen

Figura 2.5 Líneas Epipolares.

izquierda y rectas de la imagen derecha y viceversa. Una consecuencia de esta correspondencia es que, dado que todos los rayos pasan (por construcción) por el centro de proyección, todas las rectas epipolares deben pasar por el epipolo.

Por tanto, si se determina la aplicación entre puntos de la imagen izquierda (derecha) y las rectas epipolares de la imagen derecha (izquierda), se puede restringir la búsqueda para la correspondencia de $\overline{P}$ a lo largo de la línea epipolar correspondiente. Alternativamente, este conocimiento también se puede usar para verificar si una potencial pareja de puntos correspondientes, lo son de verdad o no. Esta técnica es normalmente una de las más efectivas para detectar las posibles falsas correspondencias debidas a la oclusión.

La **restricción epipolar** es la que indica que los puntos correspondientes deben estar sobre líneas epipolares conjugadas.

2.2.3 Matriz fundamental

La geometría epipolar se puede representar algebraicamente por una matriz, conocida como matriz fundamental.

El primer paso para deducir la matriz fundamental, será buscar la formulación matemática para la línea epipolar (l') de la segunda proyección ($\mathbf{M}'$), de tal forma que sea función de $\bar{P}$, $\mathbf{M}$ y $\mathbf{M}'$. Se parte del supuesto de que P es desconocido.

Para ello, se usará un punto arbitrario P_a, cuya formulación es:

$$P_a = \mathbf{M}^{+}\bar{P} \tag{2.10}$$

Es decir, P_a será el producto de la pseudoinversa de la primera matriz de proyección por la primera proyección del punto. Siendo $\mathbf{M}$ de 3×4 elementos, $\mathbf{M}^{+}$ será, forzosamente, de 4×3 elementos, según la definición de pseudoinversa por la derecha. Dado que el vector $\bar{P}$ es de 3×1 elementos, el vector resultante P_a es de 4×1 elementos, lo que significa que P_a formula matemáticamente un punto.

Este punto tiene una propiedad importante. Si se aplica a la ecuación (2.6) se tiene:

$$\bar{P}_a = \mathbf{M}P_a \tag{2.11}$$

Donde $\bar{P}_a$ es el punto P_a proyectado en el plano π. Sustituyendo el valor de P_a se obtiene:

$$\bar{P}_a = \mathbf{M}\mathbf{M}^{+}\bar{P} \tag{2.12}$$

Aplicando la definición de pseudoinversa, se tiene que

$$\mathbf{M}\mathbf{M}^{+} = \mathbf{I} \tag{2.13}$$

por lo que:

$$\bar{P}_a = \mathbf{I}\bar{P} \tag{2.14}$$

Esto quiere decir que la proyección $\bar{P}_a$ de este punto P_a es igual a $\bar{P}$, por lo que el punto está en la línea entre el centro óptico O y la proyección en $\mathbf{M}$ del punto P. Luego:

$$\bar{P}'_a = \mathbf{M}' P_a \tag{2.15}$$

Aplicando (2.10):

$$\bar{P}'_a = \mathbf{M}' \mathbf{M}^+ \bar{P} \tag{2.16}$$

Para el segundo punto, se aplica la ecuación de (2.8), al centro óptico O de la primera imagen, con objeto de obtener el segundo epipolo. Se tiene:

$$E' = \mathbf{M}' O \tag{2.17}$$

Conocidos estos dos puntos, el cálculo de la línea epipolar es el producto cruz de:

$$l' = E' \times \bar{P}'_a \tag{2.18}$$

Sustituyendo el cálculo de $\bar{P}'_a$ de (2.16) y de E' de (2.17), se tiene:

$$l' = \mathbf{M}' O \times \mathbf{M}' \mathbf{M}^+ \bar{P} \tag{2.19}$$

Aún se puede simplificar más la formulación, sustituyendo la matriz, resultado del producto de $\mathbf{M}' O$, por su antisimétrica; y así se tiene que la nueva formulación de la línea epipolar l' es:

$$l' = \left[\mathbf{M}' O \right]^{\times} \mathbf{M}' \mathbf{M}^+ \bar{P} \tag{2.20}$$

Se define la matriz $\mathbf{F}$ o **matriz fundamental** como:

$$\mathbf{F} = \left[\mathbf{M}' O \right]^{\times} \mathbf{M}' \mathbf{M}^+ \tag{2.21}$$

Así, la línea epipolar l' pasa a ser:

$$l' = \mathbf{F}\bar{P} \tag{2.22}$$

La matriz fundamental se usa para relacionar las dos proyecciones $\bar{P}$ y $\bar{P}'$ de un punto en el espacio. Para deducir la expresión que relaciona las dos proyecciones, se analizará la línea epipolar l' de nuevo. Como $\bar{P}'$ pertenece a la línea epipolar l', se tiene que:

$$\bar{P}'^{T} l' = 0 \tag{2.23}$$

Pero, aplicando la nueva definición de línea epipolar de (2.22), se tiene que:

$$\bar{P}'^{T} \mathbf{F} \bar{P} = 0 \tag{2.24}$$

Las propiedades de la matriz fundamental se pueden dividir en dos tipos, las propiedades algebraicas y las propiedades geométricas.

Algunas de las propiedades algebraicas más importantes son las siguientes:

- $\mathbf{F}$ es una matriz homogénea. En la práctica, significa que para todo κ, si $\kappa \neq 0$, la matriz $\kappa\mathbf{F}$ sigue siendo matriz fundamental.

- Es de rango 2, lo que significa que, luego de reducir la matriz fundamental por medio de operaciones elementales, solo dos renglones de ésta son distintos de cero. Es decir, una de las filas debe ser linealmente dependiente de las otras dos.

- Tiene siete grados de libertad, es decir, de los nueve componentes de la matriz, dos pueden ser calculados a partir de los otro siete.

- La siguiente no solo es una propiedad sino una restricción, llamada restricción de singularidad. En la que el determinante de la matriz fundamental es cero. Lo cual significa que es una matriz singular y no invertible. El espacio nulo izquierdo y el derecho de $\mathbf{F}$ son generados por los vectores que representan (en coordenadas homogéneas) los dos epipolos en las dos imágenes. Si $\mathbf{F}$ es no singular, entonces, las líneas epipolares no coinciden.

Algunas de las propiedades geométricas más importantes son las siguientes:

- Dada una geometría bifocal, la matriz fundamental es función exclusiva de dicha geometría. Esto significa que es independiente de todos los puntos de la escena P y de las proyecciones $\overline{P}$ y $\overline{P}'$ (de los puntos de la escena).

- La matriz fundamental define las líneas epipolares:

$l' = \mathbf{F}\overline{P}$ es la línea epipolar que corresponde a $\overline{P}$ (2.25)

$l = \mathbf{F}^T\overline{P}'$ es la línea epipolar que corresponde a $\overline{P}'$ (2.26)

- La matriz fundamental define los epipolos:

$$\mathbf{F}E = 0 \quad (2.27)$$

$$\mathbf{F}^T E' = 0 \quad (2.28)$$

- La matriz fundamental define la restricción epipolar: Si $\overline{P}$ y $\overline{P}'$ son puntos correspondientes de las dos imágenes, entonces $\overline{P}'^T\mathbf{F}\overline{P} = 0$.

Existe una gran cantidad de algoritmos desarrollados para obtener la matriz fundamental, por lo que se han formado las siguientes cuatro categorías:

a) Métodos de solución directa. Se basa en una propiedad de la matriz fundamental que dice que para cualquier par de puntos correspondientes $\overline{P} \leftrightarrow \overline{P}'$ en las dos imágenes, $\overline{P}'^T\mathbf{F}\overline{P} = 0$. Esto es porque si los puntos $\overline{P}$ y $\overline{P}'$ corresponden, entonces $\overline{P}'$ cae en la línea epipolar, $l' = \mathbf{F}\overline{P}$, correspondiente al punto $\overline{P}$. En otras palabras, $0 = \overline{P}'^T l' = \overline{P}'^T\mathbf{F}\overline{P}$. En esta opción no hay que conocer los parámetros de la cámara, solo se requiere conocer cuántas correspondencias se necesitan para calcular $\mathbf{F}$, y las circunstancias bajo las que la matriz es únicamente definida por estas correspondencias. Es una relación algebraica simple entre puntos correspondientes y la geometría epipolar, la cual permite recuperar la matriz fundamental, dados ocho puntos correspondientes en pares estereoscópicos. Ejemplos de este tipo son el algoritmo normalizado de 8 puntos (ver cuadro 2.1) [Longuet-Higgins, H.C., 1981] y el algoritmo de restricción de dos epipolos [Chengke, W. et al., 2000]. En este último, se introducen nuevas restricciones geométricas para eliminar la inestabilidad de la matriz fundamental, que surge de la inestabilidad de los epipolos. Es más robusto y mejor que el algoritmo normalizado de 8 puntos, pero es no lineal.

b) Métodos algebraicos. Estos métodos son útiles cuando existen más de 8 puntos correspondientes y con errores en las coordenadas. La forma de la matriz fundamental puede ser derivada algebraicamente, con base

en los parámetros intrínsecos y extrínsecos de las cámaras (en términos de las matrices de proyección $\mathbf{M}$ y $\mathbf{M}'$). Esto se logra aproximando una solución que minimice la suma de los cuadrados de la relación algebraica entre los puntos correspondientes y la geometría epipolar.

c) Métodos geométricos. En estos métodos, el objetivo es encontrar una configuración geométrica "legal" (i.e. una que satisfaga las restricciones de la geometría epipolar), tal como la suma de los cuadrados de las distancias de los pares correspondientes, a partir de su mínimo. El mapeo de un punto en una imagen a la línea epipolar correspondiente en la otra imagen, puede ser descompuesto en dos pasos. En el primer paso, el punto $\overline{P}$ se mapea a algún punto de $\overline{P}'$ en la otra imagen, cayendo en la línea epipolar l'. Este punto es una correspondencia potencial para el punto $\overline{P}$. Para poder obtener este mapeo se necesitan conocer los parámetros intrínsecos y extrínsecos. En el segundo paso, l' se obtiene como la línea que une $\overline{P}'$ al epipolo E'. Este problema es numéricamente más complicado que los anteriores, pero da mejores resultados. Un ejemplo de esta categoría es el método Estándar de Gold.

d) Métodos de máxima probabilidad. La idea es recuperar la geometría epipolar que tenga la mayor probabilidad de ajustarse a los datos de medición dados (en este caso, los pares correspondientes). Por ejemplo, el propuesto en [Goshen, L. et al., 2003].

La ecuación (2.24) puede ser utilizada para calcular la matriz desconocida $\mathbf{F}$. Cada punto correspondiente crea una ecuación lineal para las entradas de $\mathbf{F}$. Los coeficientes de esta ecuación pueden ser fácilmente escritos en términos de coordenadas conocidas $\overline{P}$ y $\overline{P}'$. Específicamente, la ecuación correspondiente a un par de puntos $(\overline{x},\overline{y},1)$ y $(\overline{x}',\overline{y}',1)$ es

$$\overline{x}'\overline{x}F_1 + \overline{x}'\overline{y}F_2 + \overline{x}'F_3 + \overline{y}'\overline{x}F_4 + \overline{y}'\overline{y}F_5 + \overline{y}'F_6 + \overline{x}F_7 + \overline{y}F_8 + F_9 = 0 \qquad (2.29)$$

denotando por F el 9-vector hecho por las entradas de $\mathbf{F}$.

Luego, esta ecuación puede ser expresada como el producto interno del vector

$$(\overline{x}'\overline{x}, \overline{x}'\overline{y}, \overline{x}', \overline{y}'\overline{x}, \overline{y}'\overline{y}, \overline{y}', \overline{x}, \overline{y}, 1)F = 0 \qquad (2.30)$$

A partir de un conjunto de m puntos correspondientes, se obtiene un conjunto de ecuaciones lineales de la forma

$$\mathbf{A}F = 0 \qquad (2.31)$$

Donde $\mathbf{A}$ es un conjunto de ecuaciones homogéneas, de a lo más rango 8:

$$\mathbf{A} = \begin{pmatrix} \bar{x}'_1 \bar{x}_1 & \bar{x}'_1 \bar{y}_1 & \bar{x}'_1 & \bar{y}'_1 \bar{x}_1 & \bar{y}'_1 \bar{y}_1 & \bar{y}'_1 & \bar{x}_1 & \bar{y}_1 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \bar{x}'_m \bar{x}_m & \bar{x}'_m \bar{y}_m & \bar{x}'_m & \bar{y}'_m \bar{x}_m & \bar{y}'_m \bar{y}_m & \bar{y}'_m & \bar{x}_m & \bar{y}_m & 1 \end{pmatrix} \quad (2.32)$$

Sin embargo, pueden darse los siguientes casos:

1. Caso mínimo – correspondencias con siete pares de puntos. En el caso de que la matriz $\mathbf{A}$ tenga rango siete, es posible resolver la matriz fundamental haciendo uso de la restricción de singularidad. El caso más importante es cuando son conocidos solo siete puntos correspondientes. Pero también hay otros casos llamados degenerados. La solución a estos problemas pueden encontrarse en [Hartley, R. and Zisserman, A., 2004].

2. Correspondencias con ocho pares de puntos. Se puede calcular la matriz fundamental con ocho pares de puntos. En este caso, se tendrán 8 ecuaciones con nueve incógnitas. La novena ecuación se puede obtener sabiendo que el rango de la matriz fundamental es dos. Por ello, una de las filas debe ser linealmente dependiente de otra. Se escoge una fila al azar, y se dice que es igual a otra fila (también elegida al azar) multiplicada por cualquier constante distinta de cero. Esto dará una ecuación más. Lo cual significa que se tendrán exactamente nueve ecuaciones con nueve incógnitas. Como el rango de $\mathbf{A}$ es exactamente 8, entonces la solución es única, y puede ser encontrada por métodos lineales.

3. Correspondencias con más de ocho pares de puntos. Si se tienen más de ocho puntos, quiere decir que los datos no son exactos, porque hay ruido en los puntos correspondientes. Entonces, el rango de $\mathbf{A}$ puede ser mayor que 8 (de hecho igual a 9, dado que $\mathbf{A}$ tiene 9 columnas). En este caso, se pueden emplear las restricciones del algoritmo de ocho puntos, que aseguran que la matriz fundamental va a cumplir sus propiedades matemáticas; para luego aplicar un procedimiento que optimice el error cuadrático medio de ocho variables, respecto a las restricciones derivadas de las ecuaciones del algoritmo de ocho puntos. Las opciones para minimizar dicho error pueden ser: el RANSAC, el descenso por gradiente, las redes neuronales, los algoritmos genéticos o los algoritmos miméticos.

2.2.4 Matriz esencial

Hasta ahora se ha trabajado en coordenadas de los sistemas de referencia asociados a las cámaras. Sin embargo, cuando se miden los puntos proyectados, estos están medidos en términos de píxeles. Los parámetros que transforman las coordenadas de la cámara en coordenadas de píxeles, se conocen como parámetros intrínsecos [Hartley, R. and Zisserman, A., 2004], cuyas matrices se denotan como $\mathbf{K}$ y $\mathbf{K}'$ de las cámaras izquierda y derecha respectivamente.

Si las calibraciones de las matrices $\mathbf{K}$ y $\mathbf{K}'$ son conocidas, entonces, se pueden aplicar sus inversas para obtener los puntos de la imagen expresados en coordenadas normalizadas. Es decir,

Cuadro 2.1 Algoritmo normalizado de ocho puntos

1. Selección de puntos. Se seleccionan ocho pares de puntos correspondientes, denotados como $\mathcal{P} = \bar{P}_1, \bar{P}_2, \ldots, \bar{P}_8$ para la primera imagen y $\mathcal{P}' = \bar{P}_1', \bar{P}_2', \ldots, \bar{P}_8'$ para la segunda imagen. Estos puntos son coordenadas de píxel no homogéneas que se denotan como $\bar{P}_j = (\bar{x}_j, \bar{y}_j)$ correspondientes con $\bar{P}_j' = (\bar{x}'_j, \bar{y}'_j)$ donde j = 1, 2,...,8.

2. Normalización. El objetivo es transformar las coordenadas de la imagen de acuerdo a transformaciones normalizadas que consisten en una translación y una escala. Para ello se siguen los siguientes pasos:

   a. Se calculan los centroides de tal forma que los puntos de referencia están en el origen de las coordenadas:

   Para $\mathcal{P}$ $$\mu = \frac{\sum_{j=1}^{8} \bar{x}_j}{8} \quad \text{y} \quad \varphi = \frac{\sum_{j=1}^{8} \bar{y}_j}{8} \tag{2.33}$$

   Para $\mathcal{P}'$ $$\mu' = \frac{\sum_{j=1}^{8} \bar{x}'_j}{8} \quad \text{y} \quad \varphi' = \frac{\sum_{j=1}^{8} \bar{y}'_j}{8} \tag{2.34}$$

   b. Se calcula la norma

   Para cada $\bar{P}_j$ de $\mathcal{P}$ $$A_j = \left(\bar{x}_j, \bar{y}_j\right) - \left(\mu, \varphi\right) \tag{2.35}$$

   Para cada $\bar{P}_j'$ de $\mathcal{P}'$ $$A_j' = \left(\bar{x}'_j, \bar{y}'_j\right) - \left(\mu', \varphi'\right) \tag{2.36}$$

   Para $\mathcal{P}$ $$n = \frac{\sum_{j=1}^{8} \left\| A_j \right\|}{8} = \frac{\sum_{j=1}^{8} \left\| \left(\bar{x}_j, \bar{y}_j\right) - \left(\mu, \varphi\right) \right\|}{8} = \frac{\sum_{j=1}^{8} \sqrt{(\bar{x}_j - \mu)^2 + (\bar{y}_j - \varphi)^2}}{8} \tag{2.37}$$

   Para $\mathcal{P}'$ $$n' = \frac{\sum_{j=1}^{8} \left\| A_j' \right\|}{8} = \frac{\sum_{j=1}^{8} \left\| \left(\bar{x}'_j, \bar{y}'_j\right) - \left(\mu', \varphi'\right) \right\|}{8} = \frac{\sum_{j=1}^{8} \sqrt{(\bar{x}'_j - \mu')^2 + (\bar{y}'_j - \varphi')^2}}{8} \tag{2.38}$$

c. Se obtiene la traslación y el escalamiento

Para $\mathcal{P}$ $$t = \frac{\sqrt{2}}{n} \tag{2.39}$$

$$\mathbf{T} = \begin{pmatrix} t & 0 & -t\mu \\ 0 & t & -t\varphi \\ 0 & 0 & 1 \end{pmatrix} \tag{2.40}$$

Para $\mathcal{P}'$ $$t' = \frac{\sqrt{2}}{n'} \tag{2.41}$$

$$\mathbf{T}' = \begin{pmatrix} t' & 0 & -t'\mu' \\ 0 & t' & -t'\varphi' \\ 0 & 0 & 1 \end{pmatrix} \tag{2.42}$$

d. Se trasladan y escalan los puntos (en coordenadas homogéneas)

Para cada $\overline{P}_j$ de $\mathcal{P}$ $$\tilde{P}_j = \mathbf{T}\left(\overline{x}_j, \overline{y}_j, 1\right)^T \tag{2.43}$$

Para cada $\overline{P}'_j$ de $\mathcal{P}'$ $$\tilde{P}'_j = \mathbf{T}'\left(\overline{x}'_j, \overline{y}'_j, 1\right)^T \tag{2.44}$$

3. Encontrar la matriz fundamental no restringida $\mathbf{F}'$. Se determina $\mathbf{F}'$ a partir de vectores singulares correspondientes al valor singular más pequeño de $\mathbf{A}$, donde $\mathbf{A}$ está compuesta de las correspondencias $\tilde{P} \leftrightarrow \tilde{P}'$. Para ello se aplica la solución directa siguiendo los siguientes pasos:

a. Se obtiene la matriz $\mathbf{A}$ (2.32), a partir de las correspondencias $\tilde{P} \leftrightarrow \tilde{P}'$ en coordenadas no homogéneas, es decir, $\tilde{P}_j = \left(\overline{x}_j, \overline{y}_j\right)$ correspondientes con $\tilde{P}'_j = \left(\overline{x}'_j, \overline{y}'_j\right)$ donde *j*=1,2...8.

b. Se obtiene la descomposición SVD (ver apéndice 4) de la matriz $\mathbf{A}$ tal que $\mathbf{A} = \mathbf{UDV}^{\mathbf{T}}$.

b. Se encuentra el vector correspondiente al valor propio más pequeño de $\mathbf{V}$. Este vector de 9×1, V, es la última columna de $\mathbf{V}$ porque es el correspondiente al vector más pequeño de $\mathbf{D}$, el cual está organizado en forma descendente y siempre es la última columna ([Hartley, R. and Zisserman, A., 2004], [Haralick, R. et al., 1989]).

c. La matriz fundamental no restringida $\mathbf{F}'$ está formada por el vector encontrado en el paso anterior:

$$\mathbf{F}' = \begin{pmatrix} V_1 & V_2 & V_3 \\ V_4 & V_5 & V_6 \\ V_7 & V_8 & V_9 \end{pmatrix} \tag{2.45}$$

4. Denormalización. Se deshace la transformación aplicada en el paso 2

$$\tilde{\mathbf{F}} = \mathbf{T}'^{T}\mathbf{F}'\mathbf{T} \tag{2.46}$$

5. Restringir la matriz fundamental $\tilde{\mathbf{F}}$ **.** Forzar la restricción de singularidad, en la que $\det(\tilde{\mathbf{F}}) = 0$ y el rango de $\tilde{\mathbf{F}}$ es 2. Para lograrlo se siguen los siguientes pasos:

a. Se determina la SVD de la matriz fundamental no restringida $\tilde{\mathbf{F}}$, tal que

$$\tilde{\mathbf{F}} = \tilde{\mathbf{U}}\tilde{\mathbf{D}}\tilde{\mathbf{V}}^{\mathbf{T}} \tag{2.47}$$

b. Probablemente, debido a errores numéricos en el cálculo y a errores en la exactitud de los puntos, la matriz $\tilde{\mathbf{D}}$ tendrá la forma:

$$\tilde{\mathbf{D}} = \begin{pmatrix} k_1 & 0 & 0 \\ 0 & k_2 & 0 \\ 0 & 0 & k_3 \end{pmatrix} \tag{2.48}$$

Por lo que deberá modificarse a la forma:

$$\hat{\mathbf{D}} = \begin{pmatrix} k_1 & 0 & 0 \\ 0 & k_2 & 0 \\ 0 & 0 & 0 \end{pmatrix} \tag{2.49}$$

Que es la forma que la matriz $\tilde{\mathbf{D}}$ tendría si la matriz fundamental no restringida $\tilde{\mathbf{F}}$ fuese exactamente la matriz fundamental $\mathbf{F}$.

c. Se estima la matriz fundamental definitiva $\mathbf{F}$ como

$$\mathbf{F} = \tilde{\mathbf{U}}\hat{\mathbf{D}}\tilde{\mathbf{V}}^{\mathbf{T}} . \tag{2.50}$$

$$\hat{P} = \mathbf{K}^{-1}\bar{P} \tag{2.51}$$

$$\hat{P}' = \mathbf{K}'^{-1}\bar{P}' \tag{2.52}$$

Cuando los parámetros intrínsecos de la cámara son conocidos, es posible obtener la matriz esencial ($\mathbf{E}$). La cual establece una unión natural entre la restricción epipolar y los parámetros extrínsecos del sistema estéreo. Se considera a la matriz fundamental $\mathbf{F}$ como la matriz que relaciona dos proyecciones, es decir, que relaciona a las dos imágenes de una misma escena; mientras que la matriz esencial $\mathbf{E}$ relaciona el movimiento relativo para llegar de una vista a la otra.

Si se aplican los puntos de (2.51) y (2.52) a la fórmula de la matriz fundamental (2.24), entonces, de acuerdo a la demostración dada en [Hartley, R. and Zisserman, A., 2004], se obtiene la siguiente expresión

$$\hat{P}'^{T}\left(T \times \mathbf{R}\right)\hat{P} = 0 \tag{2.53}$$

Luego, se puede definir la matriz esencial como:

$$\mathbf{E} = T \times \mathbf{R} \tag{2.54}$$

Por lo que se tendrá la expresión:

$$\hat{P'}^T \mathbf{E} \hat{P} = 0 \tag{2.55}$$

Las **propiedades de la matriz esencial** son:

- Es de rango 2, lo que significa que luego de reducir la matriz esencial por medio de operaciones elementales, solo dos renglones de ésta son distintos de cero. Es decir, una de las filas debe ser linealmente dependientes de las otras dos.

- Siendo T el vector de translación de 3×1 entre las dos vistas, se tiene que:

$$T^T \mathbf{E} = 0 \tag{2.56}$$

- Y $\mathbf{R}$ la matriz de rotación de 3×3 entre las dos imágenes, se tiene que:

$$\mathbf{E}(\mathbf{R}T) = 0 \tag{2.57}$$

- Siendo la descomposición SVD de la matriz esencial:

$$\mathbf{E} = \breve{\mathbf{U}}\breve{\mathbf{D}}\breve{\mathbf{V}}^T \tag{2.58}$$

La forma general de la matriz $\breve{\mathbf{D}}$ será:

$$\breve{\mathbf{D}} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} \tag{2.59}$$

- La matriz de rotación $\mathbf{R}$ será:

$$\mathbf{R} = \breve{\mathbf{U}} \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \breve{\mathbf{V}}^T \tag{2.60}$$

- y la matriz correspondiente a la translación T será:

$$T = \breve{\mathbf{U}} \begin{pmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \breve{\mathbf{U}}^T \tag{2.61}$$

donde el vector de traslación es la última columna de $\breve{\mathbf{U}}$.

En caso de que no se cuente con los parámetros intrínsecos de la cámara, es decir, que no se tengan ni T ni $\mathbf{R}$ (que es el caso de este trabajo), se tendrá que encontrar una matriz $\mathbf{E}'$ que se aproxime a la matriz esencial $\mathbf{E}$. Para ello, existen diversos algoritmos entre los que se pueden mencionar:

a) El algoritmo lineal (ver cuadro 2.2). Originalmente, este algoritmo aplicado a la estimación de los parámetros de movimiento, fue desarrollado en [Tsai, R.Y. and Huang, T.S., 1984], unificado en [Zhuang, X. et al., 1986], y simplificado en [Haralick, R. et al., 1989]. Su principal ventaja es que es simple y rápido, y siempre puede encontrar una solución única, excepto en los casos degenerados. Sin embargo, es muy sensible al ruido y a puntos correspondientes incorrectos. El principal problema de este algoritmo es la estimación de mínimos cuadrados, la cual se basa en una evaluación de la magnitud de los residuos.

Cuadro 2.2 Algoritmo lineal

El algoritmo lineal se calcula igual que el algoritmo normalizado de ocho puntos. La única diferencia es que, en la etapa 5 (en la que se forza la restricción de singularidad), en el paso b, la matriz $\hat{\mathbf{D}} = \breve{\mathbf{D}}$ de la ecuación (2.59). La cual es una de las propiedades de la matriz esencial.

b) El algoritmo robusto [Haralick, R. et al., 1989] (ver cuadro 2.3). Se diseñó para evitar los problemas generados por la estimación de mínimos cuadrados del algoritmo lineal. Tiene buena resistencia y robustez a errores de correspondencia.

2.3 "Thin-Plate spline"

El nombre "*Thin-Plate spline*" (TPS) se refiere a una analogía física de la deformación de una hoja de metal delgada (como un plato). El TPS se desarrolló en [Wahba, G., 1990] como una herramienta de propósito general, que genera un mapeo funcional suave para aprendizaje supervisado. El trabajo presentado en [Bookstein, F.L., 1989], fue el primero en usar TPS para generar un mapeo espacial suave entre dos conjuntos de puntos, con correspondencias uno a

uno en imágenes médicas. Esta técnica se ha elegido en varios trabajos, como en [Chui, H. and Rangarajan, A., 2000], para calcular la correspondencia y la transformación no rígida entre conjuntos de puntos.

Cuadro 2.3 Algoritmo robusto

1. Selección de puntos. Se seleccionan nueve pares de puntos correspondientes, denotados como $\mathcal{P} = \overline{P}_1, \overline{P}_2, \ldots, \overline{P}_9$ para la primera imagen y $\mathcal{P}' = \overline{P}_1', \overline{P}_2', \ldots, \overline{P}_9'$ para la segunda imagen.

2. Estimación de la matriz $\mathbf{E}'$. Se tiene que, con base en (2.55), para cualquier par de puntos correspondientes $\left[\left(\overline{x}, \overline{y}\right), \left(\overline{x}', \overline{y}'\right)\right]$, la matriz $\mathbf{E}'$ satisface la siguiente ecuación lineal homogénea con respecto a los nueve elementos de $\mathbf{E}'$.

$$\left(\overline{x}', \overline{y}', 1\right)^T \mathbf{E}' \left(\overline{x}, \overline{y}, 1\right) = 0 \qquad (2.62)$$

Donde

$$\mathbf{E}' = \begin{pmatrix} H_1 & H_2 & H_3 \\ H_4 & H_5 & H_6 \\ H_7 & H_8 & H_9 \end{pmatrix} \qquad (2.63)$$

Luego, haciendo:

$$H = \begin{pmatrix} H_1 & H_2 & H_3 & H_4 & H_5 & H_6 & H_7 & H_8 & H_9 \end{pmatrix}^T \qquad (2.64)$$

la ecuación (2.62) puede ser transformada en la siguiente ecuación lineal, para estimar $\mathbf{E}'$:

$$\mathbf{A}H = 0 \qquad (2.65)$$

Para hacer el algoritmo más robusto [Haralick, R. et al., 1989], se introduce una matriz de $m \times m$ de bipesos, $\mathbf{W}$, donde m es el número de puntos correspondientes, tal que:

$$\mathbf{W}\mathbf{A}H = 0 \qquad (2.66)$$

Obtención de $\mathbf{W}$. Inicialmente $\mathbf{W}$ es la matriz identidad. Luego, se utiliza un estimador de mínimos cuadrados en el que iterativamente se recalcula $\mathbf{W}$, hasta que algún criterio sea satisfecho. En el caso de este trabajo, se itera ω veces (establecido en 300 iteraciones debido a los resultados obtenidos en los experimentos con datos controlados).

Para obtener los valores de $\mathbf{W}$ se utiliza una función ψ de estimador de bipesos, que puede ser representada como sigue:

$$\mathbf{W} = \begin{pmatrix} \psi(u_1) & 0 & \cdots & 0 \\ 0 & \psi(u_2) & \cdots & 0 \\ \vdots & \vdots & \vdots & 0 \\ 0 & 0 & \cdots & \psi(u_m) \end{pmatrix} \tag{2.67}$$

Tal que

$$\psi(u_i) = \begin{cases} u_i\left(1-u_i^2\right)^2 & |u_i| \le 1 \\ 0 & |u_i| > 1 \end{cases} \tag{2.68}$$

Gráficamente se ve de la siguiente forma:

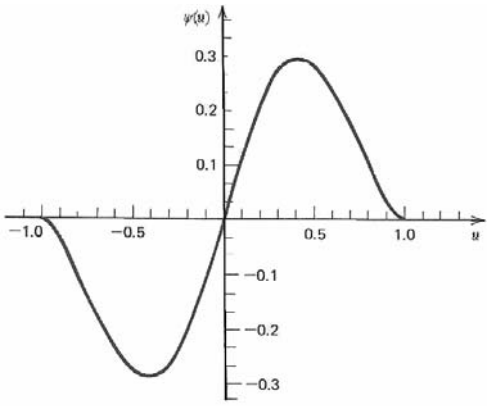


Figura 2.6 Gráfica de una función de bipesos.

Donde u_i es la función objetivo:

$$u_i = \frac{\phi_i(\varepsilon)}{cs} \tag{2.69}$$

c es una constante fijada en 4 con base en [Haralick, R. et al., 1989]

s es la mediana de $\phi_i(\varepsilon)$

$\phi_i(\varepsilon)$ $1 \le i \le m$, con el error residual, que se obtiene de la siguiente forma:

$$\phi_i(\varepsilon) = \frac{\varepsilon_i}{1 - B_i} \tag{2.70}$$

Donde B_i se define más adelante y ε es un vector de $m \times 1$ tal que

$$\varepsilon = \mathbf{A}H \tag{2.71}$$

Obtención de H

i. Se obtiene la descomposición SVD (ver apéndice 4) de $\mathbf{WA}$ tal que $\mathbf{WA} = \dot{\mathbf{U}}\dot{\mathbf{D}}\dot{\mathbf{V}}^T$. Donde $\dot{\mathbf{U}}$ es una matriz de $m \times m$, $\dot{\mathbf{D}}$ es una matriz diagonal de $m \times n$ y $\dot{\mathbf{V}}$ es una matriz de $n \times n$ donde n = 9.

ii. H es el vector correspondiente al valor propio más pequeño de $\dot{\mathbf{V}}$. Este vector de 9×1 es la última columna de $\dot{\mathbf{V}}$, y es el correspondiente al valor más pequeño de $\dot{\mathbf{D}}$, el cual está organizado en forma descendente y siempre es la última columna ([Hartley, R. and Zisserman, A., 2004], [Haralick, R. et al., 1989]).

B_i es un vector de $m \times 1$ que se obtiene de la siguiente forma:

$$B_i = \sum_{k=1}^{9} \dot{U}_{ik}^2 \tag{2.72}$$

Como puede verse en la ecuación anterior, este algoritmo requiere 9 correspondencias.

Con el objetivo de aplicar esta idea al problema de la transformación de coordenadas, el alisado del plato se puede interpretar como un desplazamiento en la dirección de las coordenadas x o y, de tal manera que el TPS permite una perfecta alineación de los puntos, mientras minimiza la energía de la deformación requerida para hacerlo. En la figura 2.7.a, se muestra un ejemplo de una proyección en perspectiva de puntos 3D en el plano de la imagen, y en la figura 2.7.b se puede ver cómo se deforma la cuadrícula, para hacer que los puntos coincidan.

El TPS, aplicado a la alineación de los puntos correspondientes, utiliza dos "*splines*", uno para el desplazamiento en la dirección de $\bar{x}$ y otra para el desplazamiento en la dirección de $\bar{y}$. El resultado de ambas transformaciones son combinadas en un único mapa.

El TPS es la generalización 2D de la "*spline*" cúbica (ver apéndice 1). Es una función base, formada por componentes afines globales y componentes no afines locales, comúnmente usada para representar el mapeo de coordenadas de $\mathbb{R}^2$ a $\mathbb{R}^2$. Su objetivo es minimizar la energía de la deformación, necesaria para hacer coincidir dos conjuntos de puntos. Para lograrlo, se obtiene la segunda derivada del mapeo espacial. Consecuentemente, el termino de suavizado del TPS es solamente dependiente de los componentes no afines. Esta es una propiedad deseable, especialmente cuando se compara contra otros "*splines*", dado que los parámetros de la posición global, incluidos en la transformación afín, no son penalizados. A continuación se dará una explicación más detallada de su funcionamiento.

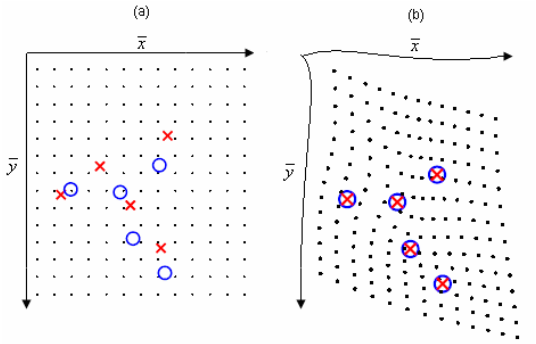


Figura 2.7 Ejemplo del TPS aplicado a la alineación de puntos correspondientes: (a) proyección en perspectiva de puntos 3D en el plano de la imagen, (b) cuadrícula plegada que muestra el resultado de la deformación necesaria para hacer que los puntos coincidan.

2.3.1 Algoritmo RANSAC-TPS

El objetivo de este algoritmo es hacer el mapeo no rígido, a través del TPS, entre puntos característicos de dos imágenes, correspondientes al mismo elemento de la escena, que son parte de una secuencia de imágenes consecutivas.

Una de las desventajas del TPS es que la solución es computacionalmente muy costosa, debido a que durante su cálculo (el cual será presentado más adelante) se debe invertir una matriz de tamaño $k \times k$, donde k es el número de puntos correspondientes. En [Donato, G. and Belongie, S., 2002] se muestran tres métodos que tratan de solucionar el

modelo de una manera más económica, al procesar solo un subconjunto de k, de tamaño n, para, posteriormente, aplicar la transformación resultante al resto de los puntos. Una alternativa a estos métodos, es aplicar el RANSAC (ver apéndice 2) para seleccionar los puntos que conformarán dicho subconjunto.

Esta última fue la opción elegida para ser implementada en esta tesis. Solo que, usualmente, el tamaño del subconjunto definido en este método debe ser constante. Pero, debido a que se asume que la distribución de correspondencias incorrectas es desconocida (y difícil de predecir, por lo menos en los problemas de estimación de movimiento), y, tomando en cuenta que incrementar n también incrementa la probabilidad de incluir correspondencias incorrectas en las muestras, se siguió una aproximación incremental. El algoritmo resultante se nombró RANSAC-TPS.

Algoritmo RANSAC-TPS (ver cuadro 2.4)

1. *Selección de un subconjunto de puntos*

   Los conjuntos de puntos correspondientes se denotan como $\mathcal{R} = \bar{R}_1, \bar{R}_2, \ldots, \bar{R}_k$ para la primera imagen y $\mathcal{R}' = \bar{R}'_1, \bar{R}'_2, \ldots, \bar{R}'_k$ para la segunda imagen. Estos puntos son coordenadas de píxel, tal que $\bar{R}_l = (\bar{x}_l, \bar{y}_l)$ y $\bar{R}'_l = (\bar{x}'_l, \bar{y}'_l)$, con l = 1, 2,…, k, donde k es el número de puntos correspondientes. De los cuales se seleccionan aleatoriamente una muestra del 30% ($n = 0.30k$). Denotando como $\mathcal{S} = \bar{S}_1, \bar{S}_2, \ldots, \bar{S}_n$ el conjunto de puntos de $\mathcal{R}$ y como $\mathcal{S}' = \bar{S}'_1, \bar{S}'_2, \ldots, \bar{S}'_n$ el conjunto de puntos de $\mathcal{R}'$ correspondientes a ellos. Estos puntos son coordenadas de píxel que se denotan como $\bar{S}_j = (\bar{x}_j, \bar{y}_j)$, $\bar{S}'_j = (\bar{x}'_j, \bar{y}'_j)$ con j=1, 2,…, n.

2. *Cálculo del "Thin-Plate spline* "

   El cálculo del TPS puede dividirse en dos tareas principales:

   a. *Transformación del TPS*. La transformación del TPS está compuesta de una parte afín y una parte deformable o no afín, cuyos coeficientes (*i*) se obtienen a partir del subconjunto de puntos $(\mathcal{S}, \mathcal{S}')$. Dichos coeficientes (*ii*) son aplicados a todos los puntos de $\mathcal{R}$, para obtener puntos aproximados (llamados $\mathcal{B}$) a sus puntos correspondientes en $\mathcal{R}'$.

      i. *Obtención de coeficientes*

         Para calcular los coeficientes, primero es necesario obtener una matriz de $(n+3)\times(n+3)$, L. Tal que,

$$\mathbf{L} = \begin{pmatrix} \mathbf{K} & \mathbf{H} \\ \mathbf{H}^{\mathbf{T}} & \mathbf{O} \end{pmatrix} \tag{2.73}$$

Donde $\mathbf{O}$ es una matriz de 3×3 de ceros, $\mathbf{H}$ es el subconjunto de puntos seleccionados, $\mathcal{S}$, en coordenadas homogéneas:

$$\mathbf{H} = \begin{pmatrix} 1 & \overline{S}_1 \\ 1 & \overline{S}_2 \\ \vdots & \\ 1 & \overline{S}_n \end{pmatrix} = \begin{pmatrix} 1 & \overline{x}_1 & \overline{y}_1 \\ 1 & \overline{x}_2 & \overline{y}_2 \\ \vdots & & \\ 1 & \overline{x}_n & \overline{y}_n \end{pmatrix} \tag{2.74}$$

Y $\mathbf{K}$ es una matriz de $n\times n$ que contiene a $\zeta(l_{ij})$, la cual es una función radial básica (Figura 2.8), que se basa en el subconjunto de puntos seleccionados $\mathcal{S}$. La función $\zeta(l_{ij})$ se conoce como TPS Kernel y contiene la información de la estructura interna del conjunto de puntos.

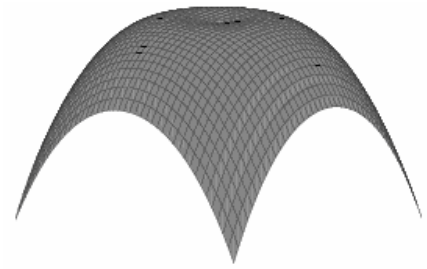


Figura 2.8 Función radial básica.

$$\mathbf{K} = \begin{pmatrix} 0 & \zeta(l_{12}) & \cdots & \zeta(l_{1n}) \\ \zeta(l_{21}) & 0 & \cdots & \zeta(l_{2n}) \\ \vdots & \vdots & \ddots & \vdots \\ \zeta(l_{n1}) & \zeta(l_{n2}) & \cdots & 0 \end{pmatrix} \tag{2.75}$$

Tal que

$$\mathbf{K}_{ij} = \zeta(l_{ij}) \tag{2.76}$$

$$\zeta(l_{ij}) = l_{ij}^2 \log l_{ij}^2 \tag{2.77}$$

$$l_{ij} = \left\| \overline{S}_i - \overline{S}_j \right\| = \left\| (\overline{x}_i, \overline{y}_i) - (\overline{x}_j - \overline{y}_j) \right\| = \sqrt{(\overline{x}_i - \overline{x}_j)^2 + (\overline{y}_i - \overline{y}_j)^2} \tag{2.78}$$

Luego, los coeficientes de la transformación del TPS están formados a partir de la matriz $\mathbf{C}$ de tamaño $(n+3)\times 2$

$$\mathbf{C} = \mathbf{L}^{-1}\mathbf{B}^{\mathbf{T}} \tag{2.79}$$

donde $\mathbf{B}$ es una matriz de $2\times(n+3)$ conformada por las matrices

$$\mathbf{B} = [\mathcal{S}'^{T} \quad \mathbf{G}] \tag{2.80}$$

tal que $\mathcal{S}'$ es una matriz de $n\times 2$

$$\mathcal{S}' = \begin{pmatrix} \overline{x}'_1 & \overline{y}'_1 \\ \overline{x}'_2 & \overline{y}'_2 \\ \vdots & \\ \overline{x}'_n & \overline{y}'_n \end{pmatrix} \tag{2.81}$$

y $\mathbf{G}$ es una matriz de ceros de 2×3.

Se nombran Cx y Cy las dos columnas de C

$$Cx = [\mathbf{C}_{11}, \mathbf{C}_{21}, \ldots, \mathbf{C}_{(n+3)1}]^T \tag{2.82}$$

$$Cy = [\mathbf{C}_{12}, \mathbf{C}_{22}, \ldots, \mathbf{C}_{(n+3)2}]^T \tag{2.83}$$

Finalmente, se obtienen los coeficientes de la parte afín, donde

$$Ax = [Cx_{(n+1)}, Cx_{(n+2)}, Cx_{(n+3)}]^T \tag{2.84}$$

$$Ay = [Cy_{(n+1)}, Cy_{(n+2)}, Cy_{(n+3)}]^T \tag{2.85}$$

y los coeficientes de la parte deformable o no afín, donde

$$\mathbf{N} = [Wx \quad Wy] \tag{2.86}$$

$$Wx = [Cx_1, Cx_2, \ldots, Cx_n]^T \tag{2.87}$$

$$Wy = [Cy_1, Cy_2, \ldots, Cy_n]^T \tag{2.88}$$

ii. *Obtención de los puntos transformados*

Para obtener los puntos $\mathcal{B}'$ (tal que $\mathcal{B}' = \overline{B}'_1, \overline{B}'_2, \ldots, \overline{B}'_n$), los cuales son el conjunto de puntos aproximados a $\mathcal{R}'$, la transformación TPS utiliza dos "s*plines*", una para el desplazamiento en la

dirección de $\bar{x}$ y otra para el desplazamiento en la dirección de $\bar{y}$. Los resultados de ambas son combinados en un único mapa denotado por γ, el cual aplica los coeficientes calculados en el punto anterior.

Así que, para cada punto $\bar{R}_i \in \mathcal{R}$, se aplica la transformación TPS, $\gamma\left(\bar{R}_i\right)$. Tal que,

$$\bar{B}_i' = \gamma(\bar{R}_i) = [\gamma_x(\bar{R}_i) \quad \gamma_y(\bar{R}_i)] = [\gamma_x(\bar{x}_i, \bar{y}_i) \quad \gamma_y(\bar{x}_i, \bar{y}_i)] \tag{2.89}$$

$$\gamma_x(\bar{x}_i, \bar{y}_i) = [1 \quad \bar{x}_i \quad \bar{y}_i] A x + \sum_{h=1}^{n} W x_h \zeta(d_{h\,i}) \tag{2.90}$$

$$\gamma_y(\bar{x}_i, \bar{y}_i) = [1 \quad \bar{x}_i \quad \bar{y}_i] A y + \sum_{h=1}^{n} W y_h \zeta(d_{h\,i}) \tag{2.91}$$

donde $1 \leq i \leq k$.

Luego, $\zeta(d_{hi}) = d_{hi}^2 \log d_{hi}^2$, es una función radial básica. Donde d_{hi} es la distancia del origen Cartesiano, entre los puntos de $\mathcal{R}$ y los puntos del subconjunto seleccionado $\mathcal{S}$. Donde

$$d_{h\,i} = \left\| \bar{S}_h - \bar{R}_i \right\| = \left\| (\bar{x}_h, \bar{y}_h) - (\bar{x}_i - \bar{y}_i) \right\| = \sqrt{(\bar{x}_h - \bar{x}_i)^2 + (\bar{y}_h - \bar{y}_i)^2} \tag{2.92}$$

b. *Cálculo de la energía de la deformación.* Una vez que se calculan los puntos $\mathcal{B}'$, se obtiene la función de energía, δ, usada para medir la diferencia que existe entre éstos y los datos reales $\mathcal{R}'$.

$$\delta = \sum_{i=1}^{k} | | \bar{R}_i' - \bar{B}_i' | |^2 + \lambda \underbrace{\iint \left(\left(\frac{\partial^2 \gamma}{\partial \bar{x}^2} \right)^2 + 2 \left(\frac{\partial^2 \gamma}{\partial \bar{x} \partial \bar{y}} \right)^2 + \left(\frac{\partial^2 \gamma}{\partial \bar{y}^2} \right)^2 \right) d\bar{x} d\bar{y}}_{i_F} \tag{2.93}$$

La primera parte de la expresión determina qué tanto se aproximan los puntos, transformados con el TPS, a los puntos reales $\mathcal{R}'$.

La segunda parte de la expresión, i_F, es llamada *norma de flexión integral* y cuantifica la energía de la deformación, es decir, es la medida de suavizado. Si se denota el bloque superior izquierdo $n \times n$ de $\mathbf{L}^{-1}$ como $\mathbf{L}_n^{-1}$, se puede mostrar ([Chui, H. and Rangarajan, A., 2003], [Donato, G. and Belongie, S., 2002]) que

$$i_F \quad \alpha \quad S'^T \mathbf{L}_n^{-1} S' = \mathbf{N}^T \zeta(d_{ij}) \mathbf{N} \tag{2.94}$$

Así, la función de energía de la deformación queda como

$$\delta(\mathcal{B}', \mathcal{R}') = med\{||\mathcal{R}' - \mathcal{B}'||\} + \lambda\, trace\left(\mathbf{N}^T \zeta(d'_{ij})\mathbf{N}\right) \quad (2.95)$$

Donde med es la mediana de los residuos de $||\mathcal{R}' - \mathcal{B}'||$, para determinar qué tanto se aproximan los puntos transformados con el TPS ($\mathcal{B}'$) a los puntos $\mathcal{R}'$, y $trace$ [Chui, H. and Rangarajan, A., 2003] es la suma de los elementos de la diagonal de la energía de la deformación $\left(\mathbf{N}^T \zeta(d_{ij})\mathbf{N}\right)$.

λ es el parámetro de regularización, el cual es un escalar positivo que controla la cantidad de suavizado. Es difícil estimarlo**,** porque, grandes valores de éste limitan el rango de no rigidez de la transformación, mientras que, pequeños valores hacen que la transformación sea muy flexible y esto provoca que el algoritmo sea inestable. El valor de este parámetro depende mucho de la cantidad de correspondencias iniciales incorrectas, ya que, un valor de λ que podría ser adecuado para datos que fueron extraídos, por ejemplo, con SIFT (el cual da datos muy precisos), no funciona bien para datos extraídos con gradiente. En [Chui, H. and Rangarajan, A., 2000] se utilizó un parámetro de entropía que ayudó a reducir λ gradualmente. Pese a ello, en esta tesis, λ se estableció como una constante (cuyo valor se fijó en 10.000 debido a los buenos resultados que se obtienen).

3. *Optimización de la transformación*

El objetivo que se persigue es aproximar lo más posible los puntos $\mathcal{B}'$ a los puntos $\mathcal{R}'$**,** para obtener una transformación óptima. Esto se logra minimizando la energía de la deformación δ**.**

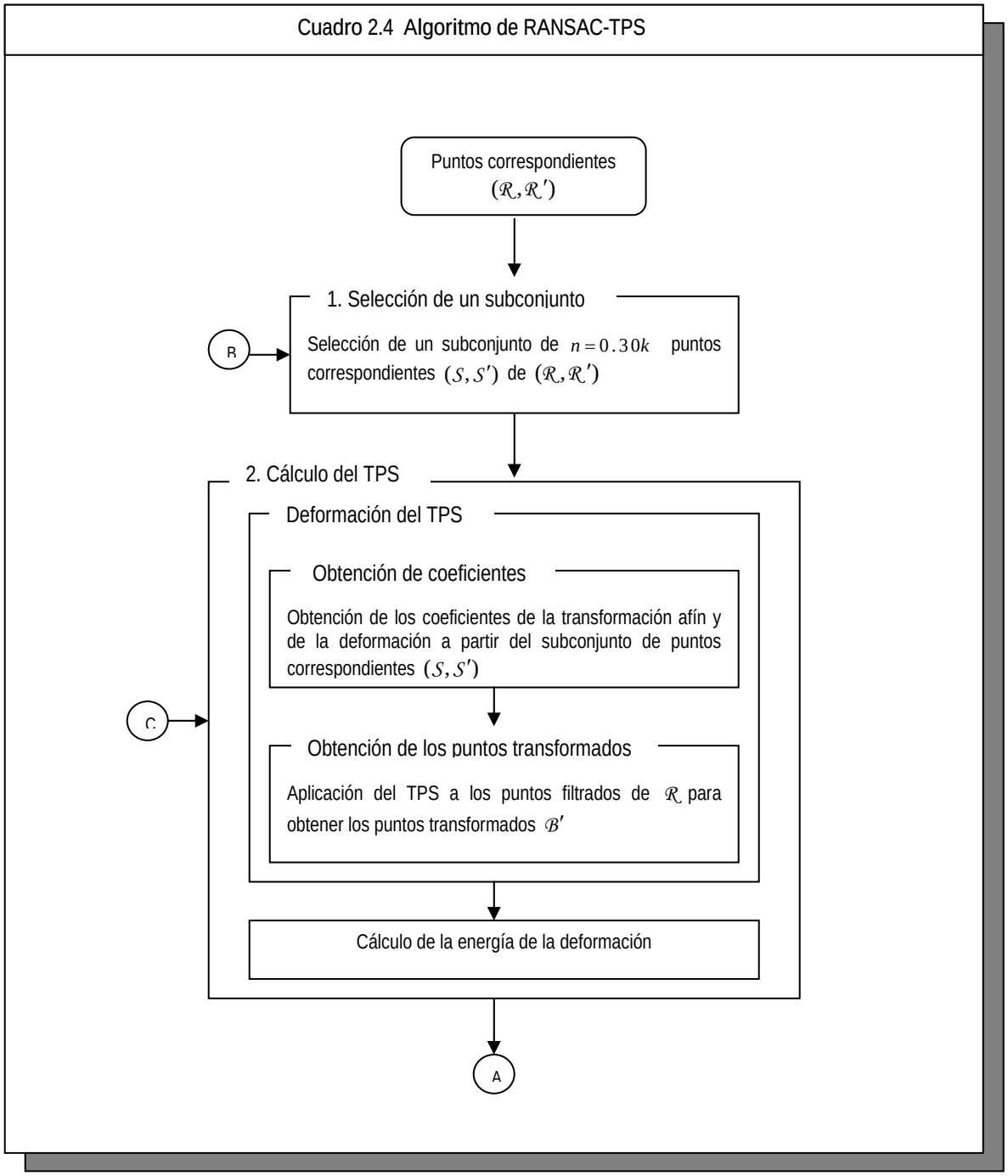
Para ello, se aplica un procedimiento de optimización (que termina de completar el RANSAC), en el que se repiten t veces los pasos 2 y 3. Cada vez que se itera, se compara la energía generada δ con la energía de la iteración anterior (la cual se llamará δ') para verificar si se optimizó, es decir, si $\delta < \delta'$**.** De ser así, se conservan tanto la energía δ como los subconjuntos de puntos correspondientes (S, S')**,** utilizados para calcular los coeficientes de la transformación.

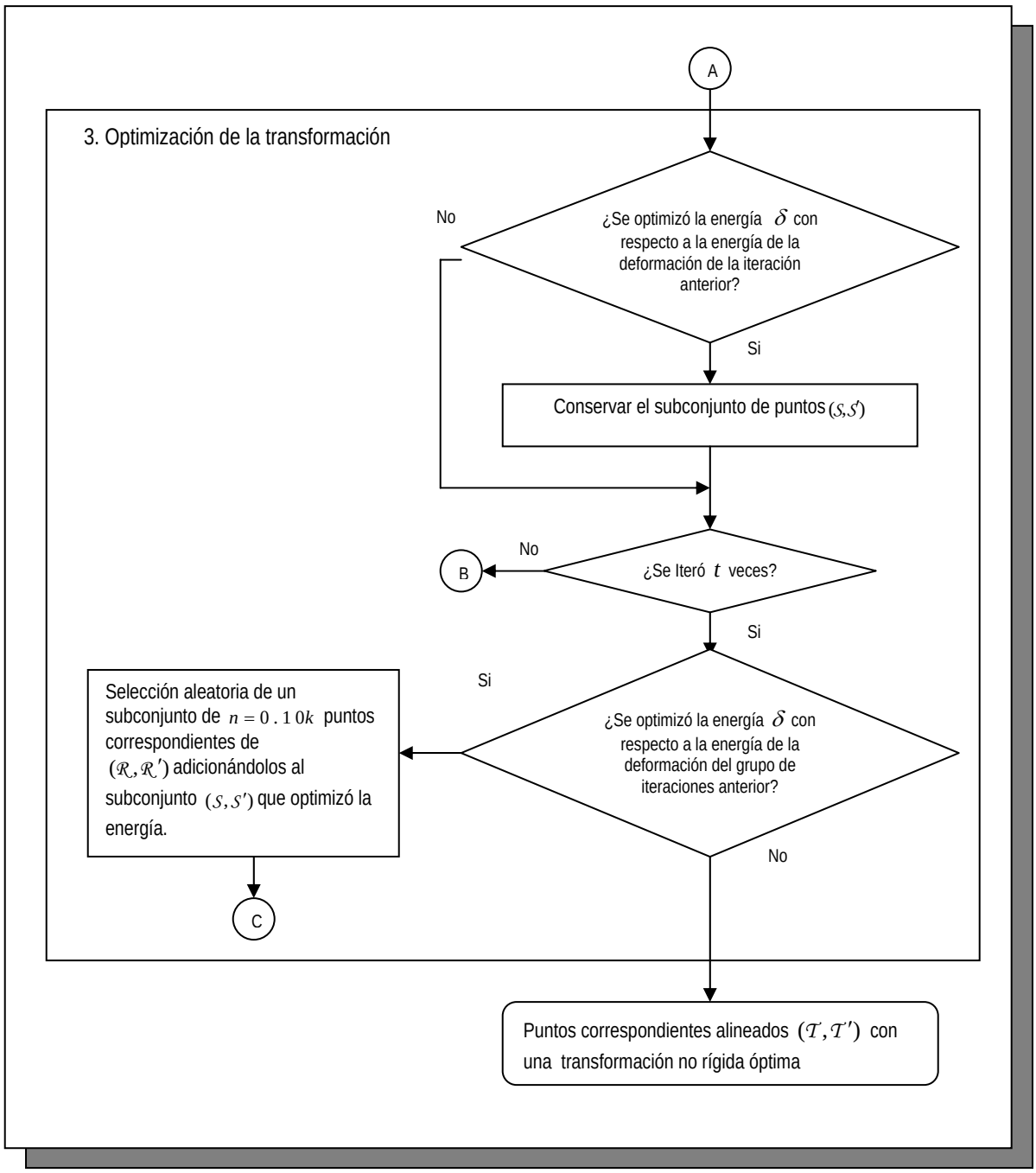
Al finalizar las t iteraciones, se selecciona en forma aleatoria un subconjunto $n = 0.10k$ puntos correspondientes de $(\mathcal{R}, \mathcal{R}')$ los cuales se adicionarán al subconjunto de puntos correspondientes (S, S') conservados en la etapa anterior. Y se repite todo el procedimiento de optimización.

Hay dos condiciones de terminación del procedimiento, la primera es cuando a partir de que se incrementan las muestras no se logra minimizar más δ, y la segunda es cuando se ha tomado el número de muestras equivalente a k.

Una vez terminado el procedimiento anterior, los subconjuntos de puntos correspondientes $(\mathcal{S}, \mathcal{S}')$ se renombran como $\mathcal{T}$, donde $\mathcal{T} = \overline{T}_1, \overline{T}_2, \ldots, \overline{T}_m$ los puntos del subconjunto $\mathcal{S}$ y como $\mathcal{T}' = \overline{T}'_1, \overline{T}'_2, \ldots, \overline{T}'_m$ los puntos del subconjunto $\mathcal{S}'$. Donde m es el número de puntos que finalmente conforman los subconjuntos $(\mathcal{S}, \mathcal{S}')$.

Cuadro 2.4 Algoritmo de RANSAC-TPS




A
3. Optimización de la transformación
No
¿Se optimizó la energía δ con respecto a la energía de la deformación de la iteración anterior?
Si
Conservar el subconjunto de puntos $(\mathcal{S},\mathcal{S}')$
No
B
¿Se Iteró t veces?
Si
Si
Selección aleatoria de un subconjunto de $n = 0.10k$ puntos correspondientes de $(\mathcal{R},\mathcal{R}')$ adicionándolos al subconjunto $(\mathcal{S},\mathcal{S}')$ que optimizó la energía.
¿Se optimizó la energía δ con respecto a la energía de la deformación del grupo de iteraciones anterior?
No
C
Puntos correspondientes alineados $(\mathcal{T},\mathcal{T}')$ con una transformación no rígida óptima

PARTE 2
APORTACIONES

Capítulo 3

Algoritmos de correspondencias y transformaciones

En este capítulo se presentan tres algoritmos para encontrar las correspondencias y las transformaciones óptimas no rígidas entre dos imágenes consecutivas. El primero, llamado ISRANSAC-TPS, tiene el objetivo de disminuir la aleatoriedad de la selección de las muestras y, con esto, obtener mejores resultados que la versión RANSAC-TPS (ver capítulo 2). El segundo, llamado GE-TPS, tiene el objetivo de aplicar la geometría epipolar a los algoritmos RANSAC-TPS e ISRANSAC-TPS para obtener mejores correspondencias. Finalmente, el tercero, llamado GE-TPS-GE-CT, tiene el objetivo de incrementar las correspondencias obtenidas con el GE-TPS.

3.1 Algoritmo ISRANSAC-TPS

El objetivo del algoritmo ISRANSAC-TPS es reducir la aleatoriedad (generada con el RANSAC-TPS) de la selección de las muestras. Esto se logra guiando dicha selección, mediante un muestreo por importancia, el cual se basa en el "comportamiento" de cada correspondencia.

Dicho "comportamiento" se califica con base en la energía de la deformación que genera el grupo de puntos al que pertenece la correspondencia evaluada. A las correspondencias que en promedio tengan el mejor "comportamiento", se les da una mayor probabilidad de ser elegidas, para formar el subconjunto de puntos que permite solucionar el TPS de una forma más económica.

Algoritmo ISRANSAC-TPS

1. *Preproceso*

a. *Selección de un subconjunto de puntos*. Los conjuntos de puntos correspondientes se denotan como $\mathcal{R} = \bar{R}_1, \bar{R}_2, \ldots, \bar{R}_k$ para la primera imagen y $\mathcal{R}' = \bar{R}'_1, \bar{R}'_2, \ldots, \bar{R}'_k$ para la segunda imagen. Estos puntos son coordenadas de píxel, tal que $\bar{R}_i = (\bar{x}_i, \bar{y}_i)$ *y* $\bar{R}'_i = (\bar{x}'_i, \bar{y}'_i)$, con i = 1, 2,…, k, donde k es el número de puntos correspondientes. De los cuales se seleccionan aleatoriamente una muestra de cierto porcentaje (se recomienda que la muestra no sea muy grande, ya que mientras mayor sea, se incrementa la posibilidad de seleccionar correspondencias incorrectas), en el caso de este trabajo, se seleccionó el 30% ($n = 0.30k$). La cual es denotada como $\mathcal{M} = \bar{M}_1, \bar{M}_2, \ldots, \bar{M}_n$ el conjunto de puntos de $\mathcal{R}$ y como $\mathcal{M}' = \bar{M}'_1, \bar{M}'_2, \ldots, \bar{M}'_n$ el conjunto de puntos de $\mathcal{R}'$ correspondientes a ellos.

b. *Generación de grupos de puntos correspondientes.* Los conjuntos de puntos correspondientes $(\mathcal{M}, \mathcal{M}')$, seleccionados en el punto anterior, forman un primer grupo de correspondencias, $\mathcal{G}_1$. De manera similar, se forman q grupos de correspondencias. Un mismo punto de $\mathcal{R}$ puede participar en varios grupos. Donde cada grupo $\mathcal{G}_i = [\,(\bar{M}_1, \bar{M}'_1), (\bar{M}_2, \bar{M}'_2), \ldots, (\bar{M}_k, \bar{M}'_k)\,]$. El objetivo es lograr que las correspondencias participen un mismo número de veces.

c. *Cálculo del "Thin-Plate spline".* Se calcula el TPS (ver en el capítulo 2: Algoritmo RANSAC-TPS, paso 2) para cada grupo $\mathcal{G}_i$, denotando a la energía de la deformación δ_i. Esta se asocia a cada par de puntos de $(\mathcal{M}, \mathcal{M}')$ que conforman el grupo, los cuales, a su vez, están asociados a cada pareja de correspondencias de $(\mathcal{R}, \mathcal{R}')$, de acuerdo con la definición dada anteriormente. Así, la suma de las energías de cada pareja de correspondencias $(\bar{R}_i, \bar{R}'_i)$ de $(\mathcal{R}, \mathcal{R}')$ se denota como $c_i = \delta_1 + \delta_2 + \ldots + \delta_p$, con i = 1, 2,…, p, donde p es el número de grupos en los que participó.

d. *Selección proporcional.*

   i. Se obtiene el promedio de las energías de cada par de correspondencias:

   Para cada $(\bar{R}_i, \bar{R}'_i)$ de $(\mathcal{R}, \mathcal{R}')$ $\qquad e_i = c_i / p$ (3.1)

   ii. Con base en la selección proporcional o de ruleta (ver anexo 3), se obtiene la probabilidad de selección de cada par de correspondencias, proporcional al promedio de sus energías:

   Para cada $(\bar{R}_i, \bar{R}'_i)$ de $(\mathcal{R}, \mathcal{R}')$ $\qquad \theta_i = \sum_{j=1}^{i} (1 - \vartheta(j))$ (3.2)

   Donde

$$\vartheta(i) = \frac{e_i}{\sum_{i=1}^{p} e_i} \tag{3.3}$$

2. *RANSAC-TPS*

Finalmente, se aplica el RANSAC-TPS, de la forma explicada anteriormente (ver capítulo 2: Algoritmo RANSAC-TPS), tomando en cuenta que ahora, la selección del subconjunto de puntos $\mathcal{S}$ y $\mathcal{S}'$ será redefinido de la siguiente forma:

Se denotan como $\mathcal{S} = \bar{S}_1, \bar{S}_2, \ldots, \bar{S}_n$ el conjunto de puntos de $\mathcal{R}$ que cumplan las condiciones i) de que para cada R_i, $\theta_{i-1} < B \leq \theta_i$ y ii) que R_i no se haya elegido anteriormente (pensando en la selección iterativa de muestras del RANSAC-TPS); y como $\mathcal{S}' = \bar{S}'_1, \bar{S}'_2, \ldots, \bar{S}'_n$ el conjunto de puntos de $\mathcal{R}'$ correspondientes a ellos. Donde

$$B = \left(0, \sum_{j=1}^{k} (1 - \vartheta(j)) \right) \tag{3.4}$$

3.1.1 Experimentos

El objetivo de los experimentos que se muestran a continuación es determinar las mejoras o, en su defecto, las desventajas, que ofrece el ISRANSAC-TPS con respecto al RANSAC-TPS, en tres aspectos principales: tiempo de procesamiento, robustez al ruido, correspondencias y transformaciones. Para lo cual se aplicaron las siguientes pruebas a 10 pares de imágenes (par1, par2, ..., par10) de 640×480, que muestran distintos movimientos de la cámara (acercamientos suaves, acercamientos bruscos, translaciones pequeñas, translaciones grandes, etc.).

Prueba 1: Robustez

El objetivo de esta prueba es conocer como afectan las correspondencias incorrectas a los algoritmos. Para ello, los puntos de interés se extrajeron de las imágenes ya sea con el extractor basado en gradiente o con el descriptor SIFT (ver capítulo 1). Los puntos de interés forman un conjunto de puntos $\mathcal{P} = \bar{P}_1, \bar{P}_2, \ldots, \bar{P}_{k_1}$ para la primera imagen y $\mathcal{P}' = \bar{P}'_1, \bar{P}'_2, \ldots, \bar{P}'_{k_2}$ para la segunda imagen. Estos puntos son coordenadas de píxel que se denotan como $\bar{P}_i = (\bar{x}_i, \bar{y}_i)$ con i = 1,2,..., k_1 y $\bar{P}'_j = (\bar{x}'_j, \bar{y}'_j)$ con j = 1,2,..., k_2. Donde k_1 es el número de puntos característicos de la primera imagen y k_2 es el número de puntos característicos de la segunda imagen.

Como los algoritmos requieren puntos correspondientes, se busca que cada $\bar{P}_i$ sea un punto correspondiente a $\bar{P}'_j$. Los datos de salida del ejecutable de SIFT son puntos correspondientes, por lo que solo es necesario hacer corresponder los puntos obtenidos con el extractor basado en gradiente. Para ello, se obtienen los coeficientes de correlación (obtenidos con una función de covarianza), denotados como s_{ij}, para cuantificar la similitud entre cada par $(\bar{P}_i, \bar{P}'_j)$. Si las características tienen una baja relación, es decir, que $s_{ij} \leq s_{\min}$, donde $s_{\min}$ es un umbral que empíricamente se determinó como 0.8 (debido a que se obtienen los mejores resultados), quiere decir que no son correspondientes, en cuyo caso, el valor s_{ij} se reemplaza por cero. Los conjuntos de puntos correspondientes se denotan como $\mathcal{R} = \bar{R}_1, \bar{R}_2, \ldots, \bar{R}_k$ para la primera imagen y $\mathcal{R}' = \bar{R}'_1, \bar{R}'_2, \ldots, \bar{R}'_k$ para la segunda imagen. Estos puntos son coordenadas de píxel, tal que $\bar{R}_i = (\bar{x}_i, \bar{y}_i)$ y $\bar{R}'_i = (\bar{x}'_i, \bar{y}'_i)$, con $i = 1, 2, \ldots, k$, donde k es el número de puntos correspondientes.

Para probar los algoritmos, se experimentó con distintos grados de correspondencias incorrectas. El mejor de los casos es usando solamente SIFT. El cual obtiene correspondencias muy buenas y en mayor cantidad que el extractor basado en gradiente. El caso intermedio es usando los siguientes dos filtros en los datos obtenidos con el extractor basado en gradiente:

- El primer filtro, llamado Filtro de Unicidad [Vincent, E. and Laganière, R., 2001], forza a que cada correspondencia del grupo de puntos $\mathcal{P}$ extraídos en la primera imagen, tenga solo una correspondencia en el grupo de puntos $\mathcal{P}'$ extraídos en la segunda imagen. Para lograrlo, para todo $\bar{P}_i \in \mathcal{P}$ se buscan los dos puntos $\bar{P}'_j \in \mathcal{P}'$ que tengan la correlación s_{ij} más alta con $\bar{P}_i$. La correlación de cada uno se denota como c_1 y c_2 respectivamente, donde c_1 es la correlación más alta y c_2 es la siguiente correlación más alta.

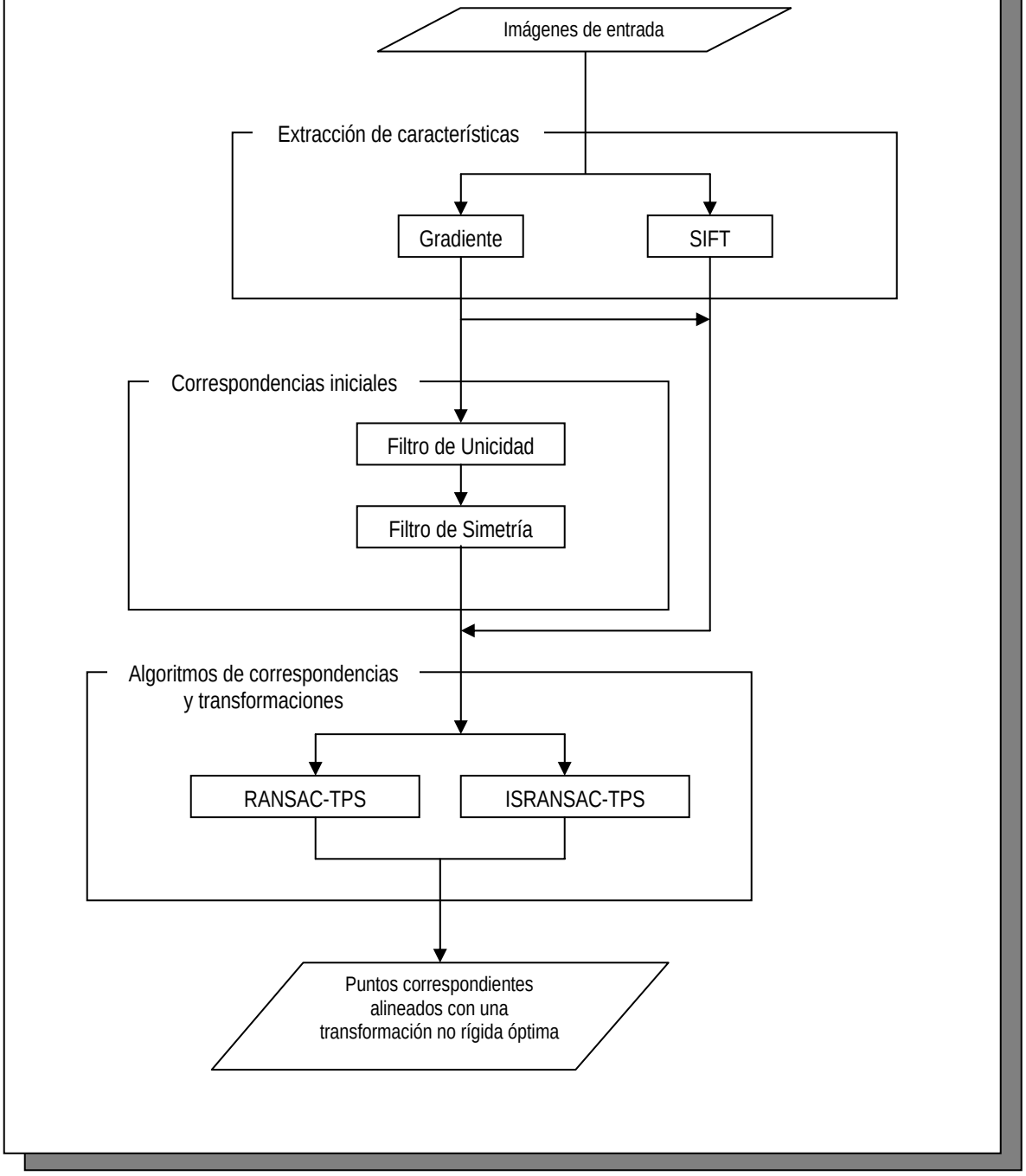
  Si $c_2/c_1 \approx 1$, quiere decir que existe poca diferencia entre ellos por lo que cualquiera de los dos puntos podría ser candidato a ser asociado con $\bar{P}_i$, por lo tanto, se descarta la correspondencia para evitar esa ambigüedad.

- El segundo filtro, llamado Filtro de Simetría [Vincent, E. and Laganière, R., 2001], verifica que el punto encontrado en $\mathcal{P}'$ con la correlación s_{ij} más alta con $\bar{P}_i$, sea el mismo punto que el encontrado en $\mathcal{P}$ con la correlación s_{ij} más alta con $\bar{P}'_j$.

  Finalmente, el peor caso es usar el extractor basado en gradiente pero sin filtros.

El número de iteraciones para minimizar la energía de la deformación se fijó en 100 para todos los casos. Pero, debido al preproceso, este número varía en el caso del ISRANSAC-TPS. El esquema general de los experimentos a los que se sometieron los algoritmos RANSAC-TPS e ISRANSAC-TPS, puede verse en el cuadro 3.1.

Cuadro 3.1 Esquema de experimentos de los algoritmos TPS



a) Caso intermedio. Uso del Filtro de Unicidad y del Filtro de Simetría en los datos obtenidos con el extractor basado en gradiente.

- Transformaciones. En el cuadro 3.2 se presentan los resultados (energía de la deformación δ y tiempo) de aplicar los algoritmos RANSAC-TPS e ISRANSAC-TPS. A partir de los cuales se concluyó que, en la mayoría de los casos (9 de 10), se encontraron mejoras, aunque en algunas poco significativas, en la δ obtenida con el ISRANSAC-TPS. Y que, el tiempo del procesamiento es similar y en algunos casos un poco menor en el RANSAC-TPS.

Par de imágenes	RANSAC-TPS		ISRANSAC-TPS	
	δ	Tiempo	δ	Tiempo
Pair1	264.8356	7.953	92.3763	7.203
Pair2	6320	0.937	3519.3	2.641
Pair3	11200	0.625	8977.3	1.578
Pair4	3840	5.75	1236.9	10.109
Pair5	7810	1.39	188.032	13.922
Pair6	19.5097	0.703	18.2055	4.625
Pair7	523.9792	0.922	496.9456	2.172
Pair8	408.8174	7.797	404.0985	12.203
Pair9	124.8409	1.235	26.7959	2.484
Pair10	1480	1.297	1581.5	8.657

Cuadro 3.2 Muestra los resultados (energía de la deformación δ y tiempo) de aplicar los algoritmos RANSAC-TPS e ISRANSAC-TPS con el Filtro de Unicidad y el Filtro de Simetría.

- Correspondencias. En el cuadro 3.3 se presentan los resultados de un proceso de validación aplicado al RANSAC-TPS y al ISRANSAC-TPS respectivamente, que determinan la proporción de correspondencias finales (Salida-Finales-E%) con respecto a las correspondencias iniciales (Entrada-Iniciales) y más específicamente, la proporción de correspondencias correctas finales con respecto a las correspondencias correctas iniciales (Salida-Correctas-%E); la proporción de correspondencias correctas finales con respecto a las correspondencias finales (Salida-Correctas-%F); la proporción de correspondencias incorrectas finales con respecto a las correspondencias incorrectas iniciales (Salida-Incorrectas-%E) y la proporción de correspondencias incorrectas finales con respecto a las correspondencias finales (Salida-Correctas-%F). El conteo de las correspondencias se obtuvo observando las imágenes y contando manualmente una a una.

  Se puede observar que los resultados son casi los mismos, salvo algunas excepciones, en las que el ISRANSAC-TPS los mejora por poco. También se puede observar que en ambos casos se conservan aproximadamente el 30% de las correspondencias iniciales. Esto debido a que la selección aleatoria de puntos es sobre una muestra del 30%. En algunos casos difíciles de resolver para el RANSAC-TPS (como

Par2 y Par10), donde son pocas las correspondencias finales y no logra dejar pasar correspondencias correctas, el ISRANSAC-TPS las resuelve mejor al incrementar el número de correspondencias correctas. El tiempo del procesamiento es similar y en algunos casos un poco menor en el RANSAC-TPS.

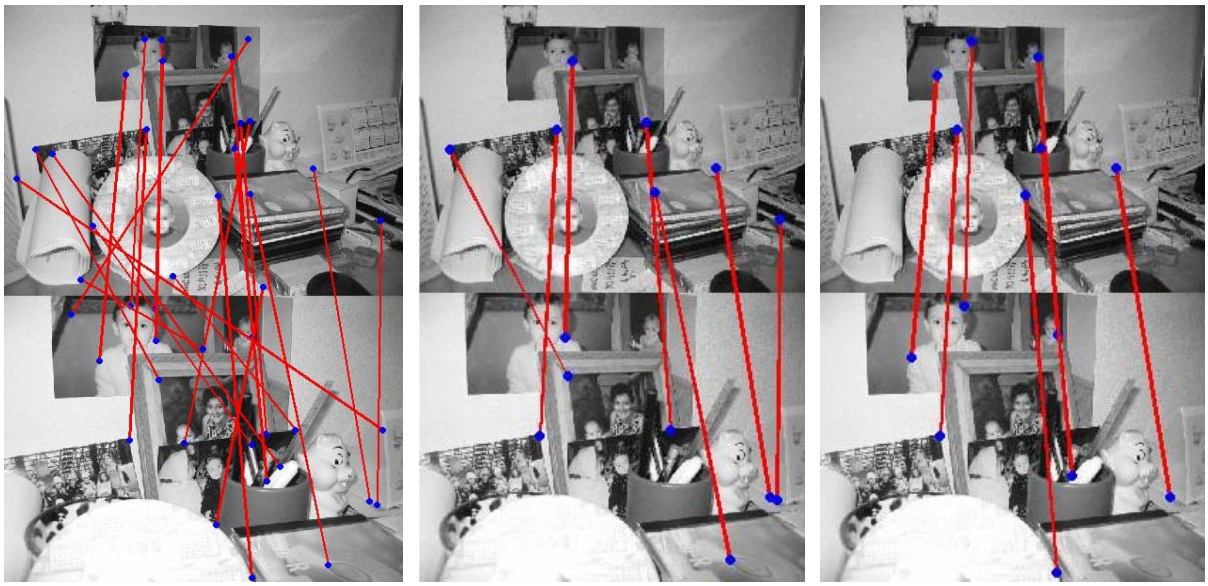
	Entrada					Salida														
								RANSAC-TPS: 100 iteraciones							ISRANSAC-TPS:100 iteraciones+ Extra						
		Correctas		Incorrectas		Finales		Correctas			Incorrectas				Correctas			Incorrectas			
Par de imágenes	Iniciales		% E		% E		% E		% E	% F		% E	% F	Tiempo		% E	% F		% E	% F	Tiempo
Par1	62	54	87	8	13	19	31	19	35	100	0	0	0	7.953	19	35	100	0	0	0	7.203
Par2	21	11	52	10	48	7	33	0	0	0	7	70	100	0.937	6	55	86	1	10	14	2.641
Par3	16	5	31	11	69	5	31	1	20	20	6	55	120	0.625	2	40	40	3	27	60	1.578
Par4	92	83	90	9	9.8	28	30	26	31	93	2	22	7.1	5.75	27	33	96	1	11	3.6	10.109
Par5	42	35	83	7	17	13	31	13	37	100	0	0	0	1.39	13	37	100	0	0	0	13.922
Par6	24	21	88	3	13	8	33	8	38	100	0	0	0	0.703	8	38	100	0	0	0	4.625
Par7	28	18	64	10	36	9	32	8	44	89	1	10	11	0.922	8	44	89	1	10	11	2.172
Par8	107	104	97	3	2.8	33	31	33	32	100	0	0	0	7.797	33	32	100	0	0	0	12.203
Par9	37	32	86	5	14	12	32	12	38	100	0	0	0	1.235	12	38	100	0	0	0	2.484
Par10	29	16	55	13	45	9	31	0	0	0	9	69	100	1.297	8	50	89	1	8	11	8.657

Cuadro 3.3 Muestra las correspondencias obtenidas al aplicar los algoritmos RANSAC-TPS e ISRANSAC-TPS con el Filtro de Unicidad y el Filtro de Simetría. La columna Entrada indica el número de correspondencias iniciales, es decir, aquellas obtenidas después de haber aplicado los filtros. La columna Finales indica el número de correspondencias finales, es decir, aquellas obtenidas después de haber aplicado los algoritmos. La columna %E indica el porcentaje de correspondencias correctas o incorrectas (según sea el caso) con respecto a las correspondencias iniciales. La columna %F indica el porcentaje de correspondencias correctas o incorrectas (según sea el caso) con respecto a las correspondencias finales. La columna Tiempo indica los segundos de procesamiento del algoritmo.

Las figuras 3.1 y 3.2 muestran ejemplos (no necesariamente correspondientes con los datos de las tablas) de los casos de Par2 y Par10, y en las figuras 3.3 y 3.4 se pueden ver otros resultados.

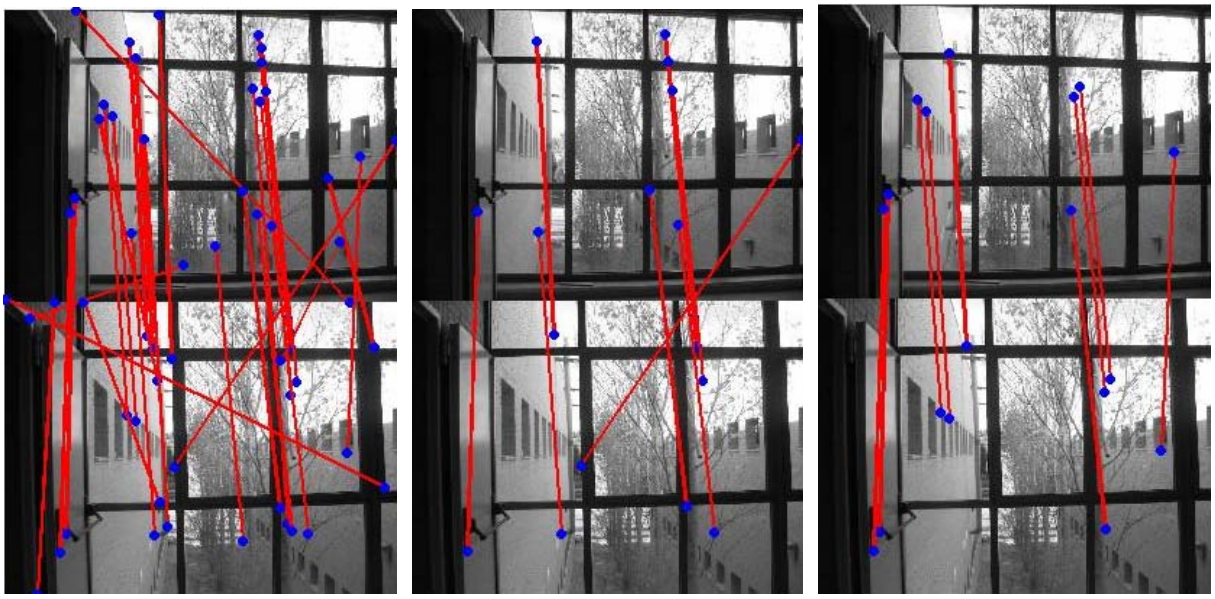
b) El peor caso. Sin filtros en los datos obtenidos con el extractor basado en gradiente

- Transformaciones. En el cuadro 3.4 se presentan los resultados (δ y tiempo) de aplicar los algoritmos RANSAC-TPS e ISRANSAC-TPS. A partir de los cuales se concluyó que, en cualquiera de los algoritmos, la δ es mucho mayor que en el peor de los casos cuando se utilizan los filtros. Esto quiere decir que el algoritmo es sensible a correspondencias incorrectas a pesar del uso de RANSAC. El ISRANSAC-TPS utiliza en algunos casos incluso el doble de iteraciones que el RANSAC-TPS, por lo tanto, es más costoso. Esto se debe a que en la etapa de preprocesamiento se itera, por lo menos, el número de correspondencias iniciales.



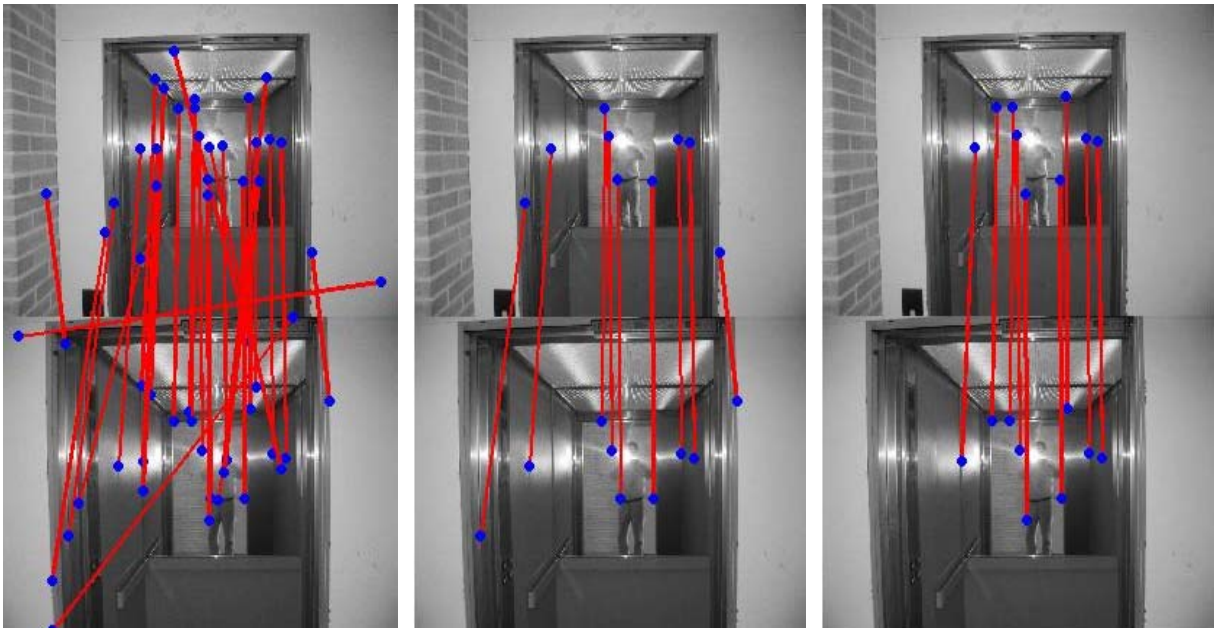
(a) Después de filtros (b) RANSAC-TPS (c) ISRANSAC-TPS

Figura 3.1 Resultados de Par2 (acercamiento y pequeña traslación) al aplicar los algoritmos RANSAC-TPS e ISRANSAC-TPS con el Filtro de Unicidad y el Filtro de Simetría.



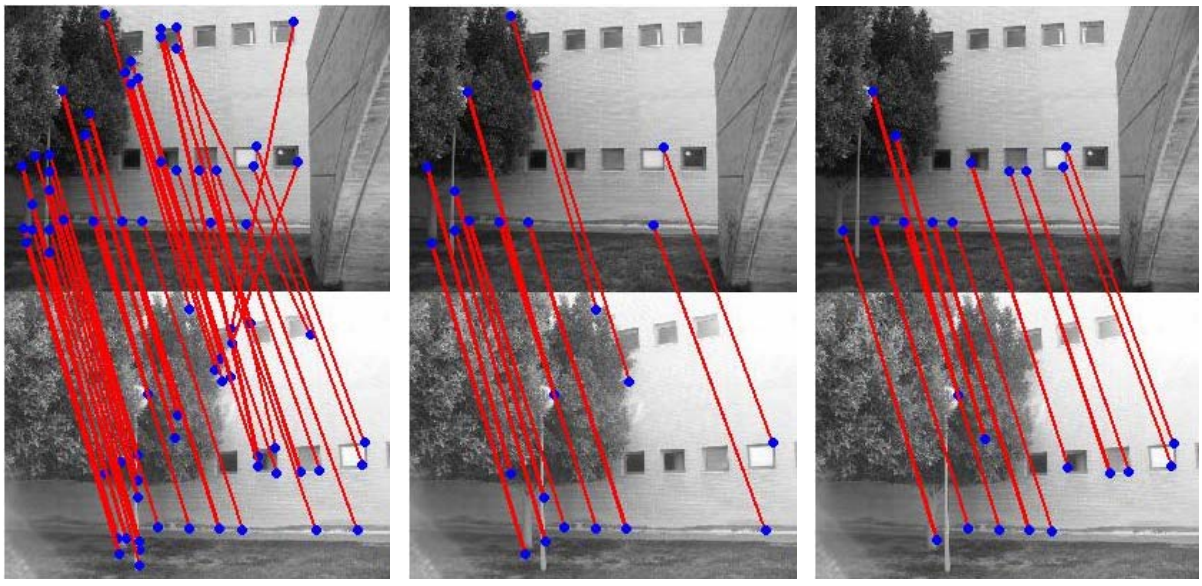
(a) Después de filtros (b) RANSAC-TPS (c) ISRANSAC-TPS

Figura 3.2 Resultados de Par10 (un caso difícil, subir escaleras) al aplicar los algoritmos RANSAC-TPS e ISRANSAC-TPS con el Filtro de Unicidad y el Filtro de Simetría.



(a) Después de filtros (b) RANSAC-TPS (c) ISRANSAC-TPS

Figura 3.3 Resultados de Par7 (pequeño acercamiento) al aplicar los algoritmos RANSAC-TPS e ISRANSAC-TPS con el Filtro de Unicidad y el Filtro de Simetría.



(a) Después de filtros (b) RANSAC-TPS (c) ISRANSAC-TPS

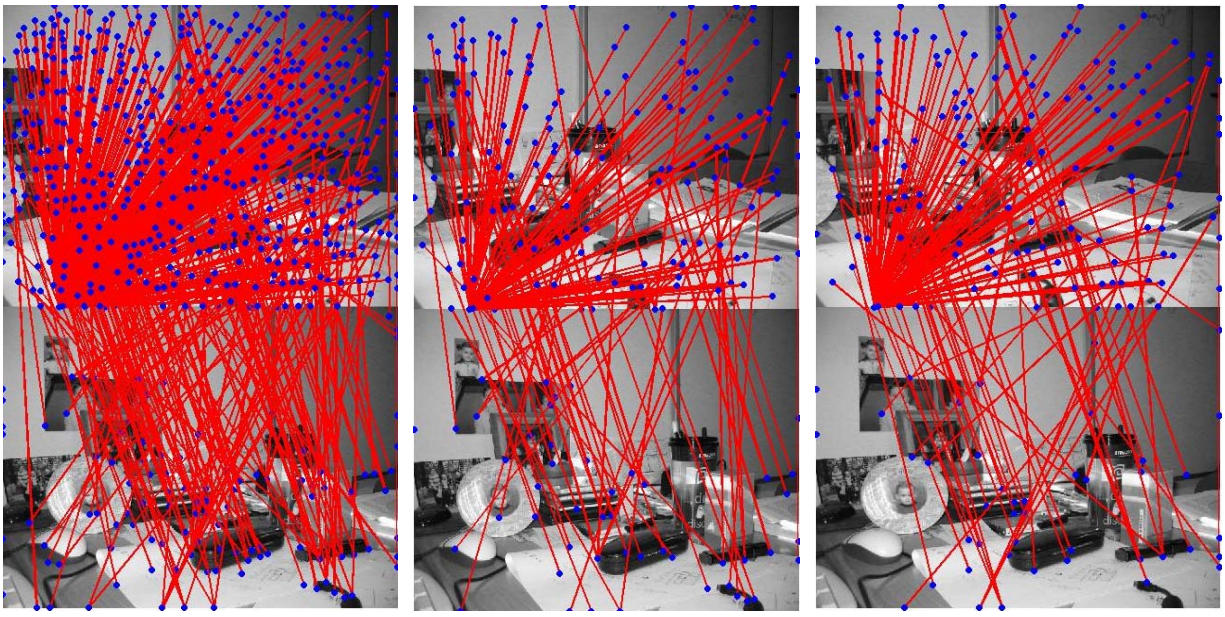
Figura 3.4 Resultados de Par9 (campo de movimiento semidenso) al aplicar los algoritmos RANSAC-TPS e ISRANSAC-TPS con el Filtro de Unicidad y el Filtro de Simetría.

Par de	RANSAC-TPS	ISRANSAC-TPS

imágenes	δ	Tiempo	δ	Tiempo
Pair1	1.50E+07	136.828	1.40E+07	399.391
Pair2	1.80E+07	135.735	2.10E+07	421.359
Pair3	2.40E+07	150	2.10E+07	705.344
Pair4	2.10E+07	139.031	2.30E+07	411.266
Pair5	1.90E+07	152.969	1.80E+07	458.688
Pair6	2.00E+07	185.61	2.10E+07	2436
Pair7	1.80E+07	140.078	2.00E+07	1173.1
Pair8	1.60E+07	196.188	1.90E+07	932.531
Pair9	1.80E+07	162.875	1.80E+07	542.61
Pair10	4.20E+07	177.719	3.70E+07	34975

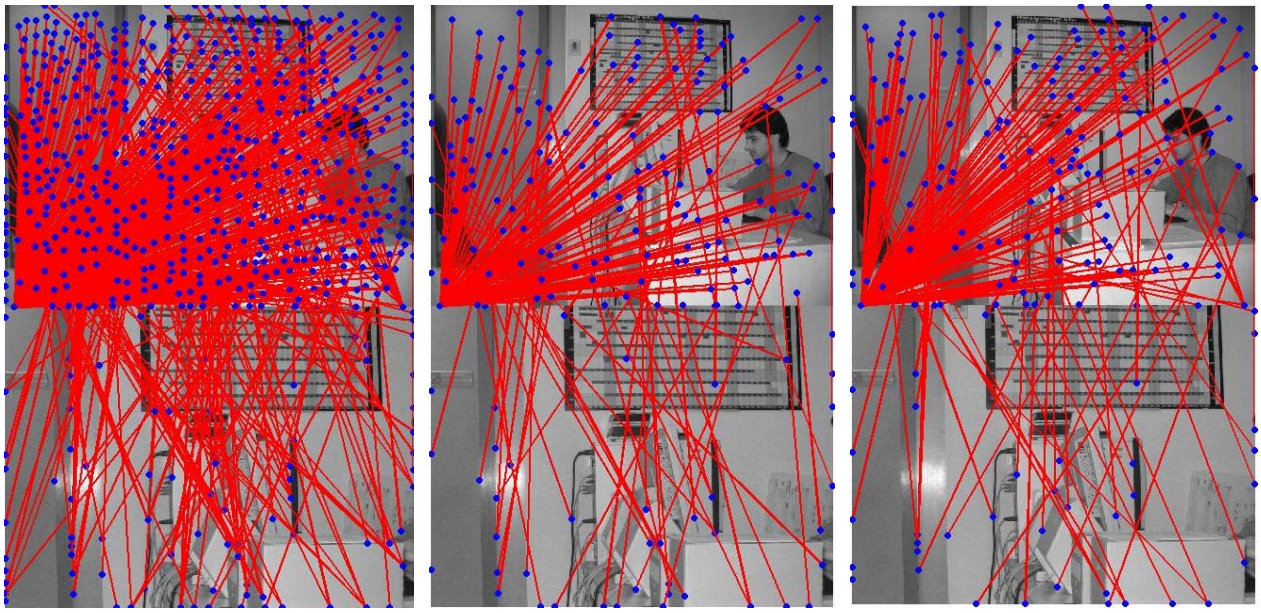
Cuadro 3.4 Muestra los resultados (energía de la deformación δ y tiempo) al aplicar los algoritmos RANSAC-TPS e ISRANSAC-TPS sin filtros.

- Correspondencias. No es posible analizar si existen correspondencias correctas, como se podrá observar en las figuras 3.5 y 3.6. Por lo que se concluye que los filtros son necesarios para disminuir las correspondencias incorrectas y, así, cualquiera de los dos métodos puedan resolver el problema, cada uno con sus respectivas ventajas y desventajas.



(a) Sin filtros (b) RANSAC-TPS (c) ISRANSAC-TPS

Figura 3.5 Resultados de Par1 (pequeña traslación) al aplicar los algoritmos RANSAC-TPS e ISRANSAC-TPS sin filtros.



(a) Sin filtros (b) RANSAC-TPS (c) ISRANSAC-TPS

Figura 3.6 Resultados de Par3 (fuerte acercamiento) al aplicar los algoritmos RANSAC-TPS e ISRANSAC-TPS sin filtros.

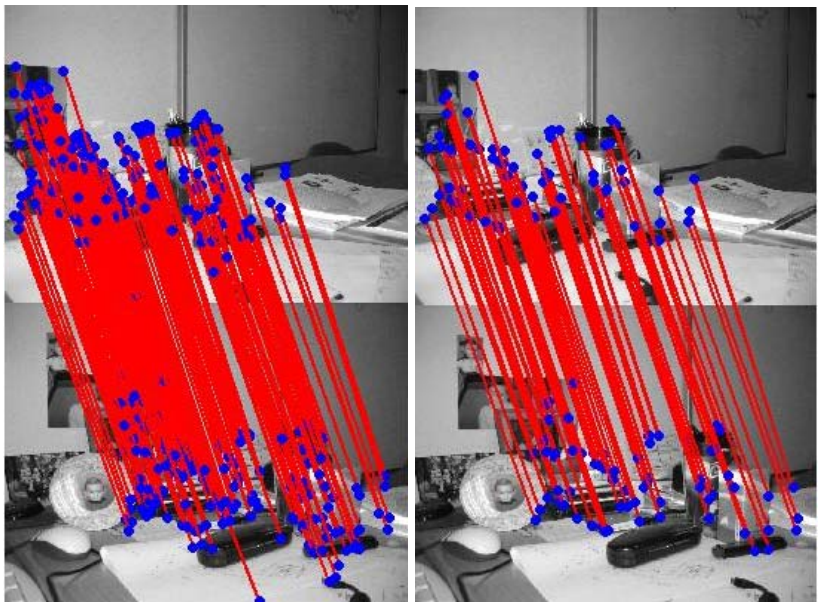
c) El mejor caso. Usando SIFT.

- Transformaciones. En el cuadro 3.5 se presentan los resultados (energía de la deformación δ y tiempo) de aplicar ISRANSAC-TPS. Solo se probó con este algoritmo, ya que, dadas las buenas correspondencias que arroja SIFT, no tiene caso comparar los algoritmos. A partir de los resultados, se concluyó que δ fue mucho mejor en todos los casos, que al utilizar el extractor basado en gradiente, debido a que SIFT es un buen extractor de características que no genera muchas correspondencias incorrectas. Además, con este experimento se pudo conocer el límite aproximado de las características que el algoritmo ISRANSAC-TPS soporta. Por ejemplo, en el Par8, el número de características que SIFT arroja (1545) es muy alto, y no se logró procesar. Esto debido a que el tiempo que tarda el algoritmo depende del número de correspondencias iniciales. Por lo tanto, el algoritmo no es útil para aplicaciones en tiempo real, a menos que se limite la cantidad de correspondencias iniciales, por ejemplo, mediante una selección aleatoria de puntos.

- Correspondencias. Lo más importante que puede observarse en la figura 3.7 es que el ISRANSAC-TPS no conserva todas las correspondencias correctas. Esto se debe a que, por construcción, este algoritmo selecciona un porcentaje determinado de puntos correspondientes, y, si considera que el error de la deformación no puede ser mejorado, no busca incrementar más dichas correspondencias.

Par de imágenes	ISRANSAC-TPS	
	δ	Tiempo
Pair1	482	92.75
Pair2	420.8142	128.984
Pair3	250.2199	77.031
Pair4	209.916	193.438
Pair5	598.9494	386.094
Pair6	122.1045	8.938
Pair7	195.209	2.766
Pair8	-	-
Pair9	127.7725	29.062
Pair10	134.4934	8.812

Cuadro 3.5 Muestra los resultados (energía de la deformación δ y tiempo) de aplicar los algoritmos RANSAC-TPS e ISRANSAC-TPS con SIFT.



(a) Después de filtros (b) ISRANSAC-TPS

Figura 3.7 Resultados de Par1 (pequeña traslación) al aplicar los algoritmos RANSAC-TPS e ISRANSAC-TPS con SIFT.

Prueba 2: Iteraciones

Los algoritmos se sometieron a variaciones en el número de las iteraciones necesarias para la minimización de la energía de la deformación. Esta prueba tiene el objetivo de identificar si la variación de las iteraciones se traduce en una mejora en los resultados.

Los resultados del cuadro 3.6 muestran que al incrementar:

a) De 100 a 5,000 iteraciones, el 100% de los casos mejoró.
b) De 5,000 a 10,000 iteraciones, el 80% de los casos mejoró.
c) De 10,000 a 15,000 iteraciones, el 60% de los casos mejoró.

Los resultados del resto de los casos que no mejoraron, se aproximan a los obtenidos en las 5,000 iteraciones. Con estas cifras se puede concluir que 5,000 son las iteraciones óptimas para tener buenos resultados, a partir de ahí, las mejoras se reducen.

Los resultados del cuadro 3.7 muestran que al incrementar:

a) De 100 a 1,000 iteraciones base, el 80% de los casos mejoró.
b) De 1,000 a 5,000 iteraciones base, el 60% de los casos mejoró. El 20% mejoró las 100 iteraciones y el resto las empeoró.
c) De 5000 a 10,000 iteraciones base, el 60% de los casos mejoró. El 20% mejoró las 100 iteraciones y el resto las empeoró.
d) De 10,000 a 15,000 iteraciones base, el 40% de los casos mejoró. Del 60% restante, el 50% mejoró las 100 iteraciones.

Con estas cifras se puede concluir que 1,000 son las iteraciones óptimas para tener buenos resultados, a partir de ahí, las mejoras se reducen.

Comparando los resultados de los cuadros 3.6 y 3.7 se puede ver que:

a) En 100 iteraciones, ISRANSAC mejora el 90% de los casos.
b) En 1,000 iteraciones, ISRANSAC mejora el 60% de los casos.
c) En 5,000 iteraciones, ISRANSAC mejora el 40% de los casos.
d) En 10,000 iteraciones, ISRANSAC mejora el 30% de los casos.
d) En 15,000 iteraciones, ISRANSAC mejora el 30% de los casos.

Y se puede concluir que el tiempo del procesamiento es similar y en algunos casos menor en el RANSAC-TPS. Esto se debe al preproceso del ISRANSAC-TPS.

Par de imágenes	RANSAC-TPS									
	100		1,000		5,000		10,000		15,000	
	δ	Tiempo	δ	Tiempo	δ	Tiempo	δ	Tiempo	δ	Tiempo
Pair1	264.8356	7.953	139.9126	31.969	78.0135	151.5940	89.7339	160.6870	78.0135	406.0310
Pair2	6320	0.937	1610	8.265	31.0747	90.829	27.3819	150.531	31.0747	256.109
Pair3	11200	0.625	6130	5.406	6129.5	27.5940	6020.4	16.0620	5722	43.0470
Pair4	3840	5.75	2640	57.078	12.1455	573.843	8.2213	1772	3.7681	2857
Pair5	7810	1.39	130.38	14.375	72.5180	73.3590	80.0495	74.0150	40.2189	194.9220
Pair6	19.5097	0.703	9.2511	5.984	7.3289	41.0150	5.7578	62.5780	7.5872	81.1400
Pair7	523.9792	0.922	352.0333	7.265	365.5792	133	347.5515	36.5940	339.2957	99.7970
Pair8	408.8174	7.797	274.0239	78.187	234.1309	393.342	225.5593	729.875	225.8679	1076.9
Pair9	124.8409	1.235	24.0805	11.515	15.8442	58.437	13.014	108.079	16.4092	172.359
Pair10	1480	1.297	1130	12.422	1010	65	957.8017	69.062	957.6413	184.937

Cuadro 3.6 Comparativo de los resultados obtenidos al iterar hasta 15,000 veces al aplicar el algoritmo RANSAC-TPS con el Filtro de Unicidad y el Filtro de Simetría. Para cada imagen se presenta en la columna Tiempo los segundos que tardó el algoritmo en resolver el problema y en la columna δ se muestra la minimización de la energía de la deformación obtenida en cada caso.

Par de imágenes	ISRANSAC-TPS									
	100		1,000		5,000		10,000		15,000	
	δ	Tiempo	δ	Tiempo	δ	Tiempo	δ	Tiempo	δ	Tiempo
Pair1	92.3763	7.203	116.1	40.625	120.8193	46.187	120.0574	46.36	137.9263	42.4070
Pair2	3519.3	2.641	2971.3	7.376	3004.4	9.0630	3217.1	7.8900	3273.5	7.5150
Pair3	8977.3	1.578	6172.8	4.76	9584.3	5.4380	6721.7	5.5320	6316	5.0150
Pair4	1236.9	10.109	2315.2	83.46	676.4700	97.1720	1369.7	86.0310	1170.6	85.8900
Pair5	188.032	13.922	165.656	20.21	58.3050	23.7970	202.3854	23.906	78.5399	20.7970
Pair6	18.2055	4.625	7.269	8.454	8.8231	10.1100	8.0851	9.7970	6.2059	8.9380
Pair7	496.9456	2.172	333.84	10.282	308.9936	13.2350	264.9948	12.8130	365.6224	10.7340
Pair8	404.0985	12.203	290.16	110.26	198.94	547.95	152.93	1082.8	162.38	1697.4
Pair9	26.7959	2.484	16.29	24.74	22.14	85.672	18.39	171.157	19.3785	249.625
Pair10	1581.5	8.657	12.172	1187.4	747.725	55.157	683.007	109.375	710.42	164.265

Cuadro 3.7 Comparativo de los resultados obtenidos al iterar hasta 15,000 veces al aplicar el algoritmo ISRANSAC-TPS con el Filtro de Unicidad y el Filtro de Simetría. Para cada imagen se presenta en la columna Tiempo los segundos que tardó el algoritmo en resolver el problema y en la columna δ se muestra la minimización de la energía de la deformación obtenida en cada caso.

3.2 Algoritmo GE-TPS

Como se mencionó anteriormente, la geometría epipolar se puede representar algebraicamente por la matriz fundamental **F** (ver capítulo 2), la cual puede ser calculada a partir de un conjunto de puntos.

Esto hace que no sea una herramienta muy robusta en si misma, porque se trata de representar la geometría de todos los puntos de una imagen, a partir de unos cuantos. Los cuales pueden tener correspondencias tanto correctas como

incorrectas, y, por lo tanto, la aproximación de la geometría epipolar puede ser muy mala. Este problema se soluciona, en gran medida, agregando algoritmos de muestreo aleatorio, como el RANSAC (ver apéndice 2) o Mínimos cuadrados (ortogonales u ordinarios) [Torr, P. and Murray, D.W., 1997]. Sin embargo, para hacer más económico el algoritmo, no se utilizó ninguno de ellos.

Con el objetivo, no solo de mejorar las correspondencias obtenidas con la GE, sino de obtener las transformaciones, se decidió unir la GE con los algoritmos basados en TPS, de manera que se explotaran las ventajas de cada uno. El algoritmo resultante se nombró GE-TPS.

Algoritmo GE-TPS

---

1. *Geometría epipolar*
   a. ***Selección de un subconjunto de puntos***. Los conjuntos de puntos correspondientes se denotan como $\mathcal{R} = \bar{R}_1, \bar{R}_2, \ldots, \bar{R}_k$ para la primera imagen y $\mathcal{R}' = \bar{R}'_1, \bar{R}'_2, \ldots, \bar{R}'_k$ para la segunda imagen. Donde k es el número de puntos correspondientes. De los cuales se seleccionan aleatoriamente 8 puntos, denotados como $\mathcal{S} = \bar{S}_1, \bar{S}_2, \ldots, \bar{S}_8$ el conjunto de puntos de $\mathcal{R}$ y como $\mathcal{S}' = \bar{S}'_1, \bar{S}'_2, \ldots, \bar{S}'_8$ el conjunto de puntos de $\mathcal{R}'$ correspondientes a ellos.

   b. Obtención de la matriz fundamental $\mathbf{F}$. Para este proyecto se utiliza una solución directa, ya que solo se cuenta con puntos correspondientes y no se tienen los parámetros de la cámara. Específicamente, el algoritmo normalizado de 8 puntos (ver cuadro 2.1) debido a que es lineal, suficientemente preciso y fácil de implementar [Hartley, R. and Zisserman, A., 2004].

   c. Aplicación de la restricción epipolar. A cada par de correspondencias iniciales $\left(\bar{R}_i, \bar{R}'_i\right)$ se le aplica la restricción epipolar (2.24):

$$\bar{R}_i'^T \mathbf{F} \bar{R}_i \leq u \qquad (3.5)$$

   Donde u es un escalar positivo (fijado como 0.09 con base en los resultados de los experimentos) que mientras más se aproxime a cero, más rígido es en la selección de las correspondencias y mientras más grande sea, la restricción se vuelve muy flexible, incrementando el paso de correspondencias incorrectas.

Las correspondencias que cumplen con esta restricción se denotan como $\mathcal{G} = \bar{G}_1, \bar{G}_2, \ldots, \bar{G}_m$ y como $\mathcal{G}' = \bar{G}'_1, \bar{G}'_2, \ldots, \bar{G}'_m$. Donde *m* es el número de correspondencias que cumplen con la restricción epipolar. Si no se encontraron puntos que cumplan esta restricción, se repiten los pasos anteriores.

2. *RANSAC-TPS ó ISRANSAC-TPS.* Las correspondencias $(\mathcal{G}, \mathcal{G}')$ son depuradas con el RANSAC-TPS o con el ISRANSAC-TPS con el objetivo de obtener mejores correspondencias y calcular las transformaciones.

   El conjunto de las correspondencias resultantes serán denotadas como $\mathcal{T} = \bar{T}_1, \bar{T}_2, \ldots, \bar{T}_r$ para la primera imagen y como $\mathcal{T}' = \bar{T}'_1, \bar{T}'_2, \ldots, \bar{T}'_r$ para la segunda imagen. Donde *r* es el número de correspondencias resultantes después de aplicar el algoritmo.

3.2.1 Experimentos

El objetivo de los experimentos que se muestran a continuación, es determinar las mejoras o, en su defecto, las desventajas, que ofrece el GE-TPS con respecto a la GE, a los algoritmos de TPS y a un algoritmo de correspondencias basado en grafos, en tres aspectos principales: tiempo de procesamiento, robustez al ruido y correspondencias.

Prueba 1: Robustez

Esta prueba se aplicó a 10 pares de imágenes (par1, par2, …, par10) de 640×480 **,** que muestran distintos movimientos de la cámara (acercamientos suaves, acercamientos bruscos, translaciones pequeñas, translaciones grandes, etc.). Es la misma prueba que la aplicada en el experimento de la sección anterior (3.1.1). Excepto por que no se probará el mejor caso (el cual usa el descriptor SIFT). Esto es porque, por un lado, la complejidad de la GE no depende del número de correspondencias iniciales, que es uno de los posibles resultados que se podrían analizar, y, por el otro, lo deseable es probar casos con correspondencias incorrectas para conocer la robustez del algoritmo.

Para el algoritmo GE-TPS se decidió utilizar el RANSAC-TPS porque es más económico, y la aplicación que se plantea en esta tesis, es en tiempo real.

a) Caso intermedio*.* Uso del Filtro de Unicidad y del Filtro de Simetría en los datos obtenidos con el extactor basado en gradiente.

- Transformaciones. En el cuadro 3.8 se presentan los resultados (energía de la deformación δ y tiempo) de aplicar el algoritmo GE-TPS con 100 iteraciones. Comparándolos con los resultados mostrados en el cuadro 3.6, se concluyó que, en todos los casos con 100 iteraciones y en el 70% de los casos con 5000 iteraciones, se encontraron mejoras en la energía de la deformación usando el GE-TPS.

Par de imágenes	GE-TPS	
	δ	Tiempo
Pair1	123.8904	2.875
Pair2	0	2.016
Pair3	831.0199	0.968
Pair4	43.7235	2.109
Pair5	23.4358	1.641
Pair6	3.4916	0.984
Pair7	101.1231	0.891
Pair8	271.7048	8.625
Pair9	14.4167	1.578
Pair10	1.2537	0.766

Cuadro 3.8 Muestra los resultados (energía de la deformación δ y tiempo) al aplicar el algoritmo GE-TPS con el Filtro de Unicidad y el Filtro de Simetría.

- Correspondencias. El cuadro 3.9 presenta los resultados de un proceso de validación aplicado al algoritmo GE y al GE-TPS.

  Comparando el cuadro 3.3, que muestra los resultados de los algoritmos de TPS (RANSAC-TPS e ISRANSAC-TPS), contra el 3.9, se puede observar que, en todos los casos, el tiempo es menor en el algoritmo de GE. Su ventaja principal, es que deja pasar muchas correspondencias correctas. La desventaja es que también pasan muchas incorrectas. Mientras que en los algoritmos basados en TPS ocurre lo contrario, dejan pasar pocas correspondencias incorrectas, aunque también pocas correctas. Por otro lado, el ISRANSAC-TPS es más caro que cualquiera de los anteriores, aunque las correspondencias sean mejores en la mayoría de los casos. El GE-TPS logró una mejora sobre los resultados del GE y RANSAC-TPS y sobre el tiempo del ISRANSAC-TPS. Pero también disminuyó el número de correspondencias correctas, lo que aún sigue siendo un problema. La ventaja hasta este punto es que se tiene un algoritmo no solo con transformaciones, sino con buenas correspondencias. Las figuras 3.8-3.11 muestran algunos resultados.

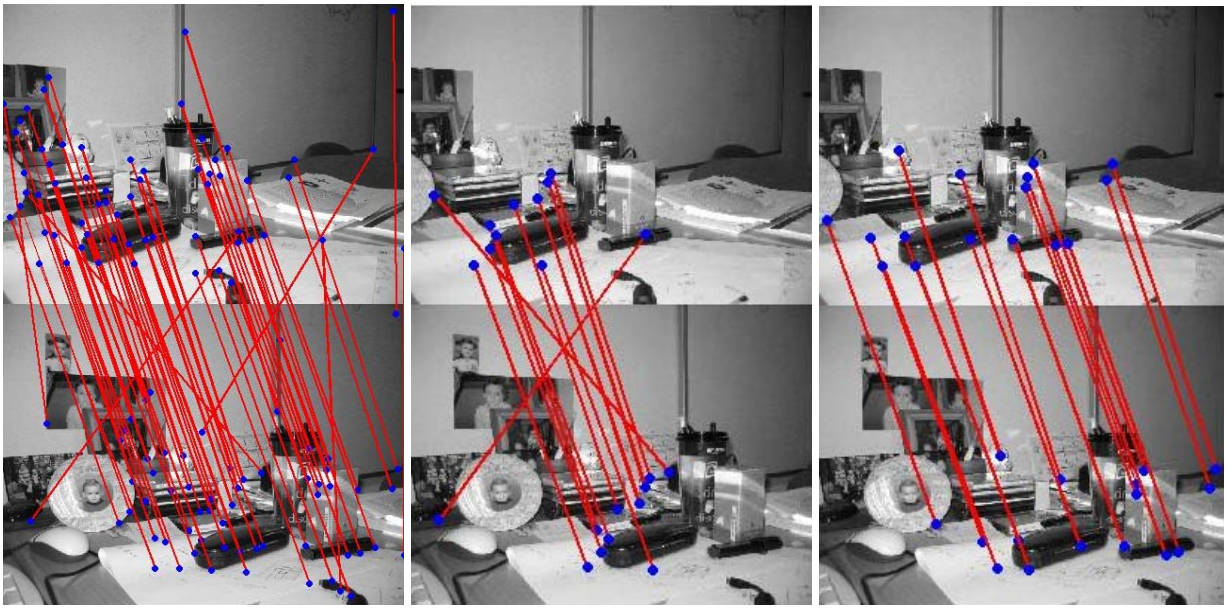
	Entrada					Salida																
								GE									GE-TPS					
		Correctas		Incorrectas		Finales		Correctas			Incorrectas				Finales		Correctas			Incorrectas			
Par de imágenes	Iniciales		% E		% E		% E		% E	% F		% E	% F	Tiempo		% E		% E	% F		% E	% F	Tiempo
Par1	62	54	87	8	13	45	73	45	83	100	0	0	0	0.015	16	26	16	30	100	0	0	0	2.313
Par2	21	11	52	10	48	8	38	4	36	50	4	40	50	0	4	19	4	36	100	0	0	0	0.301
Par3	16	5	31	11	69	8	50	3	60	38	5	45	63	0.015	3	19	2	40	67	1	9	33	0.25
Par4	92	83	90	9	9.8	52	57	47	57	90	5	56	10	0.016	22	24	22	27	100	0	0	0	4.546
Par5	42	35	83	7	17	32	76	30	86	94	2	29	6	0.062	6	14	6	17	100	0	0	0	0.547
Par6	24	21	88	3	13	21	88	21	100	100	0	0	0	0	7	29	7	33	100	0	0	0	0.719
Par7	28	18	64	10	36	6	21	5	28	83	1	10	17	0	6	21	6	33	100	1	10	17	0.578
Par8	107	104	97	3	2.8	100	93	99	95	99	1	33	1	0.016	32	30	32	31	100	0	0	0	8.89
Par9	37	32	86	5	14	15	41	15	47	100	0	0	0	0	9	24	9	28	100	0	0	0	0.938
Par10	29	16	55	13	45	19	66	14	88	74	5	38	26	0.016	4	14	3	19	75	1	8	25	0.313

Cuadro 3.9 Muestra las correspondencias obtenidas al aplicar los algoritmos GE y GE-TPS con el Filtro de Unicidad y el Filtro de Simetría. La columna Entrada indica el número de correspondencias iniciales, es decir, aquellas obtenidas después de haberles aplicado los filtros. La columna Finales indica el número de correspondencias finales, es decir, aquellas obtenidas después de haberles aplicado los algoritmos La columna %E indica el porcentaje de correspondencias correctas o incorrectas (según sea el caso) con respecto a las correspondencias iniciales. La columna %F indica el porcentaje de correspondencias correctas o incorrectas (según sea el caso) con respecto a las correspondencias finales. La columna Tiempo indica los segundos de procesamiento del algoritmo.

b) El peor caso. Sin filtros en los datos obtenidos con el extractor basado en gradiente

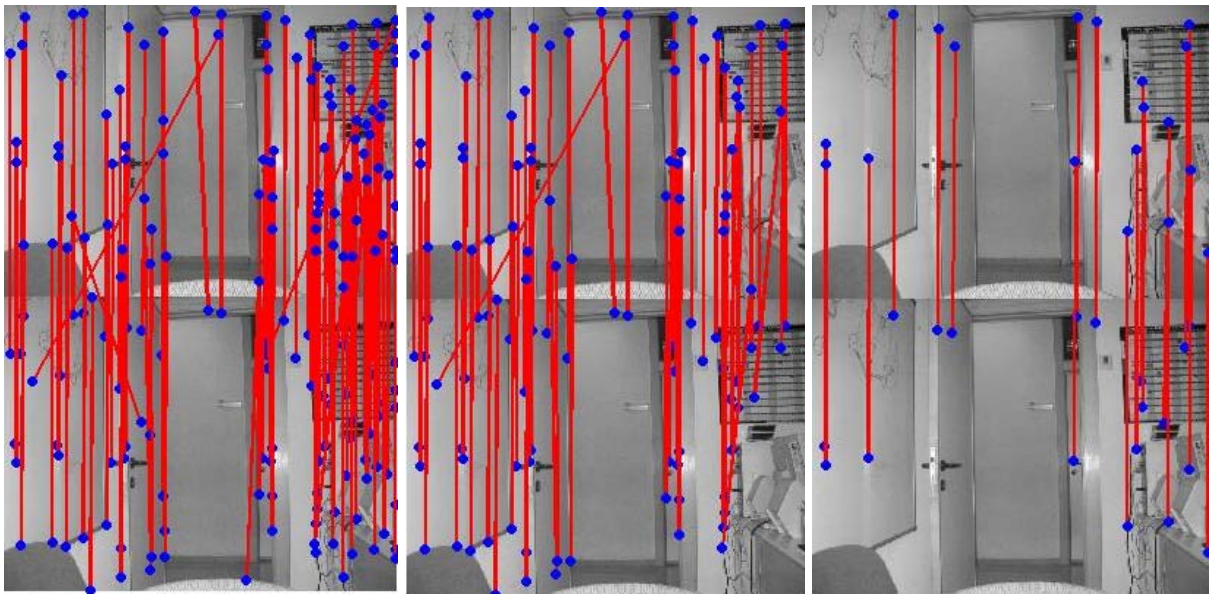
No se puede evitar usar filtros para los algoritmos basados en GE. Esto se debe a que la matriz $\mathbf{A}$ (ver capítulo 2) debe tener a lo más rango 8 para poder utilizar el algoritmo normalizado de 8 puntos. Si los datos no son exactos, porque hay ruido en los puntos correspondientes, entonces el rango de $\mathbf{A}$ puede ser mayor que 8 y, por lo tanto, este algoritmo no funcionaría.

Además, uno de los objetivos, es hacer más económico el algoritmo GE-TPS, por eso no se utilizó ningún método de muestreo aleatorio. Pero si se necesita eliminar la mayor cantidad de correspondencias incorrectas posibles. Por lo menos, aplicando el Filtro de Unicidad, ya que este evita las correspondencias de una a muchas, lo cual genera el problema del rango mencionado anteriormente.



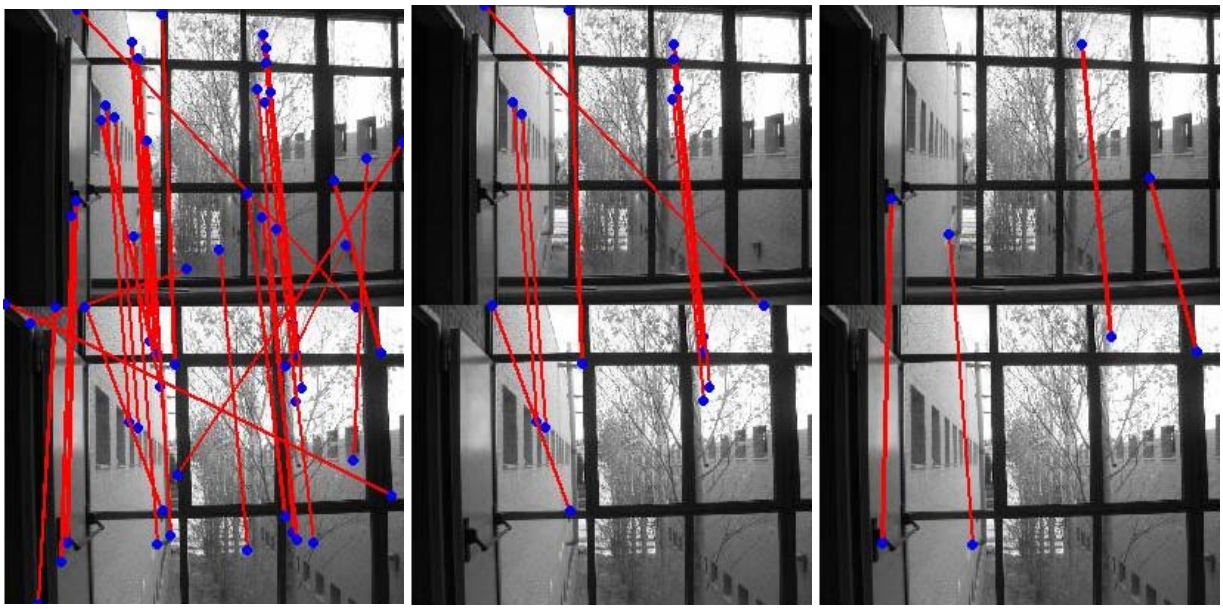
(a) Después de filtros (b) GE (c) GE-TPS

Figura 3.8 Resultados de Par1 (pequeña traslación) al aplicar los algoritmos GE y GE-TPS con el Filtro de Unicidad y el Filtro de Simetría.



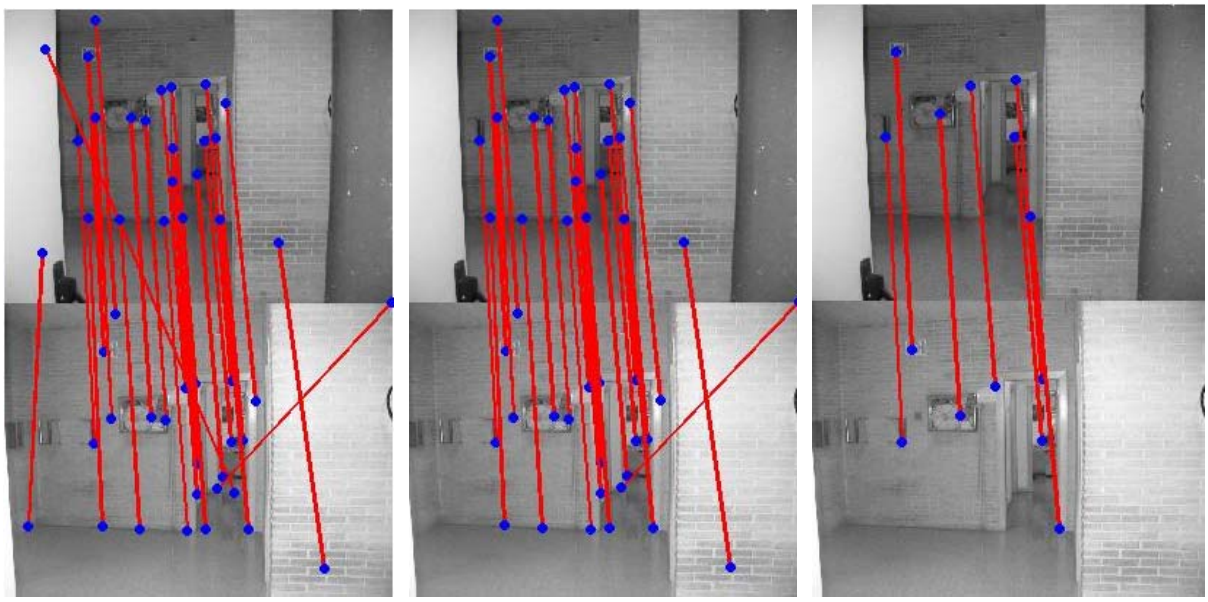
(a) Después de filtros (b) GE (c) GE-TPS

Figura 3.9 Resultados de Par4 (pequeño acercamiento) al aplicar los algoritmos GE y GE-TPS con el Filtro de Unicidad y el Filtro de Simetría.



(a) Después de filtros (b) GE (c) GE-TPS

Figura 3.10 Resultados de Par10 (un caso difícil, subir escaleras) al aplicar los algoritmos GE y GE-TPS con el Filtro de Unicidad y el Filtro de Simetría.



(a) Después de filtros (b) GE (c) GE-TPS

Figura 3.11 Resultados de Par6 (traslación) al aplicar los algoritmos GE y GE-TPS con el Filtro de Unicidad y el Filtro de Simetría.

Prueba 2: Comparación contra el algoritmo de Transformación de Grafos

Con el objetivo de poder comparar el desempeño del algoritmo GE-TPS contra otras técnicas, se decidió utilizar el algoritmo de Transformación de Grafos (TG) presentado en [Aguilar, W. et al., 2006], el cual fue diseñado con el objetivo de reconocer objetos en una escena. Consiste en extraer características tanto de la imagen de la escena como de la imagen del objeto a reconocer, para luego hacer un emparejamiento estructural entre las características de ambas imágenes.

Las estructuras son grafos que se forman con las características de cada imagen, que, al irlos transformando simultáneamente, al final del algoritmo se conservan solo las correspondencias correctas.

En la fase de extracción de características puede utilizar cualquier extractor, que puede ir desde uno muy simple basado en gradiente hasta uno mucho más sofisticado como SIFT. Una vez que se obtienen las características de ambas imágenes, se les aplica un algoritmo que realiza una comparación de los descriptores y obtiene una correspondencia inicial.

El algoritmo tiene un límite máximo de hasta 200 correspondencias iniciales para que pueda funcionar en tiempos razonables, las cuales son suficientes para realizar el reconocimiento de objetos. Es bueno trabajando con oclusiones porque el algoritmo permite que el grafo sea disconexo, con lo que, aunque las características se encuentren acumuladas en grupos espacialmente lejanos, se puede hacer una buena correspondencia.

La prueba se dividió en dos secciones principales, dependiendo del tipo de deformaciones que presentan las imágenes: deformaciones suaves y deformaciones profundas.

Las deformaciones suaves se pueden entender como transformaciones que llevan de una imagen a otra de una manera relativamente fluida, como sería el caso de una secuencia de imágenes que representan el andar de un robot o de una persona, o el caso de imágenes médicas de fondo de ojos, por nombrar algunas.

Mientras que las deformaciones profundas se pueden entender como movimientos bruscos entre una imagen y otra, por ejemplo, el tipo de imágenes utilizadas en aplicaciones como el reconocimiento de objetos en escenas, donde la imagen del objeto está fuera del contexto de la imagen que representa la escena en la que tiene que ser buscado.

a) Deformaciones suaves. Se hicieron experimentos con 2 pares de imágenes (par1, par8) que muestran distintos movimientos de la cámara. A cada par de imágenes se les aplicó una extracción de características basada en el gradiente.

   i. Caso intermedio. Uso del Filtro de Unicidad y del Filtro de Simetría en los datos obtenidos con el extractor basado en gradiente.

El cuadro 3.10 presenta los resultados de un proceso de validación aplicado al TG y al GE-TPS, a partir del cual se puede observar que el TG deja pasar más correspondencias correctas que el GE-TPS y es más rápido.

	Entrada					Salida																			
								TG									GE-TPS								
		Correctas		Incorrectas		Finales		Correctas			Incorrectas				Finales		Correctas			Incorrectas				
Par de imágenes	Iniciales		% E		% E		% E		% E	% F		% E	% F	Tiempo		% E		% E	% F		% E	% F	Tiempo
Par1	62	54	87	8	13	38	61	38	70	100	0	0	0	0.09	17	27	17	31	100	0	0	0	2.12
Par8	107	104	97	3	3	75	70	75	72	100	0	0	0	0.25	27	25	27	26	100	0	0	0	4.9

Cuadro 3.10 Muestra las correspondencias obtenidas al aplicar los algoritmos TG y GE-TPS con Filtro de Unicidad y Filtro de Simetría en imágenes suaves. La columna Entrada indica el número de correspondencias iniciales, es decir, aquellas obtenidas después de haberles aplicado los filtros. La columna Finales indica el número de correspondencias finales, es decir, aquellas obtenidas después de haberles aplicado los algoritmos. La columna %E indica el porcentaje de correspondencias correctas o incorrectas (según sea el caso) con respecto a las correspondencias iniciales. La columna %F indica el porcentaje de correspondencias correctas o incorrectas (según sea el caso) con respecto a las correspondencias finales. La columna Tiempo indica los segundos de procesamiento del algoritmo.

ii. *El peor caso.* Uso del Filtro de Unicidad en los datos obtenidos con el extractor basado en gradiente.

El Filtro de Unicidad elimina la relación una a muchas correspondencias en la etapa de correspondencias iniciales de los algoritmos. El objetivo de este experimento es ver como funcionan los algoritmos con un alto porcentaje de correspondencias iniciales incorrectas.

El cuadro 3.11 presenta los resultados de un proceso de validación aplicado al TG y al GE-TPS, a partir del cual se puede observar que el TG deja pasar más correspondencias correctas que el GE-TPS, y con menos errores en algunos casos, el TG es más rápido y puede trabajar sin filtros.

	Entrada					Salida																			
								TG									GE-TPS								
		Correctas		Incorrectas		Finales		Correctas			Incorrectas				Finales		Correctas			Incorrectas				
Par de imágenes	Iniciales		% E		% E		% E		% E	% F		% E	% F	Tiempo		% E		% E	% F		% E	% F	Tiempo
Par1	120	55	46	65	54	42	35	40	73	95	2	3	5	0.42	11	9	6	11	55	5	8	45	1.23
Par8	143	103	72	40	28	76	53	74	72	97	2	5	3	0.58	34	24	31	30	91	3	8	9	7.75

Cuadro 3.11 Muestra las correspondencias obtenidas al aplicar los algoritmos TG-GE y GE-TPS con Filtro de Unicidad en imágenes suaves. La columna Entrada indica el número de correspondencias iniciales, es decir, aquellas obtenidas después de haberles aplicado los filtros. La columna Finales indica el número de correspondencias finales, es decir, aquellas obtenidas después de haberles aplicado los algoritmos. La columna %E indica el porcentaje de correspondencias correctas o incorrectas (según sea el caso) con respecto a las

correspondencias iniciales. La columna %F indica el porcentaje de correspondencias correctas o incorrectas (según sea el caso) con respecto a las correspondencias finales. La columna Tiempo indica los segundos de procesamiento del algoritmo.

b) Deformaciones profundas. Se hicieron experimentos con 2 pares de imágenes (Póster y Lab) a los cuales se les extrajeron las características con SIFT (ver capítulo 2). A continuación se presentan los resultados de aplicar los algoritmos a imágenes que presentan deformaciones profundas. Hay que tomar en cuenta que, en general, los algoritmos de transformaciones no pueden representar transformaciones globales en los casos donde las variaciones entre las imágenes son profundas y por lo tanto, aunque resuelven las correspondencias, no necesariamente dan tan buenos resultados.

El cuadro 3.12 presenta los resultados de un proceso de validación aplicado al TG y al GE-TPS respectivamente, a partir del cual se observa que el TG, como en los casos anteriores, deja pasar más correspondencias correctas que el GE-TPS y es más rápido. En la figura 3.12 se muestran los resultados de aplicar ambos algoritmos a la imagen Póster.

	Entrada					Salida																		
								TG									GE-TPS							
		Correctas		Incorrectas		Finales		Correctas			Incorrectas				Finales		Correctas			Incorrectas				
Par de imágenes	Iniciales		% E		% E		% E		% E	% F		% E	% F	Tiempo		% E		% E	% F		% E	% F	Tiempo
Póster	291	247	85	44	15	170	58	170	69	100	0	0	0	1.15	51	18	50	20	98	1	2	2	28.8
Lab	56	50	89	6	11	35	63	35	70	100	0	0	0	0.06	15	27	15	30	100	0	0	0	2.81

Cuadro 3.12 Muestra las correspondencias obtenidas al aplicar los algoritmos TG y GE-TPS con SIFT en imágenes profundas. La columna Entrada indica el número de correspondencias iniciales, es decir, aquellas obtenidas después de haberles aplicado los filtros. La columna Finales indica el número de correspondencias finales, es decir, aquellas obtenidas después de haberles aplicado los algoritmos. La columna %E indica el porcentaje de correspondencias correctas o incorrectas (según sea el caso) con respecto a las correspondencias iniciales. La columna %F indica el porcentaje de correspondencias correctas o incorrectas (según sea el caso) con respecto a las correspondencias finales. La columna Tiempo indica los segundos de procesamiento del algoritmo.

3.3 Algoritmo GE-TPS-GE-CT

El algoritmo GE-TPS, a pesar de que su ventaja es obtener buenas correspondencias en poco tiempo, recupera pocas de ellas. Para solucionar este problema, se aprovecha dicha ventaja para obtener una matriz fundamental que represente mejor la geometría de todos los puntos y, con esta matriz, filtrar los datos de entrada. Esta idea con base en el esquema que se ha manejado en la literatura, como en [Roth, G. and Whitehead, A., 2000], [Torr, P. and Zisserman, A., 1998] y [Zhang, Z. et al., 1995], donde los pasos generales a seguir son:

1. Se detectan los puntos correspondientes y se aplica la correlación entre ellos

2. Se aplica un umbral para obtener un primer conjunto de correspondencias candidatas.

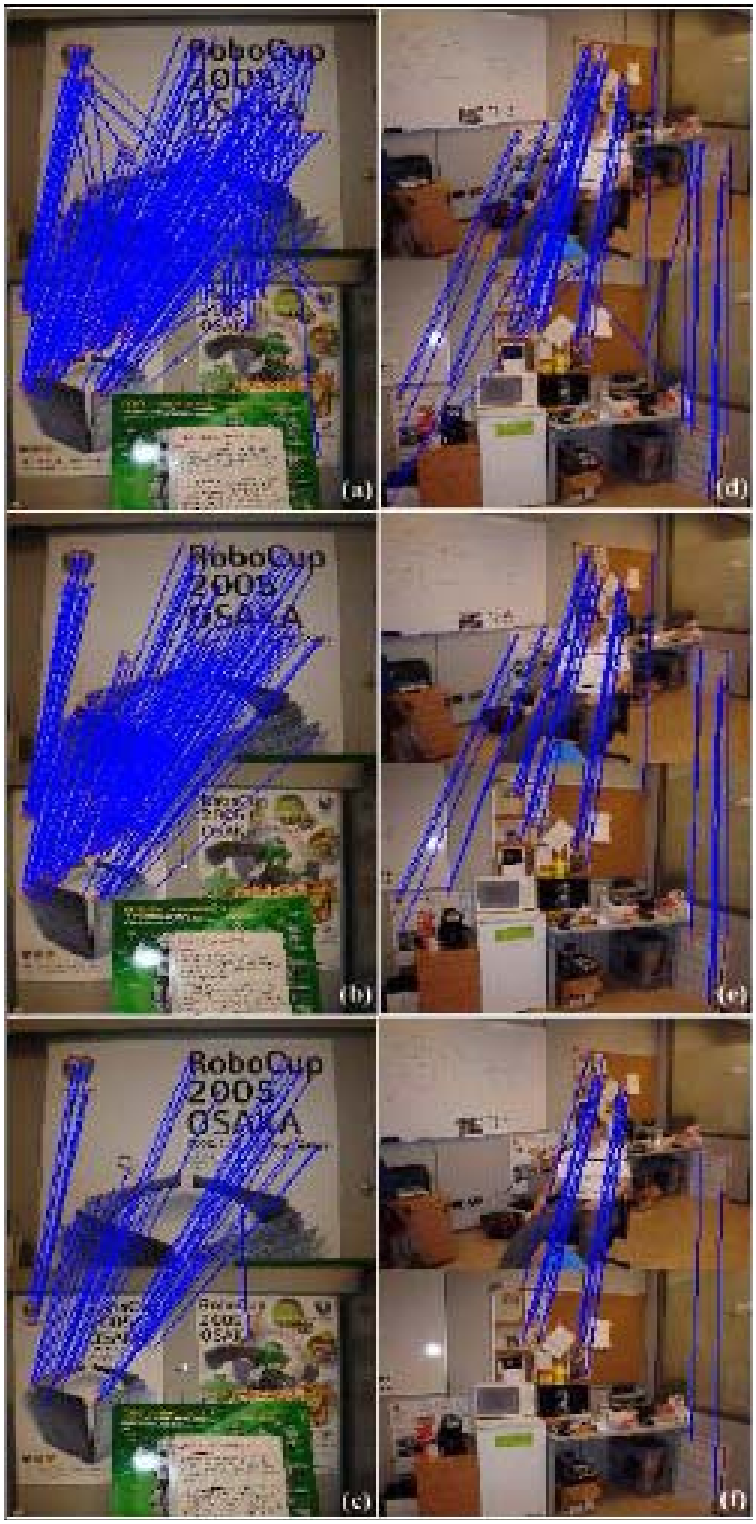


Figura 3.12 Resultados para Póster (a) – (c) y Lab (d) – (f) con SIFT. (a) Correspondencias iniciales para Póster, (b) Correspondencias finales para el TG, (c) Correspondencias finales para el GE-TPS, (d) Correspondencias iniciales para Lab, (e) Correspondencias finales para el TG, (f) Correspondencias finales para el GE-TPS.

3. Se utiliza un método robusto, como RANSAC o mínimos cuadrados, para estimar la geometría epipolar o trifocal del sistema de la cámara.
4. Los pares candidatos, que son incompatibles con la geometría de la cámara, son rechazados.
5. Se desarrolla la geometría estimada para encontrar más correspondencias.

La eficiencia y precisión de este esquema depende mucho de la calidad de las correspondencias candidatas obtenidas en el paso 2. El estimador usado en el paso 3, aunque es robusto al ruido, requiere de un conjunto inicial de correspondencias candidatas, con un número de correspondencias suficientes para encontrar una solución precisa, y con una proporción baja de errores. Por estas razones, se necesita agregar otro paso entre los pasos 2 y 3 para filtrar las correspondencias candidatas. Esto, generalmente, se hace con la introducción de algunas restricciones básicas. Sin embargo, en este trabajo, se utilizarán las correspondencias generadas con el GE-TPS.

Debido a esta modificación, no sólo se logran incrementar las correspondencias, sino que la transformación se pierde. Por lo tanto, se desarrolló una última fase, llamada campo de terminación (CT). El cual está basado en el cálculo del TPS (ver capítulo 2) cuyo objetivo es relajar algunos parámetros del TPS, para permitir que correspondencias, que originalmente no fueron seleccionadas, se consideren para formar parte del grupo de correspondencias finales. Esto logra incrementar aún más las correspondencias mientras recalcula las transformaciones. El algoritmo resultante se llama GE-TPS-GE-CT.

Algoritmo GE-TPS-GE-CT

---

1. *Algoritmo GE-TPS*

2. *Algoritmo GE*. Recordemos que, el conjunto de las correspondencias resultantes de aplicar el GE-TPS, son denotadas como $\mathcal{T} = T_1, T_2, \ldots, T_r$ para la primera imagen y como $\mathcal{T}' = T_1', T_2', \ldots, T_r'$ para la segunda imagen. Donde *r* es el número de correspondencias. Estas serán las correspondencias base de las que se seleccionarán 8 puntos en forma aleatoria que servirán para estimar de nuevo la matriz fundamental. La cual representará la geometría epipolar de una forma más precisa. De tal modo que, al aplicarse a los puntos correspondientes $(\mathcal{R}, \mathcal{R}')$, se seleccionarán puntos que se habían pasado por alto y serán rechazadas un mayor número de correspondencias incorrectas y, de esta manera, se logrará incrementar la cantidad de correspondencias. Las cuales, se denotan como $O = \bar{O}_1, \bar{O}_2, \ldots, \bar{O}_m$, el conjunto de puntos extraído de $\mathcal{R}$

y como $O' = \overline{O}'_1, \overline{O}'_2, \ldots, \overline{O}'_m$ el conjunto de puntos de $\mathcal{R}'$ correspondientes a los puntos de $\mathcal{R}$. Donde m es el número de correspondencias que cumplen con la restricción epipolar.

3. *Cálculo del CT.* (ver en el capítulo 2: Algoritmo RANSAC-TPS, paso 2)
   a. ***Obtención de coeficientes.*** Se obtienen los coeficientes del TPS a partir de los conjuntos de correspondencias resultantes (O, O').

   b. ***Obtención de los puntos transformados.*** Recordemos que $\mathcal{P}$ y $\mathcal{P}'$ son puntos característicos que se obtienen a partir de la aplicación de ciertos algoritmos de extracción de características. Luego, los coeficientes del TPS son aplicados a todos los puntos $\mathcal{P}$, para obtener los puntos transformados $\mathcal{Q}$ (tal que $\mathcal{Q} = \overline{Q}_1, \overline{Q}_2, \ldots, \overline{Q}_n$).

   c. ***Correlación de los puntos***. Los puntos $\mathcal{P}$ y $\mathcal{P}'$ serán correlacionados con base en dos condiciones:

      i. La primera se basa en la medida h_{ij}, donde

$$h_{ij} = ||\overline{Q}_i - \overline{P}'_j|| = ||(\overline{x}_i - \overline{y}_i) - (\overline{x}'_j - \overline{y}'_j)|| = \sqrt{(\overline{x}_i - \overline{x}'_j)^2 + (\overline{y}_i - \overline{y}'_j)^2} \qquad (3.6)$$

      La cual trata de seleccionar los puntos de $\mathcal{P}'$ que sean más cercanos a los puntos de $\mathcal{Q}$, de modo que si $h_{ij} \leq r_{max}$, donde r_{max} fue determinado experimentalmente como 5 píxeles, entonces, el par (P_i, P'_j) se considera como candidato a ser correlacionado.

      ii. En la segunda condición, se obtienen los coeficientes de correlación (con una función de covarianza), denotados como s_{ij}, para cuantificar la similitud entre cada par $(\overline{P}_i, \overline{P}'_j)$. Si las características tienen una alta relación, es decir, que $s_{ij} \geq s_{min}$, donde s_{min} es un umbral que empíricamente se determinó como 0.8 (debido a que se obtienen los mejores resultados), quiere decir que son correspondientes.

3.3.1 Experimentos

El objetivo de este experimento es encontrar las ventajas y desventajas de que tiene el algoritmo GE-TPS-GE-CT sobre el GE-TPS y sobre el TG.

Prueba 1: Comparación contra el algoritmo GE-TPS

Esta prueba se aplicó a 10 pares de imágenes (par1, par2, …, par10) de 640×480, que muestran distintos movimientos de la cámara (acercamientos suaves, acercamientos bruscos, translaciones pequeñas, translaciones grandes, etc.).

a) Caso intermedio. Uso del Filtro de Unicidad y del Filtro de Simetría en los datos obtenidos con el extractor basado en gradiente.

- Transformaciones. En el cuadro 3.13 se presentan los resultados (δ y tiempo) de aplicar el algoritmo GE-TPS-GE-CT con 100 iteraciones. Comparándolos con los resultados mostrados en el cuadro 3.8, se concluyó que en la mayoría de los casos empeoraron tanto la energía de la deformación como el tiempo. Aún así, comparándolo con el cuadro 3.6, se puede ver que el 50% de los casos es mejor que el RANSAC-TPS con 100 iteraciones.

Par de imágenes	GE-TPS-GE-CT	
	δ	Tiempo
Pair1	56.4472	15.437
Pair2	3329.4	15.688
Pair3	2607.1	21.64
Pair4	469.2019	25.625
Pair5	419.2321	18.75
Pair6	421.2117	18.86
Pair7	2704.7	15.719
Pair8	3478	20.297
Pair9	673.1877	11.078
Pair10	5536.5	19.61

Cuadro 3.13 Muestra los resultados (energía de la deformación δ y tiempo) de aplicar el algoritmo GE-TPS-GE-CT con el Filtro de Unicidad y el Filtro de Simetría.

- Correspondencias. El cuadro 3.14 presenta los resultados de un proceso de validación aplicado al algoritmo GE-TPS-GE-CT. Comparándolo con el cuadro 3.9, se puede observar que el número de correspondencias correctas se incrementó en el 80% de los casos, aunque también se incrementó o se igualó el número de correspondencias incorrectas. Del 20% restante, disminuyeron tanto las correspondencias correctas como las incorrectas. En todos los casos el tiempo se incrementó. Las figuras 3.13- 3.15 muestran algunos resultados. Comparándolo con el cuadro 3.3, se encontró que en la mayoría

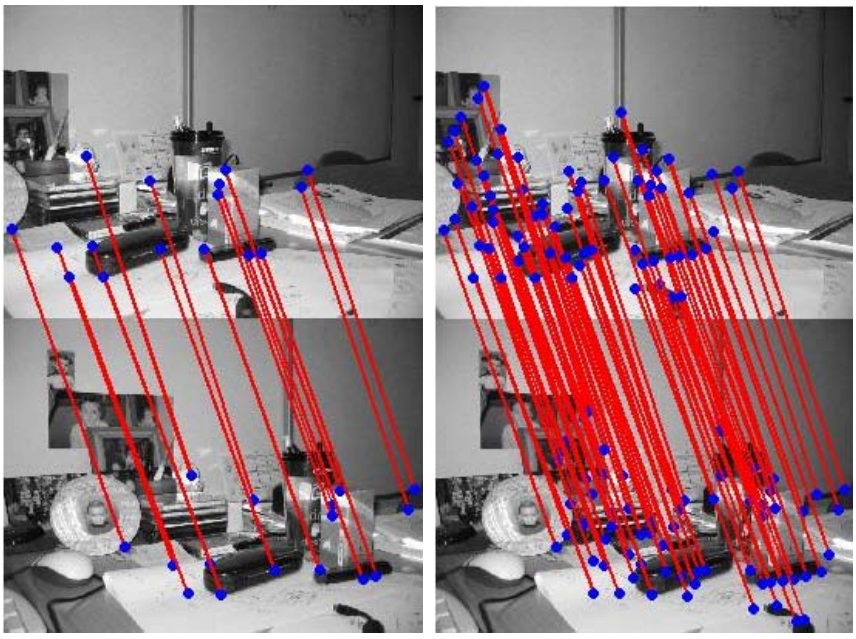
de los casos se incrementaron las correspondencias correctas, y las incorrectas se incrementaron o se mantuvieron iguales.

Par de imágenes	Entrada					Salida								
								GE-TPS-GE-CT						
		Correctas		Incorrectas		Finales		Correctas			Incorrectas			
	Iniciales		% E		% E		% E		% E	% F		% E	% F	Tiempo
Par1	62	54	87	8	13	24	38.7	24	44.4	100	0	0	0	18.36
Par2	21	11	52	10	48	22	105	13	118.1	50	9	90	50	8.579
Par3	16	5	31	11	69	4	25	1	20	25	3	27.3	75	3.531
Par4	92	83	90	9	9.8	97	105	97	117	100	0	0	0	15.453
Par5	42	35	83	7	17	33	78.6	32	91.4	97	1	14.3	3.03	12.438
Par6	24	21	88	3	13	22	91.7	20	95.2	90.9	2	66.7	9.09	13.547
Par7	28	18	64	10	36	5	17.9	2	11.1	40	3	30	60	7.094
Par8	107	104	97	3	2.8	46	43	46	44.2	100	0	0	0	24.078
Par9	37	32	86	5	14	29	78.4	29	90.6	100	0	0	0	17.766
Par10	29	16	55	13	45	38	131	26	163	68.4	12	92.3	31.6	21.172

Cuadro 3.14 Muestra las correspondencias obtenidas al aplicar el algoritmo GE-TPS-GE-CT con el Filtro de Unicidad y el Filtro de Simetría. La columna Entrada indica el número de correspondencias iniciales, es decir, aquellas obtenidas después de haberles aplicado los filtros. La columna Finales indica el número de correspondencias finales, es decir, aquellas obtenidas después de haberles aplicado los algoritmos. La columna %E indica el porcentaje de correspondencias correctas o incorrectas (según sea el caso) con respecto a las correspondencias iniciales. La columna %F indica el porcentaje de correspondencias correctas o incorrectas (según sea el caso) con respecto a las correspondencias finales. La columna Tiempo indica los segundos de procesamiento del algoritmo.

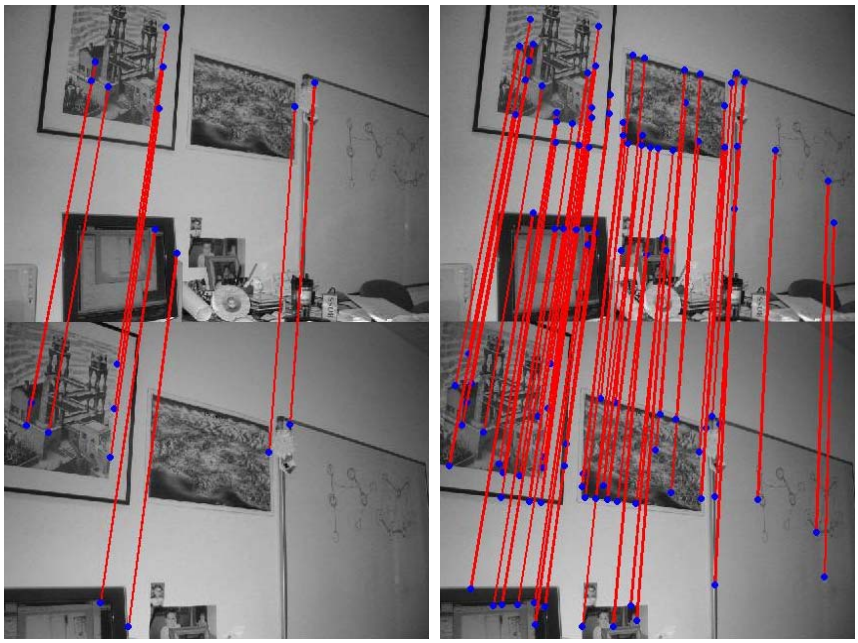
Prueba 2: Comparación contra el algoritmo de Transformación de Grafos

Se hicieron experimentos con 2 pares de imágenes (par1, par8) que muestran distintos movimientos de la cámara. A cada par de imágenes se les aplicó una extracción de características basada en el gradiente haciendo uso del Filtro de Unicidad y del Filtro de Simetría. En el cuadro 3.15 se puede observar que no se logró mejorar ni los tiempos, ni la cantidad de correspondencias del algoritmo TG.



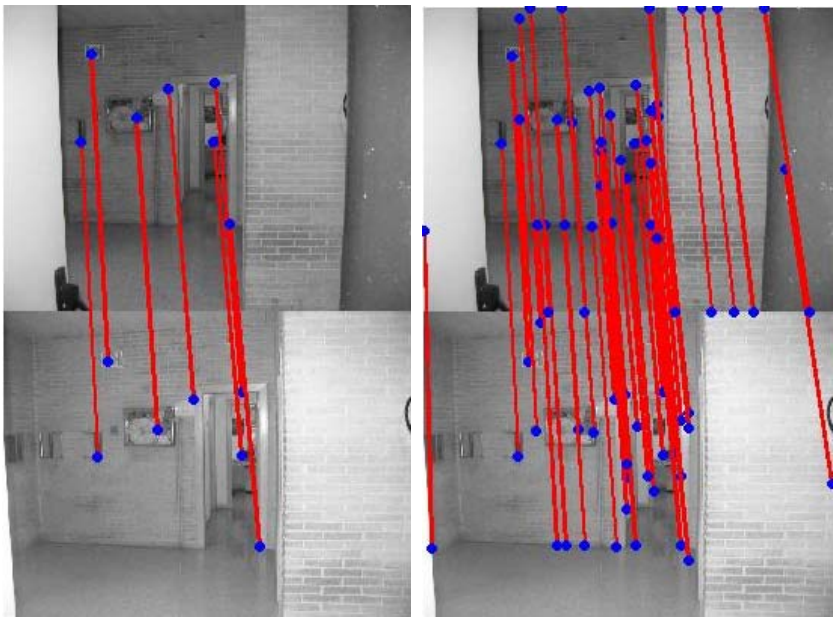
(a) GE-TPS (b) GE-TPS-GE-CT

Figura 3.13 Resultados de Par1 (pequeña traslación) al aplicar los algoritmos GE-TPS y GE-TPS-GE-CT con el Filtro de Unicidad y el Filtro de Simetría.



(a) GE-TPS (b) GE-TPS-GE-CT

Figura 3.14 Resultados de Par5 (Traslación) al aplicar los algoritmos GE-TPS y GE-TPS-GE-CT con el Filtro de Unicidad y el Filtro de Simetría.



(a) GE-TPS (b) GE-TPS-GE-CT

Figura 3.15 Resultados de Par6 (Traslación) al aplicar los algoritmos GE-TPS y GE-TPS-GE-CT con el Filtro de Unicidad y el Filtro de Simetría.

	Entrada					Salida																	
								TG								GE-TPS-GE-CT							
		Correctas		Incorrectas		Finales		Correctas			Incorrectas				Finales	Correctas			Incorrectas				
Par de imágenes	Iniciales		% E		% E		% E		% E	% F		% E	% F	Tiempo			% E	% F		% E	% F	Tiempo
Par1	62	54	87	8	13	38	61	38	70	100	0	0	0	0.09	24	39	24	44	100	0	0	0	18.36
Par8	107	104	97	3	3	75	70	75	72	100	0	0	0	0.25	46	43	46	44	100	0	0	0	24.078

Cuadro 3.15 Muestra las correspondencias obtenidas al aplicar los algoritmos TG y GE-TPS-GE-CT con el Filtro de Unicidad y el Filtro de Simetría. La columna Entrada indica el número de correspondencias iniciales, es decir, aquellas obtenidas después de haberles aplicado los filtros. La columna Finales indica el número de correspondencias finales, es decir, aquellas obtenidas después de haberles aplicado los algoritmos. La columna %E indica el porcentaje de correspondencias correctas o incorrectas (según sea el caso) con respecto a las correspondencias iniciales. La columna %F indica el porcentaje de correspondencias correctas o incorrectas (según sea el caso) con respecto a las correspondencias finales. La columna Tiempo indica los segundos de procesamiento del algoritmo.

3.4 Síntesis

En este capítulo se describieron los algoritmos ISRANSAC-TPS, GE-TPS y GE-TPS-GE-CT. También se presentaron los resultados obtenidos en los experimentos realizados a los algoritmos.

Por un lado, al comparar el ISRANSAC-TPS contra el RANSAC-TPS, se concluyó que el primero tuvo mejoras en la energía de la deformación con respecto al segundo, aunque en algunas, poco significativas. Sin embargo, el tiempo de procesamiento es menor en el RANSAC-TPS. También se observó que en ambos casos se conservan aproximadamente el 30% de las correspondencias iniciales. Esto debido a que la selección aleatoria de puntos es sobre una muestra del 30%. Se mostró que en algunos casos difíciles de resolver para el RANSAC-TPS, donde son pocas las correspondencias finales y no logra dejar pasar correspondencias correctas, el ISRANSAC-TPS las resuelve mejor al incrementar el número de correspondencias correctas. También se encontró que el algoritmo es sensible a correspondencias incorrectas a pesar del uso de RANSAC. Se encontró que si el número de correspondencias iniciales es muy alto, este método no es capas de resolverlo, ya que el número de iteraciones se basa en las correspondencias iniciales. Por lo que, el algoritmo no es útil para aplicaciones en tiempo real, a menos que se limite la cantidad de correspondencias iniciales, por ejemplo, mediante una selección aleatoria de puntos. Las iteraciones óptimas, para tener buenos resultados, son 5,000 para el RANSAC-TPS y 1,000 para el ISRANSAC-TPS.

Por otro lado, al comparar los resultados de los algoritmos de TPS (RANSAC-TPS e ISRANSAC-TPS), contra el algoritmo de GE, se pudo observar que, en todos los casos, el tiempo en éste es menor. Su ventaja principal, es que deja pasar muchas correspondencias correctas. La desventaja es que también pasan muchas incorrectas. Mientras que en los algoritmos basados en TPS ocurre lo contrario, dejan pasar pocas correspondencias incorrectas, aunque también pocas correctas. Por otro lado, el ISRANSAC-TPS es más caro que cualquiera de los anteriores, aunque las correspondencias sean mejores en la mayoría de los casos. El GE-TPS logró una mejora sobre los resultados de GE y RANSAC-TPS y sobre el tiempo del ISRANSAC-TPS, tanto en correspondencias como en la energía de la deformación (en todos los casos con 100 iteraciones y en el 70% de los casos con 5000 iteraciones), pero disminuyó el número de correspondencias correctas. También se encontró que el eliminar todos los filtros, hace inútil es uso de un algoritmo directo. La ventaja hasta este punto es que se tiene un algoritmo no solo con transformaciones, sino con buenas correspondencias.

Para tratar de solucionar el problema de la escasa cantidad de correspondencias del GE-TPS, se implementó el algoritmo GE-TPS-GE-CT. Luego de algunas pruebas, se encontró que este algoritmo logra incrementar las correspondencias correctas, sin embargo, en algunos casos, también incrementa las incorrectas. También se pudo ver que la energía de la deformación es mejor que el RANSAC-TPS con 100 iteraciones en el 50% de los casos.

Finalmente, se hizo un comparativo entre el algoritmo TG contra el GE-TPS y el GE-TPS-GE-CT y, en ambos casos, se observó que TG deja pasar más correspondencias correctas y es más rápido.

PARTE 3
CÁLCULO DE LA ESTIMACIÓN DEL MOVIMIENTO

Capítulo 4

Estimación del movimiento

Como se explicó anteriormente, el objetivo de la estimación del movimiento propio consiste en recuperar la rotación instantánea del observador y la dirección de translación mientras se mueve a través del ambiente, tomando una secuencia de imágenes como entrada.

En esta tesis se implementaron cuatro algoritmos para resolver el problema de la orientación relativa, los cuales se clasifican en:

a) Algoritmos de la matriz esencial. Son algoritmos lineales que requieren por lo menos ocho puntos, aunque se han hecho extensiones de estos algoritmos cuando solo hay cinco o siete puntos disponibles.

Recordemos que $P = (x, y, z)^T$ y $P' = (x', y', z')^T$ son los puntos del objeto antes y después del movimiento y que $\overline{P} = (\overline{x}, \overline{y}, 1)^T = P/z$ y $\overline{P}' = (\overline{x}', \overline{y}', 1)^T = P'/z$ representan las coordenadas en perspectiva de los puntos $\overline{P}$ y $\overline{P}'$ respectivamente en el plano de la imagen.

La ecuación del modelo de movimiento rígido está dada como:

$$\overline{P}' = \mathbf{R}\overline{P} + T \tag{4.1}$$

Escogiendo cualquier vector no-cero T_o que es colineal con T, y tomando su producto cruz con ambos lados de (4.1), se obtiene:

$$z' T_0 \times (\overline{x}', \overline{y}', 1)^T = z T_0 \times \left[\mathbf{R}(\overline{x}, \overline{y}, 1)^T \right] \tag{4.2}$$

Tomando el producto interno de ambos lados de (4.2) con $(\bar{x}', \bar{y}', 1)$ se llega a:

$$\left(\bar{x}', \bar{y}', 1\right)(T_0 \times \mathbf{R})\left(\bar{x}, \bar{y}, 1\right)^T = 0 \tag{4.3}$$

ó

$$\bar{P}'^T \left(T_0 \times \mathbf{R}\right)\bar{P} = 0 \tag{4.4}$$

Donde

$$T_0 \times \mathbf{R} = \left[T_0 \times r_1, T_0 \times r_2, T_0 \times r_3\right] \tag{4.5}$$

y r_1, r_2, r_3 son las columnas de $\mathbf{R}$.

De acuerdo a (2.54), se define la matriz $\mathbf{E}'$, la cual es una matriz que se aproxima a la matriz esencial $\mathbf{E}$ (ver capítulo 2) , a partir de la siguiente ecuación:

$$\mathbf{E}' = T_0 \times \mathbf{R} \tag{4.6}$$

Pero también puede descomponerse como $\left(-T_0\right) \times \mathbf{R}$. Con el objetivo de determinar la descomposición correcta se nota que

$$\mathbf{E}' = \left[T_0 \times r_1, T_0 \times r_2, T_0 \times r_3\right] \tag{4.7}$$

Así, sus tres columnas expanden a un espacio 2D. Por otro lado,

$$\left\|\mathbf{E}'\right\| = \left\|T_0\right\| \tag{4.8}$$

Por lo tanto se pueden tener tres restricciones como sigue:

1. El rango de $\mathbf{E}' = 2$.
2. $\left\|\mathbf{E}'\right\| = \left\|T_0\right\|$.
3. $\mathbf{E}'^T T_0 = 0$ (4.9)

Donde T_0 es un vector propio unitario de $\mathbf{E}'^T$ correspondiente al valor propio cero.

Así que se utilizó un método de optimización restringido no lineal para encontrar los valores del vector T_0 que minimizan la función (4.9), iniciando con un valor de T_0 aleatorio y sujeto a las restricciones 1 y 2.

Un cuerpo rígido está en movimiento en el medio espacio $z \leq 0$ (ver figura 4.1), lo cual implica que T tiene la misma orientación que T_0 o $(-T_0)$, si y solo si $(\bar{x}', \bar{y}', 1)^T \times \left[R(\bar{x}, \bar{y}, 1)^T \right]$ tiene la misma orientación que $(\bar{x}', \bar{y}', 1)^T \times T_0$ ó $\left[-(\bar{x}', \bar{y}', 1)^T \times T_0 \right]$.

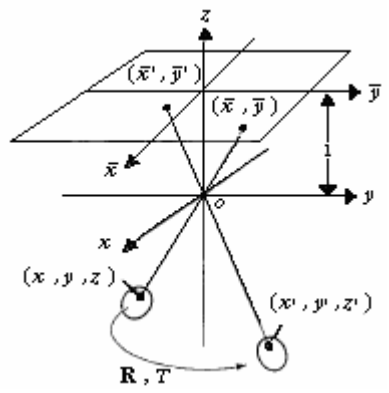


Figura 4.1 Geometría de la imagen.

Una vez que la T_0 correcta es determinada, la verdadera $\mathbf{R}$ es determinada como sigue:

$$\mathbf{R} = \left[\mathbf{E}'_2 \times \mathbf{E}'_3, \mathbf{E}'_3 \times \mathbf{E}'_1, \mathbf{E}'_1 \times \mathbf{E}'_2 \right] - \left[T_0 \right]^{\times} \times \mathbf{E}' \qquad (4.10)$$

Donde

$$\mathbf{E}' = \left[\mathbf{E}'_1, \mathbf{E}'_2, \mathbf{E}'_3 \right] \qquad (4.11)$$

El cálculo de la matriz $\mathbf{E}'$ se implementó con el algoritmo lineal (Alg1) (ver cuadro 2.2) y con el algoritmo robusto (Alg2) (ver cuadro 2.3).

b) Algoritmos directos. Estos algoritmos encuentran la rotación y la traslación directamente y requieren la minimización de funciones de costo no lineales.

A continuación se explicarán los algoritmos directos implementados como ejemplo de esta clasificación.

1. Algoritmo para minimizar la ecuación del movimiento rígido (Alg3). El objetivo de este algoritmo es encontrar los parámetros de movimiento directamente minimizando la función de costo:

$$\sum_{j=0}^{m} \bar{P}_j'^T \left(T_0 \times \mathbf{R} \right) P_j \tag{4.12}$$

que es derivada directamente de la ecuación (4.4). Para poder resolver este problema, se aplica la optimización lineal restringida, basada en el siguiente criterio de minimización:

$$\sum_{j=0}^{m} w_j \left(\bar{P}_j'^T \left(T_0 \times \mathbf{R} \right) \bar{P}_j \right)^2 \tag{4.13}$$

donde w_j es un peso que se calcula con el criterio de la distancia entre los puntos en ambas imágenes a sus líneas epipolares correspondientes:

$$w_j = \frac{1}{\left\| \mathbf{R}^T \left(T_0 \times \bar{P}_j'^T \right) \right\|^2 + \left\| \mathbf{R} \left(T_0 \times \bar{P}_j^T \right) \right\|^2} \tag{4.14}$$

Se inicia con una T_0 y una $\mathbf{R}$ aleatorias.

2. Algoritmo para minimizar la ecuación del movimiento rígido, guiando la matriz de rotación (Alg4). Es una versión del algoritmo anterior en la que, para tratar de minimizar la ecuación (4.13) de forma más precisa, en lugar de iniciar con una $\mathbf{R}$ aleatoria, se trata de guiar la matriz de rotación.

Para ello se toman en cuenta dos valores básicos para calcular dicha matriz. Por un lado se tiene el ángulo de rotación θ y por el otro el eje sobre el cual se rota.

El ángulo de rotación tiene un límite mínimo de 0 y máximo de 359 grados, aunque se podría restringir el movimiento de la cámara para incrementar la precisión como en [Zhang, Z., 1998].

Se tienen tres ejes de rotación $\left(x, y, z \right)$:

- Rotación tridimensional alrededor del eje z y en coordenadas homogéneas:

$$\mathbf{R}_z = \begin{vmatrix} \cos\theta_z & sen\theta_z & 0 & 0 \\ -sen\theta_z & \cos\theta_z & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} \quad (4.15)$$

En esta rotación, se piensa en el eje z como fijo mientras que algún objeto se mueve en el espacio. Se puede pensar también que el objeto permanece inmóvil mientras los ejes se mueven. La diferencia entre uno y otro planteamiento es la dirección de la rotación. Fijar el eje y girar el objeto en sentido antihorario es lo mismo que fijar el objeto y mover el eje en sentido horario.

- La rotación alrededor del eje x se realiza con una matriz de transformación similar a la anterior:

$$\mathbf{R}_x = \begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta_x & sen\theta_x & 0 \\ 0 & -sen\theta_x & \cos\theta_x & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} \quad (4.16)$$

- Para hacer una rotación alrededor del eje y, se realiza la siguiente matriz:

$$\mathbf{R}_y = \begin{vmatrix} \cos\theta_y & 0 & -sen\theta_y & 0 \\ 0 & 1 & 0 & 0 \\ sen\theta_y & 0 & \cos\theta_y & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} \quad (4.17)$$

Así que, para tratar de guiar la matriz de rotación, aleatoriamente se selecciona, por un lado, un ángulo de rotación $\left(\theta_x, \theta_y, \theta_z\right)$ entre los límites definidos y por otro, un eje de rotación. Con estos datos se calcula la matriz de rotación y se aplica la función a minimizar (4.13). El algoritmo no está optimizado ya que el objetivo es analizar si existe alguna mejora.

4.1 Experimentos

Para realizar los experimentos, se crearon datos controlados que permitieran evaluar los resultados de los algoritmos. Se generó un conjunto de puntos 3D, P, a los que se les aplicó una matriz de rotación $\mathbf{R}$ y un vector de

traslación T determinados. Posteriormente, se aplicó la fórmula (4.1) para obtener sus puntos correspondientes, P'. Luego, los puntos P y P' fueron proyectados en 2D para poder aplicar los algoritmos.

La forma de verificar que los resultados fueran correctos, fue aplicando la fórmula (4.1), para obtener los puntos correspondientes calculados P'', con las $\mathbf{R}$ y T obtenidas. Finalmente se obtuvo la distancia entre P' y P''. Mientras menor sea la distancia entre ellos, se considera que es mejor el algoritmo.

Se hicieron dos tipos de experimentos: datos controlados con y sin correspondencias incorrectas.

a) Datos controlados sin correspondencias incorrectas. Se generaron grupos de 10 y 100 puntos correspondientes, a los cuales se les aplicaron los algoritmos mencionados anteriormente, con el objetivo de analizar cuál obtiene las distancias entre P' y P'' más pequeñas y cuál es el más rápido. Se hicieron 10 corridas para cada algoritmo.

   En el cuadro 4.1 se muestran los resultados (distancia, D, entre P' y P'', y tiempo) de las pruebas realizadas a los algoritmos anteriormente descritos, aplicados a 10 pares de puntos correspondientes. A partir del promedio de ellos, se puede observar que el Alg2 obtiene los mejores resultados, ya que muestra las distancias entre P' y P'' más cortas. El tiempo es un poco mayor que el Alg2, sin embargo, es bastante bueno. Por otro lado, el haber guiado la matriz de rotación en el Alg4 resultó en una mejora en comparación al Alg3.

   Las figuras 4.2-4.9 muestran algunos resultados gráficados en un plano de coordenadas. Los puntos del que parten las dos líneas (puntos rojos) representan a los puntos P, los puntos obscuros (azules) representan a los puntos P' y los puntos claros (verdes) representan a los puntos P''. Las líneas que van de P a P' o de P a P'' indican la correspondencia entre esos puntos.

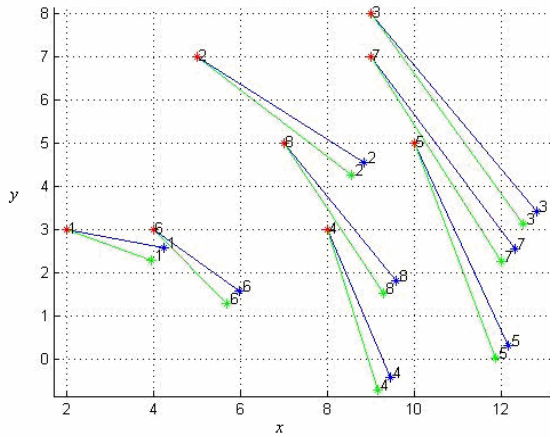


Figura 4.2 Algoritmo lineal con 10 puntos generados. Los resultados son casi iguales en cada corrida.

	Algoritmos de la matriz esencial				Algoritmos Directos			
	Alg1		Alg2		Alg3		Alg4	
Corridas	D	Tiempo	D	Tiempo	D	Tiempo	D	Tiempo
1	3.3137	0.141	1.10E-04	0.391	62.4383	3.234	62.237	3.297
2	3.3138	0.109	5.74E-05	0.359	46.57	3.063	34.7898	6.969
3	3.3141	0.156	0.0156	0.328	54.4996	3.157	88.2533	3.234
4	3.3137	0.078	0.0017	0.359	26.6839	3.234	109.4485	3.172
5	3.3136	0.094	1.04E-04	0.328	1.06E+03	3.203	122.554	3.109
6	3.3091	0.156	6.37E-04	0.312	232.9773	1.422	83.8347	0.969
7	3.3137	0.078	7.84E-05	0.344	117.6297	0.796	74.452	3.203
8	3.3138	0.141	3.13E-06	0.343	151.6369	3.25	84.2461	3.156
9	3.3136	0.109	1.38E-04	0.391	694.2791	3.157	37.8766	3.203
10	3.3138	0.156	1.06E-04	0.344	52.0534	1.36	109.7986	3.25
Promedio	3.31329	0.1218	1.85E-03	0.35	250.3368	2.5876	80.74906	3.3562

Cuadro 4.1 Muestra los resultados obtenidos al aplicar los algoritmos de la estimación del movimiento a un conjunto de 10 pares de puntos. Se hicieron 10 corridas con los mismos datos de entrada. La columna D indica la distancia entre P' y P''. La columna Alg1 se refiere al algoritmo lineal, la columna Alg2 se refiere al algoritmo robusto, la columna Alg3 se refiere al algoritmo para minimizar la ecuación del movimiento rígido y la columna Alg4 se refiere al algoritmo para minimizar la ecuación del movimiento rígido, guiando la matriz de rotación.

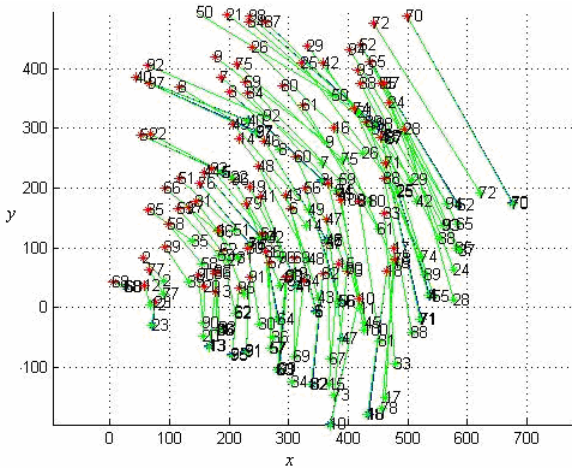


Figura 4.3 Algoritmo lineal con 100 puntos generados. Los resultados son casi iguales en cada corrida.

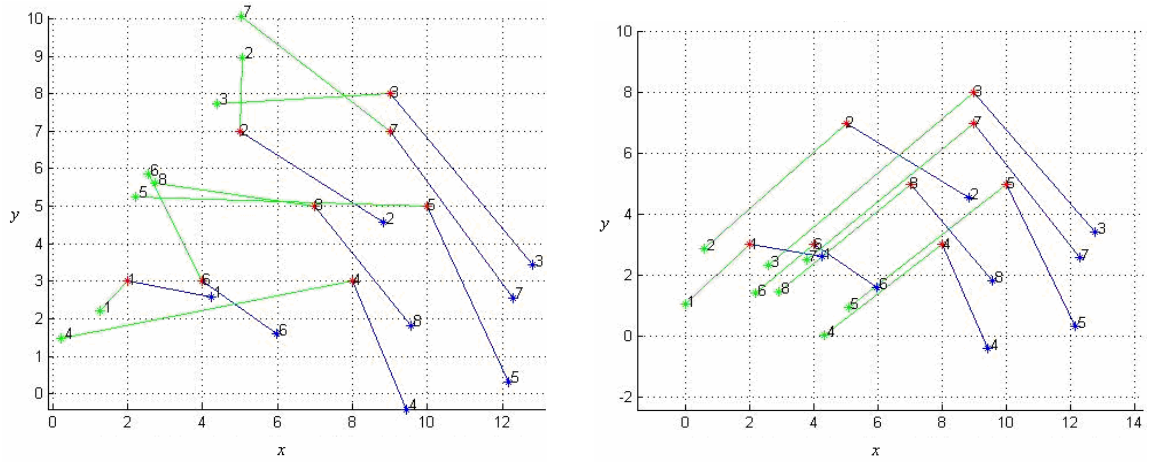


Figura 4.4 Algoritmo para minimizar la ecuación del movimiento rígido con 10 puntos generados. Los resultados pueden variar mucho entre cada corrida, aquí se presentan dos ejemplos.

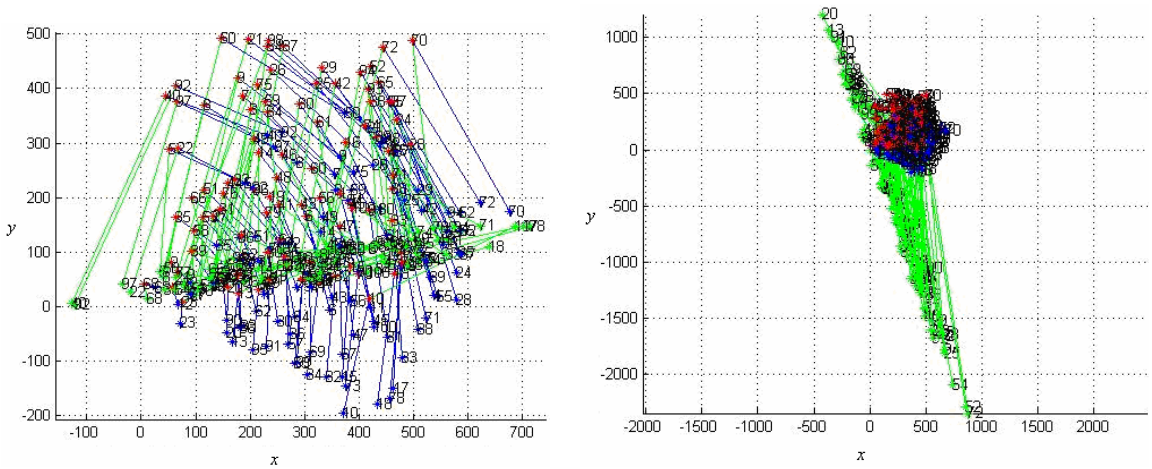


Figura 4.5 Algoritmo para minimizar la ecuación del movimiento rígido con 100 puntos generados. Los resultados pueden variar mucho entre cada corrida, aquí se presentan dos ejemplos.

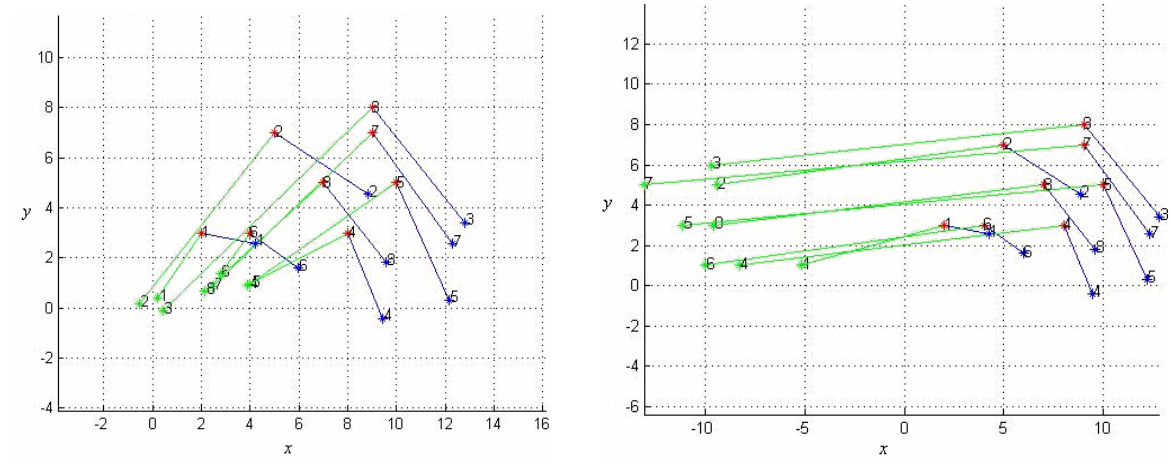


Figura 4.6 Algoritmo para minimizar la ecuación del movimiento rígido, guiando la matriz de rotación con 10 puntos generados. Los resultados pueden variar mucho entre cada corrida, aquí se presentan dos ejemplos.

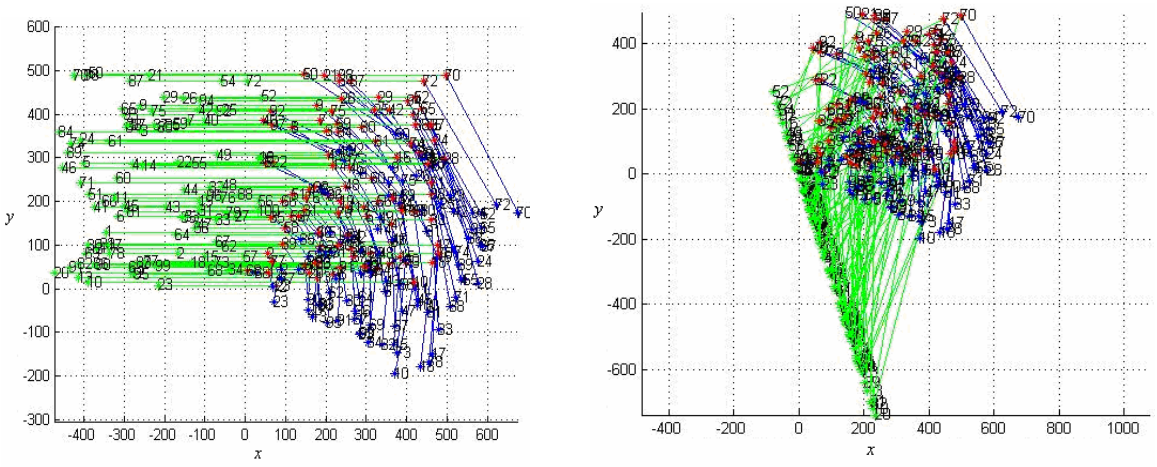


Figura 4.7 Algoritmo para minimizar la ecuación del movimiento rígido, guiando la matriz de rotación con 100 puntos generados. Los resultados pueden variar mucho entre cada corrida, aquí se presentan dos ejemplos.

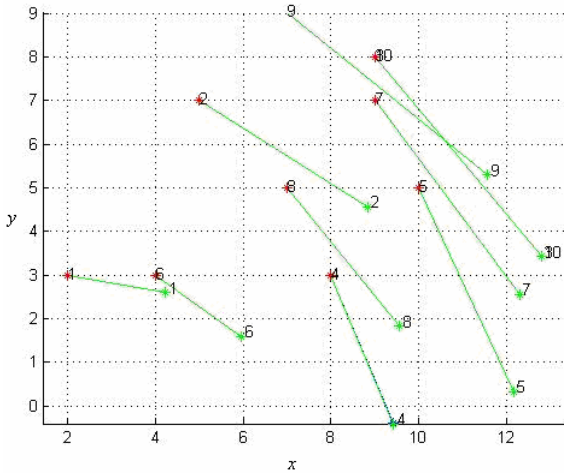


Figura 4.8 Algoritmo robusto con 10 puntos generados. Los resultados son casi iguales en cada corrida.

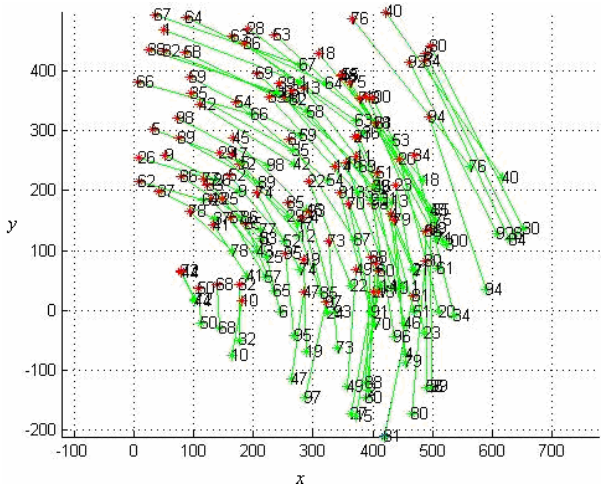


Figura 4.9 Algoritmo robusto con 100 puntos generados. Los resultados son casi iguales en cada corrida.

b) Datos controlados con correspondencias incorrectas. Como se pudo observar en el experimento anterior, los algoritmos con mejores resultados fueron el algoritmo lineal y el algoritmo robusto por lo que también fueron sometidos a un experimento con datos que tenían hasta 3 y 4% de correspondencias incorrectas (porque más de esto ya no se obtienen buenos resultados) con el objetivo de encontrar quien tiene una mayor robustez. Se hicieron 5 corridas de cada algoritmo.

En el cuadro 4.2 se muestran los resultados de las pruebas realizadas al algoritmo lineal aplicados a 100 pares de puntos correspondientes. Se encontró que el que soporta más correspondencias incorrectas es el algoritmo robusto sin embargo solo soporta menos del 4% antes de que genere malos resultados.

	Alg1			Alg2			
Corridas	1%	2%	3%	1%	2%	3%	4%
1	69.7156	201.6151	9690	0.031	0.0742	1.0472	37300
2	67900	28400	40800	0.0314	0.07	0.3011	31100
3	31900	77600	35700	0.032	0.0492	37300	43600
4	45.2049	78300	76500	0.0719	1.2197	0.0378	215.7449
5	42.7412	827.627	25900	0.1567	0.0311	37300	49400
Promedio	19991.5	37065.85	37718	0.0646	0.2888	14920.28	32323.15

Cuadro 4.2 Muestra la distancia entre P' y P'' obtenidos al aplicar el algoritmo lineal a un conjunto de 100 pares de puntos correspondientes, con hasta el 3% de correspondencias incorrectas. Se hicieron 5 corridas con los mismos datos de entrada.

4.2 Estructura de la estimación de movimiento

La estructura de la estimación del movimiento propio y las herramientas que se sugieren para cada etapa se pueden resumir en el esquema mostrado en la figura 4.10.

Se sugiere que las imágenes de entrada sean adquiridas con cámaras de tipo "*pinhole*", porque son más comunes y económicas que las cámaras de retina esférica, aunque, como ya se explicó en el capítulo 1, éstas técnicamente tienen más ventajas.

Para la extracción de características se sugiere el uso de SIFT (ver capítulo 1) por ser un muy buen extractor de características y porque el ejecutable está disponible para su uso.

Para la correspondencia entre pares de imágenes se recomienda usar el GE-TPS (ver capítulo 3). Sin embargo, debido a que el tiempo que tarda en resolver las correspondencias entre pares de imágenes, depende del

número de las características extraídas, se deben seleccionar en forma aleatoria un conjunto de éstas, dependiendo del tiempo requerido en la aplicación en la que se vaya a implementar.

Finalmente, para estimar el movimiento propio se sugiere aplicar el algoritmo robusto. Recordemos que éste necesita como mínimo 9 correspondencias correctas, así que, en caso de que las correspondencias obtenidas no sean suficientes, será necesario aplicar el algoritmo GE-TPS-GE-CT.

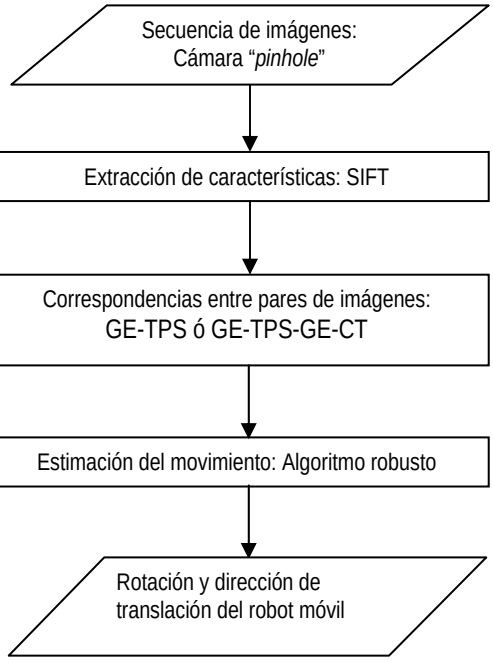


Figura 4.10 Esquema de estimación del movimiento y herramientas sugeridas

4.3 Síntesis

En este capítulo se mostraron los resultados de la implementación de dos algoritmos basados en la matriz esencial: el algoritmo lineal y el robusto; y dos basados en el método directo: algoritmo para minimizar la ecuación del movimiento rígido y su versión guiando la matriz de rotación.

Se hicieron dos experimentos con datos controlados, el primero de los cuales se hizo sin correspondencias incorrectas, para comparar las distancias entre los puntos calculados y los puntos reales, así como el costo computacional, de los cuatro algoritmos. En esta prueba se encontró que los mejores en ambos aspectos fueron el algoritmo robusto y el algoritmo lineal. El segundo experimento se hizo con cierto porcentaje de correspondencias

incorrectas, para comparar la robustez de los dos algoritmos. En esta prueba se puede ver que el que soporta más correspondencias incorrectas es el algoritmo robusto, sin embargo, solo soporta menos del 4%.

Finalmente se sugiere que, para obtener las correspondencias necesarias para estimar el movimiento propio, se utilice el GE-TPS (ver capítulo 3). El cual es un algoritmo que obtiene buenas correspondencias que, aunque pocas, son suficientes para aplicar el algoritmo robusto, ya que éste necesita como mínimo 9 correspondencias correctas. En caso de que no se consigan las necesarias, se recomienda utilizar el GE-TPS-GE-CT para incrementarlas.

Conclusiones y trabajo futuro

Para poder recuperar la rotación instantánea del observador y la dirección de translación mientras se mueve a través del ambiente (movimiento propio), se aplicó la estimación mediante la percepción del entrono, cuyo problema es que una de sus etapas, la correspondencia entre puntos, no ha sido solucionada satisfactoriamente, y esto impide que la estimación sea precisa.

Por lo tanto, en esta tesis se implementaron algoritmos que permiten encontrar el número de correspondencias correctas, necesario para llevar a cabo la estimación del movimiento. Dichos algoritmos se basan en la Geometría Epipolar (GE) y el *"Thin-Plate spline"* (TPS).

El objetivo del TPS es encontrar una transformación no rígida entre los conjuntos de puntos correspondientes de dos imágenes. Como una de sus desventajas es que la solución es muy costosa, debido a que durante su cálculo se debe invertir una matriz cuadrada del tamaño del número de puntos correspondientes, se decidió implementar el algoritmo llamado RANSAC-TPS. El cual selecciona un subconjunto de puntos correspondientes, aplicando el RANSAC de forma incremental, a partir del cual se obtienen los coeficientes de la transformación, para luego, aplicarlos al total de los puntos de la primera imagen. Esto da como resultado unos puntos que deben ser aproximados a los de la segunda imagen mediante la minimización de la energía de la deformación, que busca la transformación óptima entre ellos. Con lo cual se obtendrá no solo la transformación no rígida, sino puntos correspondientes.

Sin embargo, se creó el algoritmo ISRANSAC-TPS, con el objetivo de disminuir la aleatoriedad de la selección de las muestras y, con esto, obtener mejores resultados que la versión RANSAC-TPS.

Estos dos algoritmos fueron sometidos a ciertos experimentos, de los que se concluyó que se obtienen transformaciones similares en ambos casos y algunas un poco mejores con el ISRANSAC-TPS; que el tiempo del procesamiento para el mismo número de iteraciones es similar y en algunos casos mayor con este último; y que en

ambos casos se mejoran los resultados al incrementar el número de iteraciones. Una de las mayores desventajas del ISRANSAC-TPS es que el número de iteraciones depende del número de correspondencias iniciales, lo cual lo hace ineficiente para aplicaciones en tiempo real. También se encontró que en algunos casos difíciles de resolver para el RANSAC-TPS, donde son pocas las correspondencias finales y no logra dejar pasar correspondencias correctas, el ISRANSAC-TPS las resuelve mejor al hacer casi iguales el número de correspondencias correctas que el número de correspondencias finales. También se encontró que los algoritmos son sensibles a correspondencias incorrectas.

Luego, con el objetivo de obtener más y mejores correspondencias, se aplicó la geometría epipolar a los algoritmos anteriores. La mezcla de ambas técnicas dio origen al algoritmo llamado GE-TPS, el cual permite obtener muy buenas correspondencias, sin embargo, en algunos casos, no se obtienen las necesarias.

A través de varios experimentos se encontró que los algoritmos son muy sensibles al ruido porque la GE tiene que representar con 8 puntos, seleccionados de forma aleatoria, la geometría de todos los demás. El tiempo que tarda el algoritmo en resolver el problema está relacionado con el número de correspondencias iniciales.

Debido a que se requiere un algoritmo lo más económico posible, se utilizó en todos los casos el RANSAC-TPS. Sin embargo, queda como trabajo futuro buscar reducir los tiempos de procesamiento del ISRANSAC-TPS para aprovechar las ventajas que tiene sobre éste.

El algoritmo GE-TPS logró disminuir el número de correspondencias incorrectas que se encontraban en cada algoritmo por separado, pero también el número de correspondencias correctas. La ventaja de este algoritmo es que se obtienen transformaciones con buenas correspondencias.

Por lo tanto, aprovechando dichas ventajas, se implementó un algoritmo diseñado para incrementar las correspondencias, llamado GE-TPS-GE-CT.

El cual obtiene una matriz fundamental que representa mejor la geometría de todos los puntos y, con ella, se filtran los datos de entrada. Este algoritmo incrementa las correspondencias incorrectas pero también puede incrementar el número de correspondencias correctas, en algunos casos hasta del 70% sin incremento del error y sobre todo, conserva las transformaciones.

También se implementaron algunos algoritmos para estimar el movimiento propio, basados en la matriz esencial: el algoritmo lineal y el robusto; y algunos basados en el método directo: algoritmos para minimizar la ecuación del movimiento rígido y su versión guiando la matriz de rotación.

Se hicieron experimentos con datos controlados y se encontró que los mejores resultados se obtuvieron con el algoritmo lineal y con el algoritmo robusto, el cual soporta más correspondencias incorrectas, aunque solo menos del 4%.

Finalmente, se sugiere que para obtener las correspondencias necesarias para estimar el movimiento propio, se utilice el GE-TPS. El cual es un algoritmo que obtiene buenas correspondencias que, aunque pocas, son suficientes para aplicar el algoritmo robusto, ya que éste necesita como mínimo 9 correspondencias correctas. En caso de que no se consigan las necesarias, se recomienda utilizar el GE-TPS-GE-CT para incrementarlas.

Como trabajo futuro, por un lado, sería deseable hacer un comparativo entre el algoritmo GE-TPS y el propuesto en [Chui, H. and Rangarajan, A., 2000] y, por otro lado, se deben probar los algoritmos con imágenes reales, para que, una vez ajustados los parámetros necesarios que permitan tener buenas estimaciones, se implemente en un robot móvil, con el objetivo de mejorar el algoritmo antes de adaptarlo a algún sistema para invidentes.

Apéndice 1

"Splines"

Es una función polinomial que puede tener una forma localmente muy simple, pero al mismo tiempo es globalmente muy flexible y suave. Las "*splines*" (ver figura A1.1) son muy útiles para el modelado de funciones arbitrarias, y son extensivamente utilizadas en computación gráfica, debido, principalmente, a la simplicidad de su construcción y a su capacidad para aproximar formas complejas a través de la adaptación de curvas.

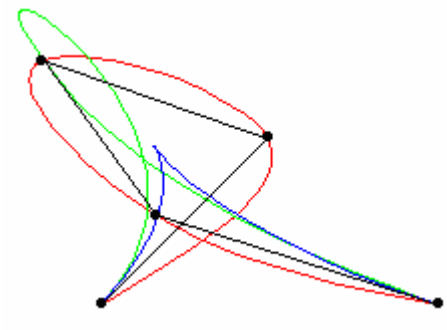


Figura A1.1 "*Splines*".

Cada polinomio de la función "*spline"* está definido sobre un subintervalo, que se unen entre sí obedeciendo a ciertas condiciones de continuidad.

Supóngase que se dispone de $n+1$ puntos denominados nudos, t_i , tales que $t_0 < t_1 < ... < t_n$. Supóngase además que se ha fijado un entero $k \geq 0$. Se dice entonces que una función *"spline"* de grado k con nudos en $t_0, t_1, \ldots, t_n$ es una función S que satisface las condiciones:

a) En cada intervalo $[t_{i-1}, t_i)$, S es un polinomio de grado menor o igual a k .

b) S tiene una derivada de orden $(k-1)$ continua en $[t_0, t_n]$.

La función *"spline"* está dividida en las siguientes curvas:

- *"Spline"* de grado cero.
- *"Spline"* de grado uno.
- *"Spline"* cúbica.
- "*Thin-Plate splines"*.
- *"B-spline"*.
- Curvas Bézier.
- Curva NURBS.

- ***"Spline" de grado cero***

Las *"splines"* de grado cero son funciones constantes por zonas. Una forma explícita de presentarlas es la siguiente:

$$S(x) = \begin{cases} S_0(x) = c_0 & x \in [t_0, t_1) \\ S_1(x) = c_1 & x \in [t_1, t_2) \\ \vdots & \vdots \\ S_{n-1}(x) = c_{n-1} & x \in [t_{n-1}, t_n) \end{cases} \qquad \text{(A1.1)}$$

Los intervalos $[t_{i-1}, t_i)$ no se intersectan entre sí, por lo que no hay ambigüedad en la definición de la función en los nudos. En la figura A1.2 se muestra la gráfica correspondiente a las *"splines"* de grado cero.

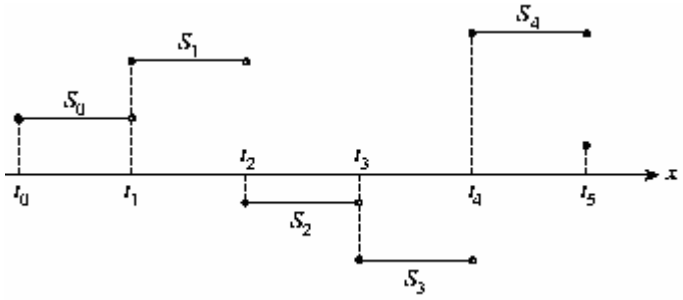


Figura A1.2 *"Spline"* de grado cero.

- ***"Spline" de grado uno***

Una *"spline"* de grado uno se puede definir por:

$$S(x)=\begin{cases} S_0(x)=a_0x+b_0 & x\in[t_0,t_1) \\ S_1(x)=a_1x+b_1 & x\in[t_1,t_2) \\ \vdots & \vdots \\ S_{n-1}(x)=a_{n-1}x+b_{n-1} & x\in[t_{n-1},t_n) \end{cases} \tag{A1.2}$$

En la figura A1.3 se muestra la gráfica correspondiente a las *"splines"* de grado uno respectivamente.

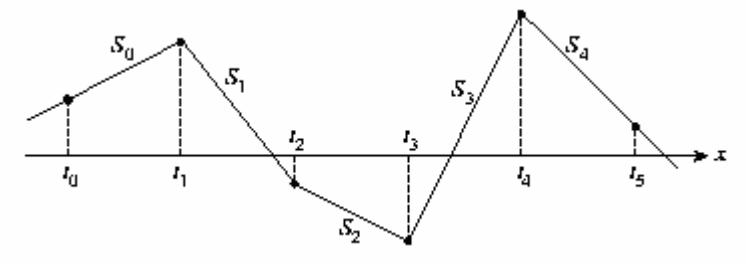


Figura A1.3 *"Spline"* de grado uno.

- ***"Spline" cúbica***

La *"spline"* cúbica (ver figura A1.4) $(k=3)$ es la *"spline"* más empleada, debido a que proporciona un excelente ajuste a los puntos tabulados y su cálculo no es excesivamente complejo.

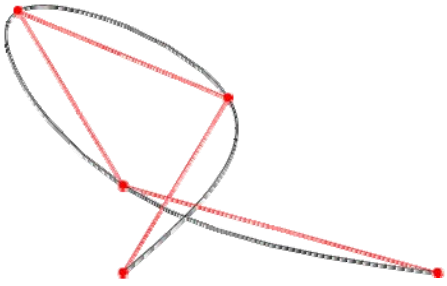


Figura A1.4 *"Spline"* cúbica.

Sobre cada intervalo $[t_0,t_1],[t_1,t_2],\ldots,[t_{n-1},t_n]$, S está definido por un polinomio cúbico diferente. Sea S_i el polinomio cúbico que representa a S en el intervalo $[t_i,t_{i+1}]$, por tanto:

$$S(x)=\begin{cases} S_0(x) & x\in[t_0,t_1) \\ S_1(x) & x\in[t_1,t_2) \\ \vdots & \vdots \\ S_{n-1}(x) & x\in[t_{n-1},t_n) \end{cases} \qquad \text{(A1.3)}$$

Los polinomios S_{i-1} y S_i interpolan el mismo valor en el punto t_i, es decir, se cumple:

$$S_{i-1}(t_i)=y_i=S_i(t_i) \qquad (1\le i\le n-1) \qquad \text{(A1.4)}$$

por lo que se garantiza que S es continuo en todo el intervalo. Además, se supone que S' y S'' son continuas, condición que se emplea en la deducción de una expresión para la función de la *"spline"* cúbica.

Aplicando las condiciones de continuidad de la *"spline"* S y de las derivadas primera S' y segunda S'', es posible encontrar la expresión analítica de la *"spline"*. La expresión resultante es:

$$S_i(x)=\frac{z_i}{6h_i}(t_{i+1}-x)^3+\frac{z_{i+1}}{6h_i}(x-t_i)^3+\left(\frac{y_{i+1}}{h_i}+\frac{z_{i+1}h_i}{6}\right)(x-t_i)+\left(\frac{y_i}{h_i}-\frac{z_ih_i}{6}\right)(t_{i+1}-x) \qquad \text{(A1.5)}$$

En la expresión anterior, $h_i=t_{i+1}-t_i$ y $z_0,z_1,\ldots,z_n$ son incógnitas. Para determinar sus valores, se utilizan las condiciones de continuidad que deben cumplir estas funciones. El resultado es:

$$h_{i-1}z_{i-1}+2(h_i+h_{i-1})z_i+h_iz_{i+1}=\frac{6}{h_{i-1}}(y_{i+1}-y_i)-\frac{6}{h_{i-1}}(y_i-y_{i-1}) \qquad \text{(A1.6)}$$

La ecuación anterior, con $i=1,2,\ldots,n-1$ genera un sistema de $n-1$ ecuaciones lineales con $n+1$ incógnitas $z_0,z_1,\ldots,z_n$. Se puede elegir z_0 y z_1 de forma arbitraria y resolver el sistema de ecuaciones resultante para obtener los valores de $z_1,z_2,\ldots,z_{n-1}$. Una elección especialmente adecuada es hacer $z_0=z_1=0$. La función *"spline"* resultante se denomina *"spline"* cúbica natural y el sistema de ecuaciones lineal expresado en forma matricial es:

$$\begin{pmatrix} u_1 & h_1 & & & & \\ h_1 & u_2 & h_2 & & & \\ & h_2 & u_3 & h_3 & & \\ & \ddots & \ddots & \ddots & & \\ & & h_{n-3} & u_{n-2} & h_{n-2} & \\ & & & h_{n-2} & u_{n-1} & \end{pmatrix}\begin{pmatrix} z_1 \\ z_2 \\ z_3 \\ \vdots \\ z_{n-2} \\ z_{n-3} \end{pmatrix}=\begin{pmatrix} v_1 \\ v_2 \\ v_3 \\ \vdots \\ v_{n-2} \\ v_{n-1} \end{pmatrix} \qquad \text{(A1.7)}$$

donde:

$$h_i = t_{i+1} - t_i \tag{A1.8}$$

$$u_i = 2(h_i + h_{i-1}) - \frac{h_{i-1}^2}{u_{i-1}} \tag{A1.9}$$

$$b_i = \frac{6}{h_i}(y_{i+1} - y_i) \tag{A1.10}$$

$$v_i = b_i - b_{i-1} - \frac{h_{i-1}v_{i-1}}{u_{i-1}} \tag{A1.11}$$

El valor de la *"spline" S* en un punto x cualquiera interpolado se puede calcular de forma eficiente empleando la siguiente expresión:

$$S_i(x) = y_i + (x - t_i)[c_i + (x - t_i)[b_i + (x - t_i)a_i]] \tag{A1.12}$$

en donde

$$a_i = \frac{1}{6h_i}(z_{i+1} - z_i) \tag{A1.13}$$

$$b_i = \frac{z_i}{2} \tag{A1.14}$$

$$c_i = -\frac{h_i}{6}z_{i+1} - \frac{h_i}{3}z_i + \frac{1}{h_i}(y_{i+1} - y_i) \tag{A1.15}$$

- ***"Thin-Plate spline"*** (ver capítulo 2)

- ***"B-spline"***

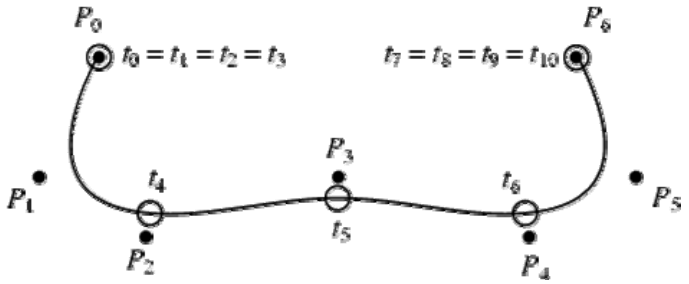


Figura A1.5 *"B-Spline".*

Es una generalización de las curvas Bézier (ver figura A1.5). Sea un vector conocido como el vector nudo a ser definido $T=(t_0,t_1,\ldots,t_n)$ donde T es una secuencia no decreciente con $t_i \in [0,1]$, y se definen los puntos de control $P_0,P_1,\ldots,P_n$. Se define el grado como $p=m-n-1$. Los nudos $t_{p+1},\ldots,t_{n-p-1}$ son llamados nudos internos.

La función base se define como

$$N_{i,0}(t)=\begin{cases}1 & \text{Si } t_1 \le t < t_{i+1} \text{ y } t_i < t_{i+1} \\ 0 & \text{De otra forma}\end{cases} \qquad \text{(A1.16)}$$

$$N_{i,p}(t)=\frac{t-t_i}{t_{i+p}-t_i}N_{i,p-1}(t)+\frac{t_{i+p+1}-t}{t_{i+p+1}-t_{i+1}}N_{i+1,p-1}(t) \qquad \text{(A1.17)}$$

Luego, la curva definida por

$$C(t)=\sum_{i=0}^{n} P_i N_{i,p}(t) \qquad \text{(A1.18)}$$

es la *"B-Spline"*. Existen dos tipos específicos, la *"B-Spline"* no periódica (el primer nudo $p+1=0$ y el último $p+1=1$) y la *"B-Spline"* uniforme (donde los nodos internos están igualmente espaciados). Una *"B-Spline"* sin nudos internos es una curva Bézier.

- ***Curva Bézier***

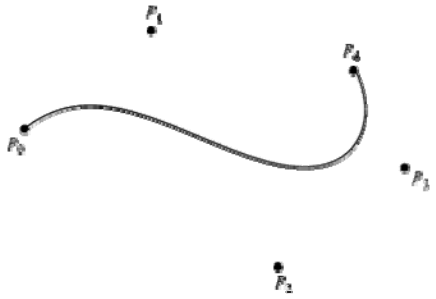


Figura A1.6 Curva Bézier.

Dado un conjunto de $n+1$ puntos de control $P_0,P_1,\ldots,P_n$, la curva correspondiente Bézier (o curva Bernstein-Bézier, ver figura A1.6) es dada por

$$C(t) = \sum_{i=0}^{n} P_i B_{i,n}(t) \qquad \text{(A1.19)}$$

Donde $B_{i,n}(t)$ son los polinomios Bernstein y $t \in [0,1]$. Una curva de Bézier racional se define como

$$C(t) = \frac{\sum_{i=0}^{n} B_{i,p}(t) w_i P_i}{\sum_{i=0}^{n} B_{i,p}(t) w_i} \qquad \text{(A1.20)}$$

Donde p es el orden, $B_{i,p}$ son los polinomios Bernstein, P_i son los puntos de control, y el peso w_i de P_i es el último punto homogéneo P_i^w. Estas curvas son cerradas bajo la perspectiva de transformación, y pueden ser representadas exactamente en secciones cónicas. La propiedad "disminución de la variación" de estas curvas es que ninguna línea puede tener más intersecciones con una curva Bézier que con la curva obtenida al tener puntos consecutivos unidos con segmentos de líneas rectas.

Una buena propiedad de estas curvas es que pueden ser trasladadas y rotadas implementando estas operaciones a los puntos de control. La desventaja de estas curvas es que son inestables numéricamente para muchos puntos de control, y el hecho de que al mover un solo punto de control cambia la forma global de la curva. Lo anterior se puede evitar parchando suavemente con curvas Bézier de bajo orden. Una generalización de las curvas Bézier es la *"B-spline"*.

- ***Curva NURBS***

La curva "*Non Uniform Rational Bézier Spline*" (NURBS) es una *"B-Spline"* racional no uniforme definida por

$$C(t) = \frac{\sum_{i=0}^{n} N_{i,p}(t) w_i P_i}{\sum_{i=0}^{n} N_{i,p}(t) w_i} \qquad \text{(A1.21)}$$

Donde p es el orden $N_{i,p}$ son la funciones base de *"B-Spline"*, P_i son los puntos de control, y el peso w_i de P_i es el último punto homogéneo P_i^w. Estas curvas son cerradas bajo la perspectiva de transformación, y pueden ser representadas exactamente en secciones cónicas.

Apéndice 2

RANSAC

De acuerdo con [13], el algoritmo llamado *"RANdom SAmple Consensus"* (RANSAC) se explica de la siguiente forma: dado un modelo que requiere un mínimo de n puntos para instanciar sus parámetros libres, y un conjunto de P puntos tal que el número de puntos en P es mayor que n, aleaotriamente se selecciona un subconjunto $S1$ de n puntos de P y se instancia el modelo. Se usa el modelo instanciado $\mathrm{M1}$ para determinar el subconjunto de $S1^*$ de puntos en P que tienen algunos errores de tolerancia de $\mathrm{M1}$. El conjunto $S1^*$ es llamado el conjunto de consenso de $S1$.

Si el número de puntos en $S1^*$ es mayor que algún umbral t, el cual es una función del número de errores burdos estimados en P, se utiliza $S1^*$ para calcular (tal vez con mínimos cuadrados) un nuevo modelo $\mathrm{M1}^*$.

Si el número de puntos en $S1^*$ es menor que t, se selecciona aleaotriamente un nuevo subconjunto $S2$ y se repite el proceso anterior. Si, después de cierto número predeterminado de intentos, no se encuentra un conjunto de consenso con t o más miembros, se resuelve el modelo con el conjunto de consenso más grande o se termina en fallo.

A continuación se explica un ejemplo para que sea más claro. El problema es el siguiente, dado un conjunto de puntos 2D, encontrar la línea que minimiza la suma de las distancias cuadradas perpendiculares (regresión ortogonal) sujeto a la condición de que ninguno de los puntos válidos se desvíe de esta línea más de t unidades. Esto plantea dos problemas: por un lado, una línea en la que encajen los datos y por otro una clasificación de los datos en puntos válidos y puntos no válidos o correspondencias incorrectas. Una característica del RANSAC es que puede manejar las situaciones en las que hay una gran cantidad de puntos no válidos.

La forma en que lo resuelve es seleccionando dos puntos aleaotriamente, estos puntos definen una línea. El soporte para esta línea es medida por el número de puntos que encajan en ella a cierto umbral de distancia. Esta selección aleatoria se repite cierto número de veces y se selecciona la que hace encajar el mayor número de puntos. La intuición dice que si uno de los puntos es un punto no válido, entonces la línea no ganará mucho soporte.

La muestra aleatoria consiste en un subconjunto de datos mínimo, en este caso dos puntos son suficientes para determinar el modelo. El número de muestras n se puede escoger lo suficientemente grande para asegurar con una probabilidad p que al menos una de las muestras aleatorias de s puntos es libre de puntos no válidos. Usualmente p se escoge en un 0.99. Si w es la probabilidad de que cualquier punto seleccionado es un punto válido, así $\varepsilon = 1 - w$ es la probabilidad de que sea un punto no válido. Entonces, por lo menos n selecciones (cada una de s puntos) son requeridas, donde $(1 - w^s)^n = 1 - p$, así que

$$n = \log(1 - p) / \log(1 - (1 - \varepsilon))^s \qquad \text{(A2.1)}$$

El umbral, que en este caso fue la distancia, depende del modelo y usualmente se selecciona empíricamente.

Apéndice 3

Selección proporcional o por ruleta

La selección proporcional o por ruleta se aplica en algoritmos genéticos para seleccionar a los individuos. En esta selección se suma la calificación de todos los individuos y esta suma se considera el 100% de una circunferencia. Luego, a cada individuo se le asigna el trozo que le corresponde de ésta según su aportación a la suma de las calificaciones.

Si se considera que la circunferencia es una ruleta, y se coloca una lengüeta que roce el borde de ella (Figura Ap3.1), la probabilidad de que dicha lengüeta quede en el arco correspondiente al individuo de comportamiento q_i, cuando la ruleta se detenga tras realizar algunos giros, es:

$$P(i) = \frac{q_i}{\sum_j q_j} \qquad \text{(A3.1)}$$

Lo que es proporcional al comportamiento (q_i) del individuo

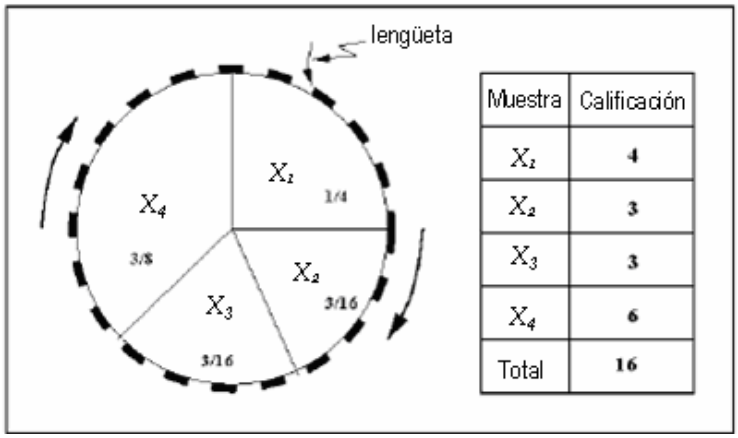


Figura Ap3.1 Selección proporcional o por ruleta.

Apéndice 4

Descomposición del valor singular

O SVD por sus siglas en inglés (Singular Value Decomposition), es una herramienta común para resolver sistemas de ecuaciones, y para diagonalizar matrices cuando el sistema es singular, o numéricamente muy próximo del singular – el error acumulado del método numérico por la precisión finita de los cálculos de una máquina es suficiente muchas veces para convertir sistemas no singulares en sistemas singulares –. En estos casos, se puede hacer una resolución por mínimos cuadrados del sistema, o plantearse la resolución del sistema por SVD.

Cualquier matriz **A**, sea singular o no, puede ser descompuesta en un producto de tres matrices, de la forma:

$$\mathbf{A} = \mathbf{U}\mathbf{V}\mathbf{D}^{\mathbf{T}} \qquad \text{(A4.1)}$$

Donde **U** y **V** son matrices ortogonales, por lo que sus inversas son iguales a sus transpuestas; y **D** es una matriz diagonal.

La aplicación más común de la SVD es el cálculo rápido de la inversa de una matriz para resolver un sistema de ecuaciones. La inversa de la matriz **A**, conocida su descomposición SVD, es:

$$\mathbf{A}^{-1} = \mathbf{V}\mathbf{D}^{-1}\mathbf{U}^{\mathbf{T}} \qquad \text{(A4.2)}$$

El cómputo de la inversa de la matriz **D** es fácil de calcular; aplicando la propiedad de las matrices diagonal por la que la inversa de una matriz diagonal se obtiene invirtiendo los elementos diagonales de dicha matriz. Esto funciona si la matriz no es singular, lo que significa en la práctica que todos los elementos diagonales de la matriz **D** son distintos de 0; lo que no es caso ni en la matriz fundamental ni en la matriz esencial, que son matrices singulares - y que en particular tendrán un elemento de **D** igual a 0.

SVD es interesante para este trabajo porque define un espacio ortogonal en $\mathbf{U}$ cuyo número de vectores directores coincide con el rango de la matriz $\mathbf{A}$, y este espacio define la parte no-singular de la matriz, y un espacio ortogonal en $\mathbf{V}$ cuyo número de vectores directores coincide con la dimensión de la matriz $\mathbf{A}$ menos su rango – es decir, el número de ceros en la diagonal de la matriz $\mathbf{D}$ –, y este espacio define la parte singular de la matriz.

4.1 Propiedades de la SVD

Las propiedades más importantes de la SVD son:

- Sea $\mathbf{A} = \mathbf{UDV}^{\mathbf{T}}$ una descomposición SVD, σ_1 el mayor elemento diagonal de $\mathbf{D}$ y σ_n el menor elemento diagonal de $\mathbf{D}$. Se llama grado de singularidad al cociente σ_1/σ_n ; y si la inversa del grado de singularidad es del orden de magnitud que la precisión de la máquina, se denomina a la matriz mal condicionada, y se puede considerar singular.

- Sea $\mathbf{A} = \mathbf{UDV}^{\mathbf{T}}$ una descomposición SVD, y $\mathbf{A}$ una matriz cuadrada no singular. La inversa de $\mathbf{A}$ será:

$$\mathbf{A}^{-1} = \mathbf{VD}^{-1}\mathbf{U}^{\mathbf{T}} \qquad \text{(A4.3)}$$

- Sea $\mathbf{A} = \mathbf{UDV}^{\mathbf{T}}$ una descomposición SVD, y $\mathbf{A}$ una matriz cuadrada singular. La pseudoinversa de $\mathbf{A}$ será:

$$\mathbf{A}^{+} = \mathbf{VD}_{\mathbf{0}}^{-1}\mathbf{U}^{\mathbf{T}} \qquad \text{(A4.4)}$$

Donde $\mathbf{D_0}$ es una matriz en la que se sustituyen los elementos de la diagonal que valgan cero por ∞; lo que es lo mismo que decir que los elementos de $\mathbf{D_0}^{-1}$ que valgan ∞ serán sustituidos por cero.

- Los autovalores no nulos de $\mathbf{A}^{\mathbf{T}}\mathbf{A}$ y de $\mathbf{AA}^{\mathbf{T}}$ son los cuadrados de los valores singulares no nulos de $\mathbf{A}$.

- Sea $\mathbf{A} = \mathbf{UDV}^{\mathbf{T}}$ una descomposición SVD, las columnas de $\mathbf{U}$ son los autovectores de $\mathbf{AA}^{\mathbf{T}}$ y las columnas de $\mathbf{V}$ son los autovectores de $\mathbf{A}^{\mathbf{T}}\mathbf{A}$.

- Sea $\mathbf{A} = \mathbf{UDV}^{\mathbf{T}}$ una descomposición SVD, todo elemento de la diagonal σ_k cumple que:

$$\mathbf{A}V_k = \sigma_k U_k \qquad \text{(A4.5)}$$

Y que :

$$\mathbf{A}^{\mathbf{T}} U_k = \sigma_k V_k \qquad \text{(A4.6)}$$

Siendo:

- U_k la columna de **U** correspondiente a σ_k
- V_k la columna de **V** correspondiente a σ_k

Glosario

Acelerómetro

Es un instrumento destinado a medir aceleraciones.

Algoritmo "Expectation – Maximization"

Es un algoritmo para encontrar la máxima probabilidad estimada de parámetros en modelos probabilísticos, donde el modelo depende de variables latentes (es decir, variables que son inferidas a partir de otras variables observadas y medidas directamente). Alterna entre un paso de expectativa (E), el cual calcula una expectativa de la probabilidad, incluyendo las variables latentes como si fueran observadas; y un paso de maximización (M), el cual calcula la máxima probabilidad estimada de los parámetros, minimizando la probabilidad esperada encontrada en el paso E. Los parámetros encontrados en el paso M son usados para empezar otro paso E, y el proceso se repite.

Algoritmo "Iterative Closest Point"

Es un algoritmo empleado para hacer corresponder dos conjuntos de puntos, estimando, iterativamente, la transformación (translación y rotación) entre ellos, para lo cual, asocia los puntos con el criterio del vecino más cercano, estima los parámetros usando una función de mínimos cuadrados, transforma los puntos usando los parámetros estimados, reasocia los puntos e itera. Ese un algoritmo simple usado para aplicaciones en tiempo real.

Baliza

Es un objeto señalizador, utilizado para indicar un lugar geográfico o una situación de peligro potencial.

Codificador

Es un circuito combinacional que presenta la salida en el código binario correspondiente a la entrada activada.

Convolución

Es un operador matemático definido como la integral del producto de dos funciones, donde una de ellas ha sido desplazada e invertida. Se denota como $*$. En un sistema unidimensional, se dice que $g(x)$ convoluciona con $f(x)$ cuando:

$$f(x) * g(x) = \int_{-\infty}^{+\infty} f(x')\, g(x - x')\, dx' \qquad \text{(G.1)}$$

Donde x' es una variable de integración.

En el caso de una función continua bidimensional, la convolución de $f(x, y)$ por $g(x, y)$ será:

$$f(x, y) * g(x, y) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} f(x', y')\, g(x - x', y - y')\, dx' dy' \qquad \text{(G.2)}$$

Para el caso de los sistemas discretos bidimensionales (como lo son las imágenes digitales) la convolución de de $f(x, y)$ por $g(x, y)$ y en la que $g(x, y)$ es una matriz de *m* filas por *n* columnas será:

$$f(x, y) * g(x, y) = \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} f(m, n)\, g(x - i, y - j) \qquad \text{(G.3)}$$

En donde, $x = 0, 1, 2, \ldots, m$ y $y = 0, 1, 2, \ldots, n$.

Curvas de nivel

Es una línea que une todos los puntos, que tienen igualdad de condiciones, en un mapa. Para el caso de un mapa topográfico, es cada una de las curvas que materializan una sección horizontal de relieve.

Distancia Hausdorff

Mida la distancia que existe entre dos subconjuntos compactos de un espacio métrico (el cual es un espacio topológico en donde la distancia entre puntos está definida).

Giróscopo

Es un aparato constituido principalmente por un volante pesado que gira rápidamente y tiende a conservar el plano de rotación reaccionando contra cualquier fuerza que lo aparte de dicho plano (ej. volante de un barco).

Filtro Kalman

Es un conjunto de ecuaciones matemáticas, que proveen una solución recursiva eficiente del método de mínimos cuadrados. Esta solución permite calcular un estimador lineal de un proceso, en cada momento del tiempo, con base en la información disponible en el momento $t-1$, y actualizarlo, con la información adicional disponible en el momento *t*.

Matriz antisimétrica

Se define una matriz antisimétrica por la izquierda $[\mathbf{A}]^{\times}$ de una matriz **A**, a aquella matriz tal que:

$$\mathbf{A} \times \mathbf{B} = [\mathbf{A}]^{\times} \mathbf{B} \tag{G.4}$$

Por ejemplo, si se tiene la matriz de 3×1:

$$\mathbf{A} = \begin{pmatrix} a_0 \\ a_1 \\ a_2 \end{pmatrix} \tag{G.5}$$

Su antisimétrica $[\mathbf{A}]^{\times}$ será:

$$\mathbf{A} = \begin{pmatrix} 0 & -a_2 & a_1 \\ a_2 & 0 & -a_0 \\ -a_1 & a_0 & 0 \end{pmatrix} \tag{G.6}$$

Matriz identidad

Son matrices cuadradas (tienen el mismo número de columnas y de filas) con todos los elementos excepto los elementos de la diagonal principal, que valen todos 1. Por ejemplo:

$$|1| \qquad \begin{vmatrix} 1 & 0 \\ 0 & 1 \end{vmatrix} \qquad \begin{vmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{vmatrix}$$

De forma matemática:

$$\mathbf{A} = \mathbf{AI} \tag{G.7}$$

Matriz Jacobiana

Es una matriz de orden $m \times n$, que tiene como elementos a las derivadas parciales (si existen) de la función.

Matriz inversa

La matriz inversa de $\mathbf{A}$ se denota por $\mathbf{A}^{-1}$ y satisface la igualdad: $\mathbf{A}\mathbf{A}^{-1} = \mathbf{A}^{-1}\mathbf{A} = \mathbf{I}$. Esta viene dada por:

$$\mathbf{A}^{-1} = \frac{1}{|\mathbf{A}|}(\text{adj}(\mathbf{A}))^{\text{T}} \qquad \text{(G.8)}$$

donde

$|\mathbf{A}|$ = determinante de $\mathbf{A}$,

$\text{adj}(\mathbf{A})$ = matriz adjunta de $\mathbf{A}$.

Nótese que $\mathbf{A}$ tiene inversa si no es una matriz singular.

Matriz no singular

Una matriz $\mathbf{A}$ de dimensiones $n \times n$ se dice que es invertible, inversible o no singular si existe una matriz $\mathbf{B}$ de dimensiones $n \times n$ tal que

$$\mathbf{AB} = \mathbf{BA} = \mathbf{I}_n \qquad \text{(G.9)}$$

donde $\mathbf{I}_n$ denota la matriz identidad de orden n (dimensiones $n \times n$) y el producto utilizado es el producto de matrices usual. En este caso, la matriz $\mathbf{B}$ es única y se dice que es la inversa de $\mathbf{A}$. Esto se denota por $\mathbf{A}^{-1}$. Una matriz no invertible se dice que es singular.

Matriz pseudoinversa

Se define como pseudoinversa por la derecha de una matriz $\mathbf{A}$, que se denota como $\mathbf{A}^{+}$, a una matriz que multiplicada por la derecha por $\mathbf{A}$ da la matriz diagonal con todos los elementos diagonales iguales a 1 – matriz $\mathbf{I}$ -; es decir:

$$\mathbf{I} = \mathbf{A}\mathbf{A}^{+} \qquad \text{(G.10)}$$

Esta definición de matriz pseudoinversa permite que la matriz **A** no sea cuadrada. Esto es fundamental para esta tesis, ya que las matrices de proyección no son cuadradas. Si se tienen las matrices de proyección M1 y M2, estas tendrán como dimensión 4×3. Esto significa que la matriz pseudoinversa por la derecha debe ser una matriz con dimensiones 3×4. Se calcula la pseudoinversa con de *A* con la ecuación:

$$\mathbf{A}^{+} = \mathbf{A}^{\mathbf{T}} \left[\mathbf{A}^{\mathbf{T}} \mathbf{A} \right]^{-1} \qquad \text{(G.11)}$$

Como la matriz $\left[\mathbf{A}^{\mathbf{T}} \quad \mathbf{A} \right]$ es cuadrada, suponiendo que sea una matriz regular, es una matriz que pude ser invertida, y su inversa es única; por lo que el cálculo de esta expresión no debe tener ningún problema adicional.

Matriz Singular

Una matriz cuadrada *A* de orden *n* es singular si su determinante es nulo. En tal caso se dice que dicha matriz no tiene inversa.

Mínimos cuadrados

Es una técnica clásica para la estimación de parámetros que optimiza (de acuerdo a una función objetivo específica) el ajuste de una descripción funcional (modelo) a todos los datos presentados. Esta técnica no tiene mecanismos internos para detectar y rechazar errores burdos. Es una técnica de porcentajes que se basan en el supuesto de que la máxima desviación esperada de cualquier conjunto de datos del modelo asumido es una función directa del tamaño de los datos, y así, sin importar el tamaño del conjunto de datos, siempre habrá suficientes valores buenos para suavizar cualquier desviación burda.

Multiplicación de Matrices

La multiplicación de matrices supone varios productos sencillos y sumas de los elementos de la matriz. Se pueden multiplicar dos matrices **A** y **B** si el número de columnas de la primera matriz **A** es igual al número de filas de la segunda matriz **B**. La multiplicación de matrices no es conmutativa. Cuando se multiplican dos matrices se obtiene como resultado otra matriz. Esta matriz producto tendrá el mismo número de filas que la primera de las matrices que se multiplican y el mismo número de columnas que la segunda.

Los elementos de la matriz producto **C** se expresan en función de los elementos de las matrices **A** por **B** mediante la siguiente fórmula:

$$\mathbf{C}(i,k) = \sum_{j} \mathbf{A}(i,j)\mathbf{B}(j,k) \qquad \text{(G.12)}$$

La multiplicación de matrices es una operación asociativa. Esto significa que si se tienen varias matrices para multiplicar a la vez, no importa cuales se multipliquen primero. De forma matemática: ***s***

$$\mathbf{A(BC) = (AB)C} \qquad \text{(G.13)}$$

Ortonormal

Una matriz cuadrada $\mathbf{A}$ con matriz transpuesta $\mathbf{A^T}$ y matriz inversa $\mathbf{A^{-1}}$ es ortogonal, siempre que $\mathbf{A^T = A^{-1}}$.

Paralaje

Es la desviación angular de la posición aparente de un objeto, dependiendo del punto de vista elegido. Determina que lo captado a través del visor no coincide con la imagen capturada a través del objetivo de la cámara. Este desplazamiento por paralaje puede ser vertical, horizontal o ambos a la vez.

Rango de una matriz

Es el número de líneas de esa matriz (filas o columnas) que son linealmente independientes.

Una línea es linealmente dependiente de otra u otras cuando se puede establecer una combinación lineal entre ella y es linealmente independiente de otra u otras cuando no se puede establecer una combinación lineal entre ellas.

Como mínimo, el rango de una matriz siempre será 1, salvo para la matriz nula, cuyo rango es cero. Para poder calcular el rango de una matriz ésta no tiene por que ser necesariamente cuadrada. Una matriz cuadrada de orden n, como máximo su rango es n. Una matriz cuadrada de orden n es inversible (regular) si el rango es n. Es decir, cuando las filas (columnas) son linealmente independientes.

Dos matrices $\mathbf{A}$ y $\mathbf{B}$ son equivalentes ($\mathbf{A \sim B}$) si tienen el mismo rango.

Tacómetro

Es un dispositivo para medir la velocidad de giro de un eje, normalmente de un motor.

Transformada de Hough

Es un algoritmo para aislar características de formas particulares (i.e. líneas, círculos, elipses) en una imagen. Tiene dos funciones principales, por un lado, detectar las características para las cuales tiene una descripción paramétrica u otra descripción, y por otro, indicar cuántas de éstas existen en la imagen. Pude ser empleada en aplicaciones donde no es posible hacer una descripción analítica simple de una característica.

Triangulación Delaunay

Para un conjunto de puntos en el plano, es una triangulación tal que ningún punto está dentro de la circunferencia de ningún triángulo generado en el conjunto.

Vectores propios

Los vectores propios (o eigenvector del alemán eigen que significa "propio, inherente, característico") de un operador lineal son los vectores diferentes de cero que, cuando son transformados por el operador, dan lugar a un múltiplo escalar de sí mismo. El escalar entonces se llama el valor propio asociado al vector propio.

Por ejemplo, considere la matriz

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & -1 \\ 1 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix}$$

que representa un operador lineal $\mathbb{R}^3 \to \mathbb{R}^3$. Uno puede comprobar que

$$\mathbf{A}\begin{bmatrix} 1 \\ 1 \\ -1 \end{bmatrix} = \begin{bmatrix} 2 \\ 2 \\ -2 \end{bmatrix} = 2\begin{bmatrix} 1 \\ 1 \\ -1 \end{bmatrix}$$

y por lo tanto 2 es un valor propio de $\mathbf{A}$ y hemos encontrado un vector propio correspondiente.

Bibliografía

[Aguilar, W. et al., 2006] Aguilar, W., Frauel, Y. and Escolano, F., "Reconocimiento de objetos basado en la correspondencia estructural de características locales", Tesis de Maestría, Universidad Nacional Autónoma de México, México, 2006.

[Andrews Space & Technology, 2007a] Andrews Space & Technology, "GLONASS - Summary", last update: 01/01/1999, acc. year 2007, http://www.spaceandtech.com/spacedata/constellations/glonass_consum.shtml.

[Armangue, X. et al., 2003] Armangue, X., Araujo, H. and Salvi, J., "A review on egomotion by means of differential epipolar geometry applied to the movement of a mobile robot", *Pattern recognition*, vol. 36, no. 12, pp. 2927-2944, 2003.

[Audette, R. et al., 2000] Audette, R., Balthazaar, J., Dunk, C. and Zelek, J., "A Stereo-vision System for the Visually Impaired", *Technical Report,* School of Engineering, University of Guelph , 2000.

[Baird, H.S., 1985] Baird, H.S., "Model-based image matching using location", MIT Press, 1985.

[Baker, S. and Nayar, S., 1998] Baker, S. and Nayar, S., "A Theory of Catadioptric Image Formation", *ICCV '98: Proceedings of the Sixth IEEE International Conference on Computer Vision*, pp. 35-42, Bombay, India, 1998.

[Baker, S. and Nayar, S., 1999] Baker, S. and Nayar, S., "A theory of single-viewpoint catadioptric image formation", *International Journal of Computer Vision (IJCV)*, vol. 35, no. 2, pp. 175-196, 1999.

[Besl, P.J. and McKay, N.D., 1992] Besl, P.J. and McKay, N.D., "A Method for Registration of 3-D Shapes", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 2, pp. 239-256, 1992.

[Bookstein, F.L., 1989] Bookstein, F.L., "Principal Warps: Thin-Plate Splines and the Decomposition of Deformations", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, no. 6, pp. 567-585, 1989.

[Brown, B. and Rusinkiewicz, S., 2004] Brown, B. and Rusinkiewicz, S., "Non-Rigid Range-Scan Alignment Using Thin-Plate Splines", *3DPVT '04: Proceedings of the 2nd International Symposium on 3D Data Processing, Visualization, and Transmission*, pp. 759-765, Thessaloniki, Greece, 2004.

[Bruss, A. and Horn, B.K.P., 1983] Bruss, A. and Horn, B.K.P., "Passive Navigation", *Computer Vision, Graphics, and Image Processing*, vol. 21, no. 1, pp. 3-20, 1983.

[Burgard, W. et al., 1996] Burgard, W., Fox, D., Hennig, D. and Schmidt, T., "Estimating the absolute position of a mobile robot using position probability grids", *AAAI '96: Proceedings of the Thirteenth National Conference on Artificial Intelligence*, vol. 2, pp. 896-901, Portland, Oregon, 1996.

[Chengke, W. et al., 2000] Chengke, W., Yong, L. and Zezhi, C., "Robust estimation of epipolar geometry and hierarchical reconstruction of 3D surface", *ACCV '00: Proceedings of the Fourth Asian Conference on Computer Vision*, pp. 899-904, Taipei, Taiwan, 2000.

[Chui, H. and Rangarajan, A., 2000] Chui, H. and Rangarajan, A., "A New Algorithm for Non-Rigid Point Matching", *CVPR '00: Proceedings of the 2000 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 44-51, San Juan, Puerto Rico, 2000.

[Chui, H. and Rangarajan, A., 2003] Chui, H. and Rangarajan, A., "A new point matching algorithm for non-rigid registration", *Computer Vision and Image Understanding*, vol. 89, no. 2-3, pp. 114-141, 2003.

[Cootes, T.F. et al., 1995] Cootes, T.F., Taylor, C.J., Cooper, D.H. and Graham, J., "Active shape models: Their training and application", *Computer Vision and Image Understanding*, vol. 61, no. 1, pp. 38-59, 1995.

[Coughlan, J. et al., 2006] Coughlan, J., Manduci, R. and Shen. H., “Cell Phone-based Wayfinding for the Visually Impaired”, *IMV ’06: ECCV Workshop on Mobile Vision,* Graz, Austria, 2006.

[Cox, I., 1991] Cox, I., "Blanche - an experiment in guidance and navigation of an autonomous robot vehicle", *IEEE Transactions on Robotics and Automation*, vol. 7, no. 3, pp. 193-204, 1991.

[Cross, A.D.J. and Hancock, E.R., 1998] Cross, A.D.J. and Hancock, E.R., "Graph Matching With a Dual-Step EM Algorithm", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 11, pp. 1236-1253, 1998.

[Cui, N. et al., 1990] Cui, N., Weng, J. and Cohen, P., "Extended structure and motion analysis from monocular image sequences", *ICCV '90: Proceedings of the Third IEEE International Conference on Computer Vision*, pp. 222-229, Osaka, Japan, 1990.

[Daniilidis, K. and Nagel, H., 1993] Daniillidis, K. and Nagel, H., "The coupling of rotation and translation in motion estimation of planar surfaces", *CVPR '93: Proceedings of the 1993 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 188, New York City, NY, USA, 1993.

[Debrunner, C.H. and Ahuja, N.A., 1990] Debrunner, C.H. and Ahuja, N.A., "Direct data approximation based motion estimation algorithm", *ICPR '90: Proceedings of the 1990 IEEE International Conference on Pattern Recognition*, pp. 384-389, Atlantic City, NJ, USA, 1990.

[Demirdjian, D. and Horaud, R., 2000] Demirdjian, D. and Horaud, R., "Motion-Egomotion Discrimination and Motion Segmentation from Image-Pair Streams", *Computer Vision and Image Understanding*, vol. 78, no. 1, pp. 53-68, 2000.

[Donato, G. and Belongie, S., 2002] Donato, G. and Belongie, S., "Approximation Methods for Thin Plate Spline Mappings and Principal Warps", *ECCV '02: Proceedings of the 7th European Conference on Computer Vision-Part III*, pp. 21-31, Copenhagen, Denmark, 2002.

[Elfes, A., 1990] Elfes, A., "Sonar-based real-world mapping and navigation", *IEEE Journal of Robotics and Automation*, vol. 3, no. 3, pp. 233-249, 1990.

[European Space Agency,2007b] European Space Agency, "Galileo European Satellite Navigation System", last update: 04/10/2007, acc. year 2007, http://ec.europa.eu/dgs/energy_transport/galileo/index_en.htm.

[Faugeras, O., 1993] Faugeras, O., "Three-dimensional computer vision: a geometric viewpoint", MIT Press, 1993.

[Feldmar, J. and Ayache, N., 1996] Feldmar, J. and Ayache, N., "Rigid, affine and locally affine registration of free-form surfaces", *International Journal of Computer Vision (IJCV)*, vol. 18, no. 2, pp. 99-119, 1996.

[Fischler, M. and Bolles, R., 1981] Fischler, M. and Bolles, R., "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography", *Communications of the Association for Computing Machinery*, vol. 24, no. 6, pp. 381-395, 1981.

[Florack, L. et al., 1991] Florack, L., Ter Haar Romeny, B., Koenderink, J. and Viergever, M., "General Intensity Transformations and Second Order Invariants", *Proceedings of the Seventh Scandinavian Conference on Image Analysis*, pp. 338-345, Alborg, Denmark, 1991.

[Foo, P.S. et al., 2004] Foo, P.S., Harrison, M., Duchon, A., Warren, W.H. and Tarr, M.J., "Humans Follow Landmarks Over Path Integration", *Journal of Vision*, vol. 4, no. 8, pp. 892-892, 2004.

[Gold, S. et al., 1995] Gold, S., Lu, C.P., Rangarajan, A., Pappu, S. and Mjolsness, E., "New algorithms for 2D and 3D Point Matching: Pose Estimation and Correspondence", *Advances in Neural Information Processing Systems*, vol. 7, MIT Press, 1995.

[Gonzalez, J., 1996] Gonzalez, J., "Recovering Motion Parameters from a 2D Range Image Sequence", *ICPR '96: Proceedings of the 1996 IEEE International Conference on Pattern Recognition*, vol. 1, pp. 433, Los Alamitos, CA,USA, 1996.

[González, J. and Ollero, A., 1996] González, J. and Ollero, A., "Estimación de la Posición de un Robot Móvil", *Informática y Automática*, vol. 29, no. 4, pp. 3-18, 1996.

[González, J. et al., 1995] González, J., Stentz, A. and Ollero, A., "A Mobile Robot Iconic Position Estimator Using a Radial Laser Scanner", *Journal of Intelligent Robotics Systems (JIRS)*, vol. 13, no. 2, pp. 161-179, 1995.

[Goshen, L. et al., 2003] Goshen, L., Shimshoni, I., Anandan, P. and Keren, D., "Recovery of epipolar geometry as a manifold fitting problem", *ICCV '03: Proceedings Ninth IEEE International Conference on Computer Vision*, vol. 2, pp. 1321-1328, Nice, France, 2003.

[Haralick, R., 1988] Haralick, R., "A simplified linear optic-flow motion algorithm", *Computer Vision, Graphics, and Image Processing*, vol. 42, no. 3, pp. 334-344, 1988.

[Haralick, R. et al., 2000] Haralick, R., Joo, H., Lee, C., Zhuang, X., Vaidya, V. and Kim, M., "Pose Estimation From Corresponding Point Data", *IEEE Transactions on Systems, man, and Cybernetics*, vol. 19, no. 6, pp. 1426-1445, 1989.

[Harris, C. et al., 1992] Harris, C., Stephens, M., Sparks, E. and Pike, M., "DROID analysis of the NASA helicopter images", *Journal of Robotic Systems*, vol. 9, no. 66, pp. 773-785, 1992.

[Hartley, R. and Zisserman, A., 2004] Hartley, R. and Zisserman, A., "Multiple View Geometry in Computer Vision", Cambridge University Press, Second, 2004.

[Hinton, G.E. et al., 1992] Hinton, G.E., Williams, C.K.I. and Revow, M.D., "Adaptive Elastic Models for Hand-Printed Character Recognition", *Advances in Neural Information Processing Systems*, vol. 4, Morgan Kaufmann Publishers, Inc., 1992.

[Huttenlocher, D.P. et al., 1993] Huttenlocher, D.P., Klanderman, G.A. and Rucklidge, W.J., "Comparing images using the Hausdor distance", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, no. 9, pp. 850-863, 1993.

[Johnson, A. and Hebert, M., 1997] Johnson, A. and Hebert, M., "Object Recognition by Matching Oriented Points", *CVPR '97: Proceedings of the 1997 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 684-689, 1997.

[Kanatani, K., 1993] Kanatani, K., "3D interpretation of optical flow by renormalization", *International Journal of Computer Vision (IJCV)*, vol. 11, no. 3, pp. 267-282, 1993.

[Kimura, M. and Saito, H., 2002] Kimura, M. and Saito, H., "3D Reconstruction based on Epipolar Geometry", *MVA '00: IAPR Workshop on Machine Vision Applications*, pp. 578-581, Tokyo, Japan, 2002.

[Krotkov, E., 1989] Krotkov, E., "Mobile robot localization using a single image", *ICRA '89: Proceedings of the IEEE International Conference on Robotics and Automation*, vol. 2, pp. 978-983, Scottsdale, AZ, 1989.

[Lee, J.W. et al., 2000] Lee, J.W., You, S. and Neumann, U., "Large Motion Estimation for Omnidirectional Vision", *OMNIVIS '00: Proceedings of the IEEE Workshop on Omnidirectional Vision*, pp. 161Hilton Head, South Carolina, 2000.

[Lohmann, G. and von Cramon, D.Y., 2000] Lohmann, G. and von Cramon, D.Y., "Automatic labeling of the human cortical surface using sulcal basins", *Medical Image Analysis*, vol. 4, no. 3, pp. 179-188, 2000.

[Longuet-Higgins, H.C., 1981] Longuet-Higgins, H.C., "A computer algorithm for reconstruction a scene from two projections", *Nature*, pp. 133-135, 1981.

[Longuet-Higgins, H.C., 1984] Longuet-Higgins, H.C., "The reconstruction of a scene from two projections - configurations thet defeat the 8-point algorithm", *CAIA '84: Proceedings of the First Conference on Artificial Intelligence Applications*, pp. 395-397, Denver, CO, USA, 1984.

[Lowe, D., 2004] Lowe, D., "Distinctive image features from scale-invariant keypoints", *International Journal of Computer Vision (IJCV)*, vol. 60, no. 2, pp. 91-110, 2004.

[Lucas, B., 1985] Lucas, B., "Generalized image matching by the method of differences", PhD Thesis, *Computer Science Department*, Carnegie Mellon University, USA, 1985.

[MacLean, W. et al., 1994] MacLean, W., Jepson, A. and Frecker, R., "Recovery of egomotion and segmentation of independent object motion using the EM algorithm", *BMVC '94: Proceedings of the 5th British Machine Vision Conference*, pp. 13-16, York, UK, 1994.

[Megyesi, Z. and Chetverikov, D., 2003] Megyesi, Z. and Chetverikov, D., "Affine dense matching for wide baseline stereo", *Proceeding Grafika 2003*, pp. 109-114, Hungary, Budapest, 2003.

[Matsumoto, Y. et al., 2000] Matsumoto, Y., Ikeda, K., Inaba, M. and Inoue, H., "Exploration and navigation in corridor environment based on omni-view sequence", *IROS '00: Proceedings of the International Conference on Intelligent Robots and Systems*, pp. 1505-1510, Takamatsu, Japan, 2000.

[Matthies, L. et al., 1989] Matthies, L., Kanade, T. and Szeliski, R., "Kalman Filter-based Algorithms for Estimating Depth from Image Sequences", *International Journal of Computer Vision (IJCV)*, vol. 3, no. 3, pp. 209-238, 1989.

[Metaxas, D. et al., 1997] Metaxas, D., Koh, E. and Badler, N.I., "Multi-Level Shape Representation Using Global Deformations andLocally Adaptive Finite Elements", *International Journal of Computer Vision (IJCV)*, vol. 25, no. 1, pp. 49-61, 1997.

[Mikolajczyk, K. and Schmid, C., 2005] Mikolajczyk, K. and Schmid, C., "A performance evaluation of local descriptors", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 10, pp. 1615-1630, 2005.

[Muñoz, A. and González, J., 1996] Muñoz, A. and González, J., "Localizing Mobile Robots with a Single Camera in Structured Environments", *WAC '96: The first International Symposium on Intelligent, Automation and Control (ISIAC) in the World Automation Congress*, pp. 539-544, Montpellier, France, 1996.

[NAVCEN: United States Coast Guard's Navigation Cente, 2007c] NAVCEN: United States Coast Guard's Navigation Cente, "GPS FREQUENTLY ASKED QUESTIONS", last update: 04/21/2005, acc. year 2007, http://www.navcen.uscg.gov/faq/gpsfaq.htm.

[Pratt, W., 2001] Pratt, W., "Digital Image Processing: PIKS Inside", Wiley-Interscience, Third Edition, 2001.

[Prazdny,K., 1980] Prazdny,K., "Egomotion and relative depth map from optical flow", *Biological Cybernetics*, vol. 36, no. 2, pp. 87-102, 1980.

[Rohr, K. et al., 2001] Rohr, K., Stiehl, H.S., Sprengel, R., Buzug, T.M., Weese, J. and Kuhn, M.H., "Landmark-based elastic registration using approximating thin-plate splines", *IEEE Transactions on Medical Imaging*, vol. 20, no. 6, pp. 526-534, 2001.

[Roth, G. and Whitehead, A., 2000] Roth, G. and Whitehead, A., "Using Projective vision to Find Camera Positions in an Image Sequence", *VI '00: Vision Interface*, Montreal, Canada, 2000.

[Saez, J.M. et al., 2005] Saez, J.M., Escolano, F. and Penalver, A., "First Steps towards Stereo-based 6DOF SLAM for the Visually Impaired", *CVPR '05: Proceedings of the 2005 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 23, San Diego, CA, USA, 2005.

[Sclaroff, S. and Pentl, A., 1995] Sclaroff, S. and Pentl, A., "Modal Matching for Correspondence and Recognition", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, no. 6, pp. 545-561, 1995.

[Scott, G. and Longuet-Higgins, H.C., 1991] Scott, G. and Longuet-Higgins, H.C., "An algorithm for associating the features of two images", *Proceedings Royal Society of London*, vol. B244, pp. 21-26, Great Britain, 1991.

[Shapiro, L.S. and Brady, J.M., 1992] Shapiro, L.S. and Brady, J.M., "Feature-based correspondence: an eigenvector approach", *Image and Vision Computing*, vol. 10, no. 5, pp. 283-288, 1992.

[Shapiro, L.G. and Haralick, R.M., 1981] Shapiro, L.G. and Haralick, R.M., "Structural descriptions and inexact matching", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 3, no. 9, pp. 504-519, 1981.

[Se, S. et al., 2002] Se, S., Lowe, D. and Little, J., "Mobile robot localization and mapping with uncertainty using scale-invariant visual landmarks", *The International Journal of Robotics Research (IJRR)*, vol. 21, no. 8, pp. 735-758, 2002.

[Spetsakis, M. and Aloimonos, J., 1990] Spetsakis, M. and Aloimonos, J., "Structure from motion using line correspondences", *International Journal of Computer Vision (IJCV)*, vol. 4, no. 3, pp. 171-183, 1990.

[Stein, F. and Medioni, G., 1995] Stein, F. and Medioni, G., "Map-based localization using the panoramic horizon", *IEEE Transactions on Robotics and Automation*, vol. 11, pp. 892-896, 1995.

[Stockman, G., 1987] Stockman, G., "Object recognition and localization via pose clustering", *Computer Vision, Graphics, and Image Processing*, vol. 40, no. 3, pp. 361-387, 1987.

[Sugihara, K., 1988] Sugihara, K., "Location of robot using sparse visual information", *4th International Symposium on Robotics Research*, MIT Press, 1988.

[Sutherland, K. and Thompson, W., 1994] Sutherland, K. and Thompson, W., "Localizing in Unstructured Environments: Dealing with Errors", *IEEE Transactions on Robotics and Automation*, vol. 10, no. 6, pp. 740-754, 1994.

[Szeliski, R., 1988] Szeliski, R., "Estimating motion from sparse range data without correspondence", *ICCV '88: Proceedings of the Second IEEE International Conference on Computer Vision*, pp. 207-216, Tarpon Spring, FL, USA, 1988.

[Szeliski, R. and Lavallée, S., 1996] Szeliski, R. and Lavallée, S., "Matching 3-D anatomical surfaces with non-rigid deformations using octree-splines", *International Journal of Computer Vision (IJCV)*, vol. V18, no. 2, pp. 171-186, 1996.

[Tagare, H.D. et al., 1995] Tagare, H.D., O'Shea, D. and Rangarajan, A., "A geometric criterion for shape-based non-rigid correspondence", *ICCV '95: Proceedings of the Fifth International Conference on Computer Vision*, pp. 434-439, Cambridge, MA, USA, 1995.

[Tardos, J., 1990] Tardos, J., "Integración Multisensorial para Reconocimiento y Localización de Objetos en Robotica", Tesis doctoral, Universidad de Zaragoza, España, 1990.

[Tian, T. et al., 1996] Tian, T., Tomasi, C. and Heeger, D., "Comparison of Approaches to Egomotion Computation", *CVPR '96: Proceedings of the 1996 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 315-320, San Francisco, CA, USA, 1996.

[Tomasi, C. and Kanade, T., 1990] Tomasi, C. and Kanade, T., "Shape and motion without depth", *ICCV '90: Proceedings of the Third IEEE International Conference on Computer Vision*, pp. 91-95, Osaka, Japan, 1990.

[Tomasi, C. and Shi, J., 1993] Tomasi, C. and Shi, J., "Direction of heading from image deformation", *CVPR '93: Proceedings of the 1993 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 422-427, New York, NY, USA, 1993.

[Torr, P. and Murray, D.W., 1997] Torr, P. and Murray, D.W., "The development and comparison of robust methods for estimating the fundamental matrix", *International Journal of Computer Vision (IJCV)*, vol. 24, no. 3, pp. 271-300, 1997.

[Tsai, R.Y. and Huang, T.S., 1984] Tsai, R.Y. and Huang, T.S., "Uniqueness and estimation of 3-D motion parameters of rigid objects with curved surfaces", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 6, pp. 13-17, 1984.

[Ullman, S., 1989] Ullman, S., "Aligning pictorial descriptions: An approach to object recognition", *Cognition*, vol. 32, no. 3, pp. 193-254, 1989.

[University of Newcastle, 2007d] University of Newcastle, "What is GNSS", last update: 02/01/2005, acc. year 2007, http://ce-gw343.ncl.ac.uk/forum/fm_what_is_gnss.asp.

[Vassallo, R. et al., 2002] Vassallo, R., Santos-Victor, J. and Schneebeli, J., "A general approach for egomotion estimation with omnidirectional images", *OMNIVIS '02: Proceedings of the IEEE Workshop on Omnidirectional Vision*, pp. 97-103, Copenhagen, Denmark, 2002.

[Vincent, E. and Laganière, R., 2001] Vincent, E. and Laganière, R., "Matching Feature Points in Stereo Pairs: A Comparative Study of Some Matching Strategies", *Machine Graphics and Vision*, vol. 10, no. 3, pp. 237-259, 2001.

[Wahba, G., 1990] Wahba, G., "Spline Models for Observational", *CBMS-NSF Regional Conference Series in Applied Mathematics*, vol. 59, 1990.

[Wells, W., 1997] Wells, W., "Statistical approaches to feature-based object recognition", *International Journal of Computer Vision (IJCV)*, vol. 21, no. 1-2, pp. 63-98, 1997.

[Weng, J. et al., 1992] Weng, J., Ahuja, N. and Huang, T.S., "Motion and Structure from Image Sequences", Springer-Verlag, New York, Inc., 1992.

[Weng, J. et al., 1989] Weng, J., Ahuja, N. and Huang, T.S., "Optimal motion and structure information", *CVPR '89: Proceedings of the 1989 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 144-152, San Diego, CA, USA, 1989.

[Wexler, Y. et al., 2003] Wexler, Y., Fitzgibbon, A. and Zisserman, A., "Learning epipolar geometry from image sequences", *CVPR '03: Proceedings of the 2003 IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 209, Madison, WI, 2003.

[Winters, N. et al., 2000] Winters, N., Gaspar, J., Lacey, G. and Santos-Victor, J., "Omni-Directional Vision for Robot Navigation", *OMNIVIS '00: Proceedings of the IEEE Workshop on Omnidirectional Vision*, pp. 21, Hilton Head, South Carolina, 2000.

[Zabih, R. and Woodfill, J., 1994] Zabih, R. and Woodfill, J., "Non-parametric Local Transforms for Computing Visual Correspondence", *ECCV '94: Proceedings of the 3th European Conference on Computer Vision*, pp. 151-158, Stockholm, Sweden, 1994.

[Zhang, Z., 1998] Zhang, Z., "Determining the Epipolar Geometry and its Uncertainty: A Review", *International Journal of Computer Vision (IJCV)*, vol. 27, no. 2, pp. 161-195, 1998.

[Zhuang, X. et al., 1986] Zhuang, X., Haralick, R. and Huang, T.S., "Two-view motion analysis: A unified algorithm", *Journal of the Optical Society of America A: Optics, Image Science, and Vision*, vol. 3, no. 9, pp. 1492-1500, 1986.