UNIVERSIDAD DE ALCALÁ

Escuela Politécnica

INGENIERÍA DE TELECOMUNICACIÓN

Universidad de Alcalá

Trabajo Fin de Carrera

Sistema de detección de objetos mediante cámaras. Aplicación en Espacios Inteligentes.

Amaia Santiago Pé

2007

UNIVERSIDAD DE ALCALÁ

Escuela Politécnica Superior

INGENIERÍA DE TELECOMUNICACIÓN

Trabajo Fin de Carrera

Sistema de detección de objetos mediante cámaras. Aplicación en Espacios Inteligentes.

Alumno: Amaia Santiago Pé.

Director: D. Daniel Pizarro Pérez.

Tribunal:

Presidente: Dr. D. Manuel Mazo Quintas.

Vocal 1º: Dª. M. Soledad Escudero Hernanz.

Vocal 2º: D. Daniel Pizarro Pérez.

CALIFICACIÓN: .....

FECHA: .....

Rompe las cadenas de tu pensamiento,
y romperás también las cadenas de tu cuerpo.

No creas lo que tus ojos te dicen. Sólo muestran limitaciones.
Mira con tu entendimiento, descubre lo que ya sabes
y hallarás la manera de volar.
– ¡Puedo volar! ¡Escuchen! ¡PUEDO VOLAR!

"Juan Salvador Gaviota"
Richard Bach

Agradecimientos

AGRADECIMIENTOS.

Índice general

Índice de figuras

Parte I

Resumen

Resumen

Este proyecto aborda el problema de la obtención de la posición y orientación métrica de los objetos tridimensionales en un entorno. Se propone un sistema de cámaras calibradas e interconectadas que forman un “Espacio Inteligente”.

El objetivo es realizar tareas de búsqueda y reconocimiento de objetos en imágenes. Cada objeto es detectado a partir de un modelo geométrico de puntos identificados en la imagen mediante características pseudo-invariantes a la deformación proyectiva. Mediante una posterior identificación robusta, el sistema es capaz de obtener la posición y orientación del objeto buscado.

El uso de características basadas en información local, permiten que el sistema sea invariante a cambios de iluminación, grandes oclusiones y cambios de perspectiva.

Se proponen varios algoritmos de estimación robusta que permiten realizar la detección y el proceso de correspondencia para varias repeticiones espaciales del mismo objeto. Además, el sistema es ampliable a una o varias cámaras de manera sencilla.

Para validar el enfoque propuesto se presentan ejemplos de detección de objetos en imágenes reales y se analizará las limitaciones y problemas asociados a dicho enfoque.

Palabras Clave : “Espacio Inteligente”, SIFT, Transformada de Hough, RANSAC.

Parte II

Memoria

Capítulo 1

Introducción

La **Visión Artificial** es la ciencia que estudia, utilizando computadores, los procesos de obtención, caracterización e interpretación de la información procedentes de imágenes tomadas del mundo tridimensional.

Una de las áreas de aplicación de la Visión Artificial es la de **Reconocimiento de Objetos** tridimensionales. El fin es el de determinar si una imagen contiene o no, uno o varios objetos concretos. Además, los sistemas de reconocimiento no sólo deben centrarse en determinar la presencia de objetos en la escena de análisis, sino que también deben ser capaces de obtener la localización de los mismos en el espacio.

El ser humano es capaz de reconocer objetos instantáneamente sin ningún tipo de esfuerzo. Sin embargo, dotar a las maquinas de esta capacidad es una tarea complicada. Si somos capaces de implementar algoritmos basados en visión que permitan detectar objetos, podremos conseguir maquinas más autónomas capaces de realizar acciones análogas a las que realiza el ser humano.

Por tanto, la idea es llegar a sistemas de reconocimiento y estimación de la pose que sean:

- **Genéricos**: capaces de detectar objetos arbitrarios de distintas clases, de forma que puedan aplicarse no sólo en situaciones definidas y muy concretas.
- **Invariantes**: existen multitud de factores que influyen negativamente en el proceso de reconocimiento. Algunos de estos factores son los cambios de iluminación en la escena, las rotaciones y traslaciones que puede sufrir el objeto, la escala, distorsiones de las cámaras, deformaciones proyectivas, etc.
- **Robusto**: debe ser un sistema tolerante a ruidos, a posibles oclusiones del objeto que se desea detectar, fondos complejos, etc.
- **Eficiente**: capaz de usarse en aplicaciones de tiempo real.

El reconocimiento adquiere gran importancia en multitud de campos donde se aplica la Visión Artificial:

- Sistemas de seguridad y vigilancia.
- Sector industrial. El proceso de reconocimiento de objetos tridimensionales es importante en distintas tareas que se realizan en las líneas de producción (tareas de inspección, ensamblado, manipulación de piezas, etc.).
- Motores de búsqueda que permiten reconocer objetos en archivos de imagen. Muchos de estos sistemas se basan en las descripciones que se suministran junto con las imágenes y otros identificadores escritos (como el nombre del archivo). Sin embargo, no explotan toda la capacidad de búsqueda que se tendría si también se utilizara la información que aportan las imágenes en sí mismas.
- Navegación y localización de robots móviles. Los robots interactúan con un mundo tridimensional y dinámico. Si somos capaces de dotar a los robots del sentido de la vista, podrán modelar el mundo real que los rodea, detectar obstáculos y así, serán capaces de interactuar de forma más eficiente y autónoma en ambientes complejos.
- Sistemas inteligentes de transporte y asistencia a la conducción. Por ejemplo, el proceso de reconocimiento es importante en los sistemas de asistencia a la navegación, en la detección de información vial, en la detección de otros vehículos y peatones.
- En la actualidad, se busca cada vez más conseguir una mayor interacción entre el hombre y la máquina. No sólo se buscan sistemas que sean capaces de realizar tareas concretas, sino que sean capaces de tomar decisiones y de comunicarse con el entorno que les rodea. Surge así el concepto de **"Espacio Inteligente"**. Un "Espacio Inteligente" consiste en una habitación dotada de "inteligencia" donde existe una interacción entre los usuarios y el propio entorno, con el fin de ayudarles en algún tipo de tarea. Para poder percibir y comunicarse con el entorno, el espacio necesita de sensores, como por ejemplo cámaras. Por tanto, el proceso de reconocimiento de objetos es muy importante en los "Espacios Inteligentes".

Este proyecto fin de carrera se encuentra enmarcado en este último campo de aplicación. Se pretende aportar un trabajo dedicado a la detección de objetos utilizando un sistema de una o múltiples cámaras como sistema sensorial de los "Espacios Inteligentes". Para resolver el problema de la detección, se propone utilizar un modelo de características invariantes a las deformaciones proyectivas. De este modo, la detección de objetos tridimensionales puede llevarse a cabo con independencia de la posición que ocupe con respecto a la cámara.

El modelo de características se basa en recientes avances realizados en el campo de la detección de puntos de interés en las imágenes [Lowe, 2004], que presentan un alto grado de inmunidad ante transformaciones afines. Este tipo de técnicas posibilitan el emparejamiento o "matching" de puntos, entre una imagen patrón del objeto que se desea detectar y una imagen cualquiera que contenga una o varias repeticiones de dicho objeto. Por cada punto de interés se obtiene un descriptor basado en la información local que es

suficiente para obtener el “matching”. Además, se propone el uso de varias técnicas robustas de estimación [Duda and Hart, 1971] [Fischler and Bolles, 1981] para eliminar los inevitables falsos emparejamientos.

1.1. Objetivos y definición del problema

Este proyecto aborda el problema de la obtención de la posición y orientación métrica de objetos tridimensionales que se encuentran en un entorno. Surge como solución a un problema común que existe en los “Espacios Inteligentes”.

Dentro del entorno definido por un “Espacio Inteligente” existen distintos tipos de elementos, ya sean objetos estáticos o agentes móviles sobre los que tiene control el propio entorno o simplemente usuarios autónomos. Para cumplir los objetivos para los que fue diseñado dicho “Espacio Inteligente” deberá ser capaz de detectar e interpretar todo lo que ocurre en el entorno. Por tanto, la capacidad de detección de objetos es crucial en los “Espacios Inteligentes”. A continuación se enumeran algunas situaciones donde está presente este problema de la detección de objetos:

- En todo momento, el entorno debe tener conocimiento de los agentes y los elementos estáticos que ocupan el espacio de trabajo. Además, cada agente del entorno tendrá su propia función dentro del “Espacio Inteligente”, por lo que éste debe ser capaz de distinguirlos.
- Los distintos elementos interaccionan dentro del entorno por lo que el “Espacio Inteligente” debe ser capaz de detectar su posición y orientación dentro del mismo para poder controlar los distintos agentes sobre los que tiene acceso y evitar posibles interferencias no deseadas entre todos ellos.
- Para cumplir alguno de los objetivos del “Espacio Inteligente”, es posible que se necesite esta capacidad de detección automática de objetos (por ejemplo, buscar objetos concretos en espacios orientados a la asistencia y ayuda de personas).

Para resolver el problema de la detección, se propone utilizar un conjunto de cámaras calibradas que forman una red distribuida de sensores interconectados. Este conjunto de cámaras forman parte de la *Capa de Percepción* de un “Espacio Inteligente” y se encuentran separadas una distancia que consiste en varios órdenes de magnitud la distancia focal de cada una de ellas. En esta situación, la distancia entre cada cámara y los objetos que hay en el entorno también es similar a la distancia entre centros. Este esquema es conocido en la literatura como “wide-baseline”.

Esto hace posible que, con un número reducido de cámaras se de cobertura a un amplio espacio. Además, el uso de cámaras permite al entorno tener a su disposición un flujo de información completo y de alta complejidad.

Todo proceso de detección utilizando cámaras consta de los siguientes pasos generales:

- Extracción de información en cada una dé las imágenes disponibles.
- Correspondencia de la información entre diferentes cámaras.
- Obtención de la pose del objeto mediante recuperación de la geometría.

El principal inconveniente que presentar la configuración "wide-baseline" es la dificultad para relacionar la información que aporta cada cámara. La correlación entre las distintas imagenes es baja debido a la deformación proyectiva resultado de la separación entre los centros de proyección (en ocasiones, es posible que cada cámara registre en las imagenes el mismo objeto pero desde vistas diferentes). Por tanto, es necesario métodos que permitan realizar la correspondencia ante esta distorsión proyectiva. Existen dos tendencias claras:

- Detección basada en marcas artificiales.
- Detección basada en marcas naturales.

De estas dos propuestas, en este proyecto se va a utilizar la detección por marcas naturales, pues aprovecha la información intrínseca que aporta el objeto al ser proyectado en el plano imagen. El inconveniente que presentan respecto a un sistema basado en marcas artificiales es que hay que determinar qué puntos del objeto presentan ciertas características que los convierte en candidatos idóneos para ser detectados en otras imagenes.

Existen numerosos enfoques para solucionar el problema del reconocimiento de objetos mediante el uso de marcas naturales. Uno de estos enfoques se basa en modelar los objetos que se desean detectar en una fase de aprendizaje previa. Para la detección, se buscan correspondencias entre dichos modelos y la información que se extrae de la escena a analizar. Dentro de este método de reconocimiento basado en modelos, existen a su vez distintas vertientes como las basadas en modelos geométricos y de apariencia.

En los últimos años, están tomando mucha importancia las técnicas basadas en la extracción de modelos geométricos locales a partir de puntos con características invariantes. A estos puntos se les denominan **puntos característicos o de interés**. Al ser invariantes, se podrán detectar ante diferentes poses del objeto en la imagen. A pesar de su éxito, la robustez y generalidad de estas técnicas depende de la repetitividad de las características locales en los modelos y en las imágenes a analizar y a la dificultad de realizar una correspondencia correcta de éstas.

En general, estos métodos basados en características invariantes siguen el esquema que se muestra en la figura 1.1:

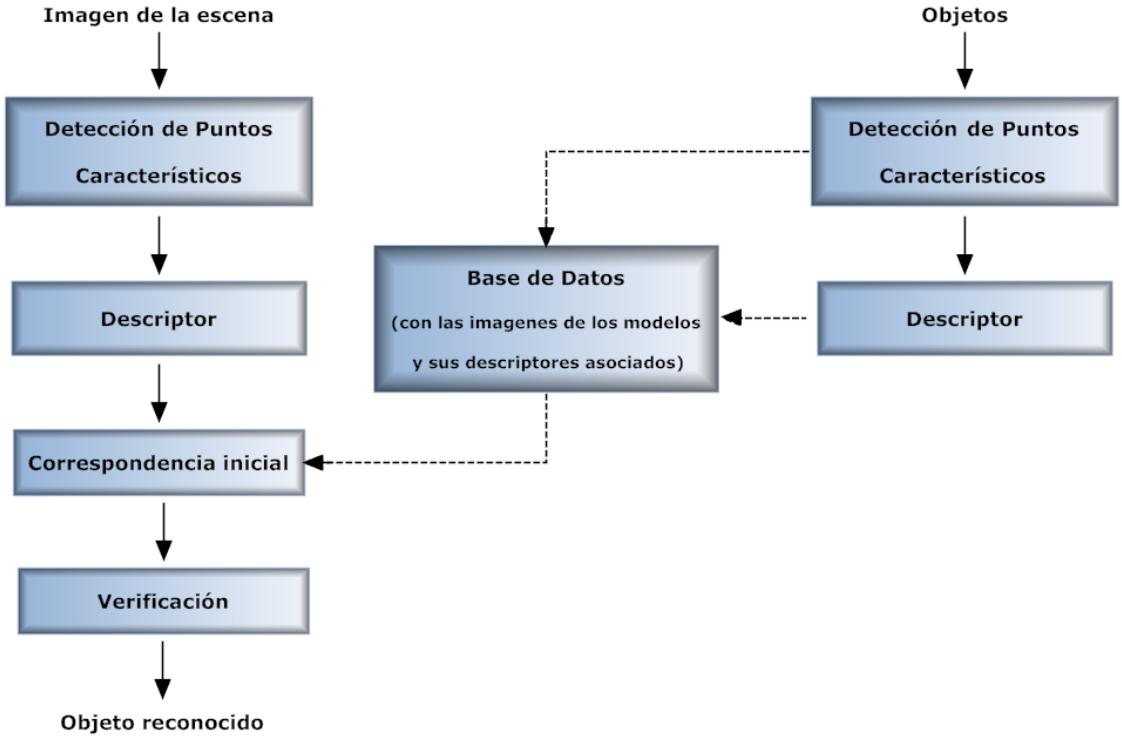


Figura 1.1: **Objetivos y definición del problema.** *Esquema del proceso seguido en los métodos de correspondencia mediante características locales e invariantes*

1. **Detección de puntos característicos** - Este primer paso consiste en la detección y localización de los puntos de interés (e.g detector de Harris, detección de extremos en espacio de escala, etc.). Es importante que estos puntos presenten una alta repetitividad entre las imágenes y los modelos generados y que a su vez, el número de puntos detectados sea elevado para poder reconocer objetos con alto grado de oclusión.

2. **Obtención del descriptor** - A través de la información que aportan las imágenes, se obtiene la apariencia local de cada punto de interés y se codifica en función de su vecindad. Debido a que dicha información suele cambiar en función de la pose del objeto y de otros factores, el objetivo es encontrar descriptores que no varíen en exceso al modificarse estos factores y que, a su vez, aporten suficiente información para poder reconocer el punto buscado del resto de puntos en la imagen (e.g. histogramas de orientación, descriptores SIFT, etc).

3. **Generación de la base de datos** - Como ya se ha comentado, estos métodos tienen una fase previa de aprendizaje, en la cual se crea una base de datos con los modelos de los objetos a detectar. Estos modelos están constituidos por todos los puntos de interés detectados en los objetos y sus correspondientes descriptores.

4. **Correspondencia o "matching" inicial** - Se compara, mediante alguna medida de similitud, cada descriptor encontrado en la imagen con todos los descriptores de la base de datos para establecer una correspondencia.

5. **Verificación** - El principal problema de estos métodos es la elevada probabilidad de generar correspondencias erróneas, denominadas **"outliers"**. Mediante algún sistema de estimación robusta, hay que intentar separar los "outliers" de las correspondencias correctas, denominadas **"inliers"**, agrupar dichos "inliers" en función del objeto detectado al que pertenezcan y obtener la pose (localización en el espacio y orientación) de los mismos.

En este proyecto, se propone utilizar uno de estos métodos basados en características invariantes. En concreto, se utiliza el método **SIFT** (Scale Invariant Feature Transform) propuesto por David Lowe [Lowe, 2004].

Por otro lado, el proceso de detección de objetos tridimensionales haciendo uso de múltiples cámaras está condicionado a la difícil tarea de relacionar la información aportada por cada una de las cámaras para poder recuperar la información tridimensional del objeto. Por tanto, para simplificar el problema, en este proyecto se va a considerar que se parte de una única cámara.

El hecho de usar una sola cámara hace imposible la tarea del posicionamiento 3D de objetos pues al proyectar un objeto en una imagen se pierde la información de profundidad. Por tanto, se debe hacer uso de información complementaria. Se va a conocer la matriz de calibración de la cámara y su posicionamiento respecto al sistema de referencia del "Espacio Inteligente". De esta forma, con la información adicional y mediante el proceso de emparejamiento entre el patrón y los puntos característicos encontrados en la imagen se puede conocer la posición y orientación real del objeto.

La memoria de proyecto se ha estructurado en nueve capítulos, incluyendo éste. El contenido de los capítulos restantes se detalla a continuación:

- **Capítulo 2: Estado del Arte.** Se describen los conceptos generales de un ”Espacio Inteligente” y el estado del arte de los distintos métodos de detección de objetos.

- **Capítulo 3: Geometría de formación de la imagen.** En este capítulo se realiza un repaso de todos los conceptos de geometría aplicados a la visión por computador que van a servir de base para la realización de este proyecto.

- **Capítulo 4: Detección de puntos de interés en la imagen.** Se detalla el método concreto que se va a utilizar para la extracción de puntos característicos de una imagen cualquiera y el proceso de emparejamiento entre los puntos característicos del patrón y los de la imagen en la que se desea detectar el objeto.

- **Capítulo 5: Estimación mediante algoritmos robustos.** En este capítulo se describe de forma general tres métodos de estimación robusta propuestos para determinar los falsos emparejamientos.

- **Capítulo 6: Detección de objetos planares.** Se detalla el sistema completo de reconocimiento de objetos que se ha implementado. En concreto, este sistema de detección se centra en el reconocimiento de objetos planares.

- **Capítulo 7: Detección de objetos tridimensionales.** En el capítulo anterior, sólo se resolvía el problema de la detección para objetos planares. Ahora se intenta dar una solución al caso general en el que el objeto puede ser tridimensional.

- **Capítulo 8: Resultados y simulaciones** Se analizarán los resultados obtenidos al aplicar el sistema de detección a una imagen y se hará un estudio detallado de los errores y factores que influyen en la detección.

- **Capítulo 9: Conclusiones** Para terminar, se exponen las conclusiones obtenidas tras analizar los resultados del capítulo anterior y se propone una serie de ideas o alternativas para futuras mejoras.

Para terminar, cabe destacar que todos los algoritmos implementados para la realización de este proyecto se han programado utilizando Matlab 7.0.

Capítulo 2

Estado del Arte

2.1. Espacio Inteligente

Los **"Espacios Inteligentes"** [Hashimoto and Lee, 2002] son habitaciones o áreas capaces de percibir y entender lo que ocurre en ellas. Están equipados con diferentes tipos de sensores (cámaras, micrófonos, etc), actuadores (displays, speakers, proyectores, agentes móviles etc) y dispositivos de comunicación. A través de los sensores, los "Espacios Inteligentes" captan lo que ocurre en ellos, analizan las diferentes situaciones, reaccionan ante ellas y se comunican con los usuarios a través de los actuadores. Son capaces de observar a las personas, analizar sus acciones y gestos y actuar de acuerdo a ellos.

Además, los "Espacios Inteligentes" no sólo se centran en las personas. Pueden dar soporte a robots, de forma que estos adquieren mayor inteligencia a través de la interacción con el espacio. Por tanto, los robots pueden convertirse en actuadores del "Espacio Inteligente" y proporcionar servicios a los usuarios.

De esta forma, conseguimos transformar espacios pasivos en áreas sensitivas capaces de actuar y reaccionar ante ciertas situaciones. Estas cualidades hacen que los "Espacios Inteligentes" tengan aplicación en multitud de escenarios como hospitales, oficinas, fabricas, hogares, etc.

2.1.1. Historia del Espacio Inteligente

El idea de dotar de inteligencia a los entornos que nos rodean ha sido propuesta por diversos autores y ha ido separándose en distintas implementaciones a medida que evolucionaba. Uno de los campos donde se ha aplicado este concepto es en la asistencia a personas y en la localización y navegación de robots.

El enfoque de "Espacio Inteligente" proviene de áreas de investigación relacionadas con el interfaz hombre-máquina (HMI). El objetivo es definir espacios donde exista una interacción no intrusiva entre los usuarios y el propio entorno, con la finalidad de ayudar

a los usuarios en algún tipo de tarea. La interacción con el entorno se basa en la comunicación que mantienen los usuarios con él a través de un lenguaje natural entendido por ambos.

El concepto de "Espacio Inteligente" proviene originalmente de la idea de **"Computación Ubicua"**, propuesta por Weiser [Weiser, 1999][Weiser, 1993b]. Este autor hace una clasificación en función de cuatro elementos básicos:

- **Ubicuidad**: Múltiples sistemas embebidos con total interconexión entre ellos.
- **Conocimiento**: Habilidad del sistema para localizar y reconocer lo que ocurre en su entorno y su comportamiento.
- **Inteligencia**: Capacidad de adaptarse al mundo que percibe.
- **Interacción Natural**: Capacidad de comunicación entre el entorno y los usuarios.

Uno de los primeros sistemas ubicuos implementados con esta filosofía se propuso en el proyecto Xerox PARC de Computación Ubicua (UbiComp) [Weiser, 1993a], desarrollado a finales de los 80. La red de sensores utilizada no estaba basada en información visual debido al coste prohibitivo para aquella época. Hoy en día, varios grupos de investigación desarrollan y amplían el concepto de "Espacio Inteligente" y de "Computación Ubicua". Alguno de los más importantes se indican a continuación:

- *Intelligent Room* [Coen, 1998]. Desarrollado por el grupo de investigación del *Artificial Intelligent Laboratory* en el MIT (Masachussets Institute of Technology). Es uno de los proyectos más evolucionados en la actualidad. Se trata de una habitación dotada de cámaras, micrófonos y otros sensores que realizan una actividad de interpretación con el objetivo de averiguar las intenciones de los usuarios. La comunicación entre estos y la habitación es mediante voz, gestos y contexto.
- *Smart Room* [Pentland, 1996]. Desarrollado en el *Media Lab* del MIT. Mediante el uso de cámaras, micrófonos y sensores se pretende analizar el comportamiento humano dentro del entorno. Se localiza al usuario, identificándolo mediante su voz y apariencia y se realiza un reconocimiento de gestos. La línea de investigación actual es la capacidad del entorno de aprender a través del usuario que, a modo de profesor, enseña nombres y apariencia de diferentes objetos.
- *Easy Living* [Shafer et al.,]. Se trata de un proyecto de investigación de la empresa Microsoft con el objetivo de desarrollar "entornos activos" que ayuden a los usuarios en tareas cotidianas. Al igual que en los otros proyectos comentados, se intenta establecer una comunicación entre el entorno y el usuario mediante un lenguaje natural. Se hace uso de la visión artificial para realizar identificación de personas e interpretación de comportamientos dentro del espacio inteligente.

- Un trabajo más evolucionado fue propuesto en la universidad de Tokyo por Lee y Hashimoto [Hashimoto et al., 2003][Lee et al., 2005]. En este caso se dispone de un "Espacio Inteligente" completo en el que se mueven robots y usuarios. Cada dispositivo de visión es tratado como un dispositivo inteligente distribuido conectado a una red (denominado DIND). Cada DIND tiene capacidad de procesamiento por sí solo, por lo que se posee flexibilidad a la hora de incluir nuevos DIND. Puesto que se cuenta con dispositivos calibrados, la localización se realiza en espacio métrico. Los robots están dotados de balizamiento pasivo mediante un conjunto de bandas de colores fácilmente detectables por cada DIND. En este espacio de trabajo (Ispace), se realizan experimentos de interacción entre humanos y robots y navegación con detección de obstáculos.

- En el Departamento de Electrónica de la Universidad de Alcalá se ha formado un grupo de investigación sobre "Espacios Inteligentes" [Villadangos et al., 2005] [Basagoitia et al., 2004] [Pizarro et al., 2005] en el que mediante un conjunto de sensores se pretende realizar posicionamiento de robots móviles. Las alternativas incluyen visión artificial, ultrasonidos e infrarrojos. Este proyecto fin de carrera se enmarca en dicho grupo.

2.1.2. Estructura del Espacio Inteligente

Para entender la estructura de los "Espacios Inteligentes" es necesario definir una serie de conceptos:

- *Entorno:* se considera una entidad fija en el espacio. En dicho entorno se producirán una serie de eventos que deberán ser observados, analizados y comprendidos por el entorno. A su vez, el entorno se divide en varias capas funcionales:

  - *Capa Pasiva de Percepción:* Esta compuesta por un número de sensores que permiten captar información del espacio de trabajo. Esta información será utilizada por el entorno. Los sensores que componen esta capa pueden ser de distintos tipos (cámaras, sensores láser y de ultrasonidos,...), estar fijos en el espacio o incluso ir unidos a la *Capa de Interacción*.
  - *Capa de Interacción:* Esta capa se compone de un conjunto de *Agentes Controlables* que pueden interactuar físicamente con el mundo. Esta capa tiene doble funcionalidad: por un lado, se utiliza para cumplir con los objetivos del entorno para el que fue diseñado y por otro lado, puede utilizarse como parte de la *Capa de Percepción Activa*. Debido a que estos agentes interaccionan con el mundo, tendrán que estar dotados a su vez de un sistemas sensorial, por lo que la información que captan puede utilizarse para complementar la información captada por los propios sensores del entorno.

- *Objetivos del entorno:* En general, existen dos fuentes encargadas de fijar los objetivos del entorno. Por un lado están los *Agentes Autónomos* (por ejemplo, una persona) que se encuentran en el espacio y con los que el entorno tiene que establecer una comunicación a través de las *Capas de Percepción*. Por otro lado, puede haber agentes que se encuentren conectados a la *Capa de Comunicaciones*.
- *Capa de Comunicaciones:* Los *Agentes Controlables* y eventualmente los *Agentes Autónomos* se encuentran conectados al entorno a través de una red de comunicaciones que representa el sistema nervioso central del "Espacio Inteligente". Esta red conecta la *Capas de Percepción* y la *Capa de Interacción* con la *Capa de Inteligencia*.
- *Capa de Inteligencia*: Se necesita un sistema de procesado que sea capaz de gestionar toda la información captada por los sensores, que genere bases de datos con toda esta información y que permita el intercambio de información con los distintos agentes del entorno. No se compone necesariamente de una gran unidad de proceso central y se estructura en una jerarquía de capas.

La interacción del entorno con el resto del mundo se realiza mediante la interpretación realizada en la *Capa de Percepción* y los *Agentes Controlables*.

- *Mundo de Percepción*: La Capa de Percepción y la Capa de Acción están relacionadas con los eventos que tienen lugar en el entorno. Dichos eventos pueden catalogarse como:

  - *Agentes Autónomo*: Desarrolla un tipo de actividad en el entorno sin que éste tenga control sobre él. Puede generar objetivos o simplemente puede ser un obstáculo no estático que los *Agentes Controlables* deben evitar.
  - *Objetos Estáticos*: Los objetos estáticos son parte del conocimiento que el entorno debe de adquirir con el fin de desarrollar satisfactoriamente su capacidad de interacción y el cumplimiento de los objetivos.
  - *Agentes Controlables*: Forman parte de la *Capa de Acción*, y el entorno sabe acerca de su existencia gracias a la *Capa de Comunicaciones*. Están presentes en el mundo de percepción del entorno, por lo que poseen una firma o apariencia desde el punto de vista de la *Capa de Percepción*.

En resumen, tenemos una serie de elementos en el entorno de distinta naturaleza. Independientemente del tipo de elemento y de su funcionalidad, el entorno debe ser capaz de detectarlos y posicionarlos espacialmente utilizando la información que aportan los distintos sensores del "Espacio Inteligente". En este proyecto se intenta dar una posible solución al problema de detección y localización de objetos en "Espacios Inteligentes" utilizando la visión artificial.

En la figura 2.1, se muestra un esquema aclaratorio de la estructura general de los “Espacios Inteligentes”.

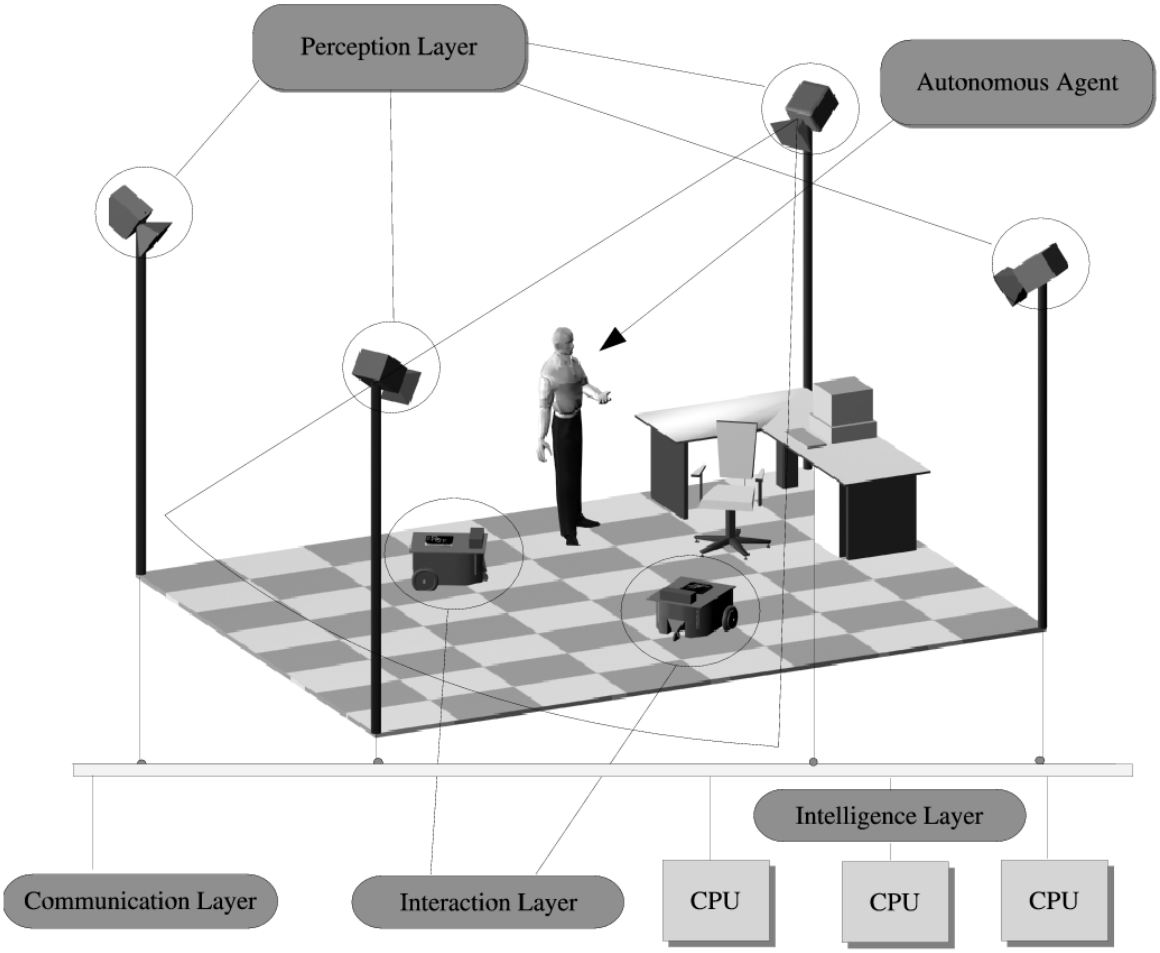


Figura 2.1: **Estado del Arte.** *Estructura de un “Espacio Inteligente”*

2.2. Detección de objetos tridimensionales mediante cámaras

Este proyecto se centra en aportar una posible solución al problema de la detección de objetos tridimensionales utilizando un sistema de una o múltiples cámaras. El objetivo final es conseguir detectar objetos que se encuentran repartidos por todo el espacio de trabajo y ser capaces de posicionarlos y orientarlos a partir de una imagen de la escena.

El primer paso es encontrar algún método que permita detectar la proyección del objeto en la imagen, a partir de la información que se puede extraer de la misma. Existen numerosas técnicas que permiten extraer esta información necesaria para la detección. En este apartado se enumeran y clasifican alguna de estas técnicas.

2.2.1. Detección de objetos basado en marcas artificiales

La utilización de marcas artificiales es una técnica clásica en visión artificial y su uso está bastante extendido en campos como el de la calibración y la fotogrametría. De entre los objetivos que se persiguen mediante la utilización de marcas artificiales, se pueden destacar los siguientes:

- Las proyecciones de cada marca artificial en el plano imagen debe ser unívoca, de forma que no haya confusión a la hora de determinar, a partir de la proyección en la imagen, a qué marca pertenece.
- El proceso de extracción de una marca artificial en las imágenes debe ser sencillo y robusto.
- El diseño de las marca artificiales debería garantizar que en el proceso de detección y localización las soluciones obtenidas tienen suficiente precisión.

Por lo general, la posición tridimensional de estas marcas o la distancia relativa entre ellas es conocida. La objetivo del uso de marcas artificiales es el de asegurar un proceso robusto de correspondencia y detección en el plano imagen. Dentro de este método de detección, se puede diferenciar dos tendencias en función del tipo de marcas que utilizan:

- **Marcas artificiales puntuales** - Se basan en el uso de marcas que se corresponden con regiones circulares. Una propiedad importante de estas marcas es que presentan una apariencia relativamente invariante ante deformaciones proyectivas y la obtención de su centroide proporciona una localización de la marca muy estable.
- **Marcas artificiales planares** - En este caso el método de detección no se basa en un proceso de correspondencia puntual, sino que las marcas artificiales consisten en planos de los cuales se calcula directamente su pose.

2.2.2. Detección de objetos mediante marcas naturales

El uso de marcas artificiales requiere un cierto grado de intervención a priori en el objeto que se desea detectar. Además, se necesita realizar ciertos procesos de calibración previos y no hace uso de la propia información que puede aportar la imagen del objeto de manera natural. El reto en los sistemas de visión artificial en la actualidad es el de realizar el seguimiento de objetos mediante información visual debida a la propia geometría de la escena. Esto requiere en general un tiempo de proceso mayor.

2.2.2.1. Métodos basados en modelo previo

En este tipo de métodos se parte de un modelo previo de cada uno de los objetos que se desean detectar. A partir de este modelo, se buscan aquellos métodos que sean capaces de encontrar puntos en la imagen y asociarlos a las primitivas (puntos, líneas, volúmenes) que definen el modelo, es decir, generar la información que permitirá diferenciar unos puntos de otros en función de las características que presenten dichos puntos. Una vez asociados, se deberá ser capaz de calcular la correspondencia entre el modelo y la imagen a analizar. A partir de este emparejamiento de puntos, el sistema debería ser capaz de obtener la posición y orientación del objeto.

Los trabajos de detección basados en este método es muy extensa. Se puede realizar una posible división de todos estos trabajos en función del tipo de características que se buscan en la imagen y que se utilizan para generar los modelos A continuación se enumeran alguno de ellos:

- **Métodos basados en detección de bordes.**

  Los primeros trabajos de detección de objetos estaban basados en la detección de bordes debido a su eficiencia computacional y a su fácil implementación. Además, estos métodos son inmunes a los cambios de iluminación.

- **Métodos basados en ajuste de plantilla.**

  Estos métodos se basan en el ajuste de una plantilla bidimensional genérica cuando ésta se ve sometida a un conjunto de deformaciones afines. En general, el objetivo de este tipo de algoritmos consiste en encontrar, para una determinada imagen, los parámetros que definen la transformación que habría que aplicarle a la plantilla para obtener el mismo resultado que en la imagen. Una vez obtenida la plantilla es posible recuperar su pose mediante la homografía que define el plano que la forma y los parámetros intrínsecos de la cámara.

  Para realizar la detección, no se utiliza características locales del objeto sino que se toma toda la imagen correspondiente a la plantilla como fuente de información para la detección. En general, este tipo de métodos de detección presentan varias limitaciones: es poco robusta ante cambios de iluminación y posibles oclusiones en la plantilla.

- **Métodos basados en extracción de puntos.**

  Este tipo de métodos ya no se basa en marcas globales (como los contornos o las plantillas) sino que utiliza marcas naturales localizadas. Se basan en la detección de puntos individuales, lo que permite que dichos algoritmos sean inmunes a cambios de iluminación, oclusiones o errores de correspondencia. Normalmente, para obtener los puntos, este tipo de enfoque se basa en la información contenida en el plano imagen más que en la apariencia real del objeto.

  Los métodos de detección basados en extracción de puntos deben resolver fundamentalmente dos problemas:

  - **Identificación de Puntos de Interés**

    En general, para identificar un punto se utiliza la información aportada por la textura de una pequeña región de la imagen situada en torno al punto. No todos los puntos proyectados del objeto son susceptibles a ser detectados en otras imágenes, pues la textura que rodea a un punto puede sufrir deformaciones proyectivas, cambios de iluminación y ruido.

    Por tanto, el proceso de identificación consiste en detectar los puntos proyectados del objeto que presenten ciertas características que los convierte en candidatos idóneos para ser detectados de nuevo en otras imágenes. Estos puntos se denominan **Puntos de Interés**.

  - **Correspondencia entre imagenes**

    Para poder detectar el objeto y obtener la transformación que ha sufrido, es necesario un método de correspondencia entre puntos de interés extraídos de una imagen y los puntos de interés del objeto. Normalmente el método de correspondencia tiene asociado algún algoritmo de estimación robusta que permita descartar falsas correspondencias.

- **Métodos basados en descriptores invariantes**

  El concepto de estos métodos es el mismo que para el caso anterior. La diferencia es que cada punto de interés tendrá asociado un descriptor local. Dicho descriptor busca extraer información de un área localizada de la imagen (como puede ser la textura alrededor del punto de interés) y define un método de codificación que sea invariante a cambios de iluminación y transformaciones afines. Los descriptores deben ser definidos de tal modo que a su vez permitan separar de manera apreciable dos áreas que no pertenezcan al mismo punto.

En este proyecto, se propone utilizar un método basado en descriptores invariantes para resolver el problema de detección de objetos. En concreto, se va a utilizar los **descriptores SIFT (Scale Invariant Features Transform)** propuestos originalmente por Lowe [Lowe, 1999]. Las marcas que detecta el método SIFT son invariantes ante cambios de escala, rotación y parcialmente inmunes a cambios de iluminación y deformación proyectiva. Poseen propiedades de localización buenas y ofrecen descriptores que permiten ser distinguidos en imágenes reales. Este método es la solución propuesta con más éxito en entornos reales hasta la fecha.

Capítulo 3

Geometría de formación de la imagen

En este capítulo se va a realizar un breve repaso a ciertos conceptos fundamentales de la visión por computador que han servido de base para la realización de este proyecto y que irán apareciendo a lo largo del texto:

- **Modelado de las cámaras** - Se va a explicar el proceso de formación de una imagen en una cámara y como éste puede modelarse matemáticamente mediante una proyección en un plano del espacio 3D. Dicho proceso de correspondencia entre los puntos tridimensionales y los bidimensionales se representa mediante una matriz P.
- **Calibración de la cámara** - Se va a describir brevemente los parámetros de los que depende la matriz P de la cámara y el proceso de obtención de dicha matriz.
- **Transformaciones proyectivas 2D** - Se va a realizar una descripción de algunas de las transformaciones geométricas que sufre un plano al proyectarse en una cámara y la manera matemática de estimar dichas transformaciones.
- **Visión estéreo** - Al proyectar un punto en una imagen se pierde la información de profundidad pues las imágenes son una representación bidimensional del mundo 3D. La utilización de dos cámaras permite extraer la información tridimensional del entorno. Esto se denomina visión estereoscópica.

De la misma forma, este capítulo tiene como objetivo familiarizarse con la notación que se va a utilizar en el resto de capítulos.

3.1. Modelado de la cámara

La formación de las imágenes consiste en una representación bidimensional del mundo 3D, perdiéndose la información de profundidad.

La óptica geométrica clásica se basa en los modelos de lentes gruesas y finas para modelar el proceso de formación de las imágenes. Sin embargo, podemos simplificar este proceso suponiendo que todos los rayos que llegan a la cámara atraviesan un único punto y se proyectan en la película. Este modelo se conoce como **modelo pinhole**.

Debido a que las lentes no tienen un comportamiento ideal, habrá que añadir al modelo unos parámetros de distorsión que permitan corregirlo y aproximarlo al comportamiento real de la cámara.

3.1.1. Estructura del modelo pinhole

El **modelo pinhole** [Hartley and Zisserman, 2004] permite modelar el proceso de formación de las imágenes mediante una **proyección central**, en la cual, de cada punto del espacio tridimensional parte un rayo de luz que pasa por un punto fijo del espacio e interseca en un plano dando lugar a la imagen.

En la figura 3.1 se muestra un esquema del modelo pinhole.

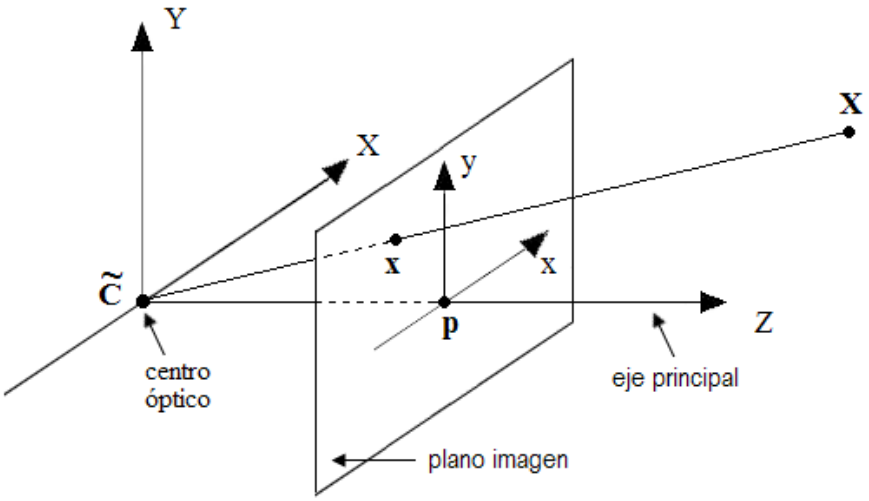


Figura 3.1: **Modelado de la cámara.** *El punto* C *es el centro óptico y* p *es el punto principal. El centro óptico es el centro de la cámara y es donde se fija el origen de coordenadas de la cámara.*

Se va a definir una serie de términos a los que se hará referencia a lo largo de este apartado:

- **Centro óptico** o **de proyección central** - Es el punto fijo del espacio por donde pasan todos los rayos de luz. Se corresponde con el centro de la cámara y es donde se fija el sistema de referencia de la cámara.
- **Plano imagen** o **plano focal** - De cada punto del espacio parte un rayo de luz que pasa por el centro de proyección e interseca con este plano formando la imagen. Como se puede ver en la figura, el plano focal se ha situado delante del centro óptico. Si éste estuviese detrás, las imágenes estarían invertidas.
- **Distancia focal** - Se define como la distancia entre el centro de proyección y el plano imagen.
- **Eje** o **rayo principal** - Es la línea que pasa por el centro de proyección y es perpendicular al plano imagen.
- **Punto principal** - Es el punto de intersección del eje principal con el plano imagen. Coincide con el centro de la imagen.
- **Plano principal** - Es el plano paralelo al plano imagen y que contiene al centro de proyección. Además, este plano está formado por todos los puntos cuyas proyecciones se corresponden con puntos en el infinito en la imagen.

3.1.2. Matriz de proyección

El modelo pinhole fija un sistema de coordenadas proyectivas en el centro óptico. El eje Z de este sistema coincide con el eje principal de la cámara. A partir de ahora nos referiremos a este sistema de coordenadas como sistema de referencia de la cámara. Además, el plano imagen se fija en el plano $Z = f$.

A partir de ahora, la notación que se va a seguir es la siguiente:

- $\mathbf{X_c} = (X_c, Y_c, Z_c)^T$ [1] $\rightarrow$ Coordenadas no homogéneas de cualquier punto del espacio tridimensional respecto al sistema de referencia de la cámara.
- $\mathbf{X_w} = (X_w, Y_w, Z_w)^T \rightarrow$ Coordenadas no homogéneas del mismo punto del espacio tridimensional respecto a un marco de referencia ligado al modelo del objeto. Este nuevo sistema de referencia es distinto al de la cámara.
- $\mathbf{x} = (x, y, z)^T \rightarrow$ Coordenadas del punto proyectado en el plano imagen.
- $f \rightarrow$ Distancia focal

[1] Se va a usar símbolos en negrita, como por ejemplo $\mathbf{X}$, para representar vectores columna. Por tanto, su traspuesta, $\mathbf{X}^T$, será un vector fila. Por convenio con la notación anterior, las coordenadas de un punto cualquiera se van a representar por un vector columna $\mathbf{X} = (X, Y, Z)^T$.

Mediante relación de triángulos, observando la figura 3.2, podemos determinar el proceso de correspondencia entre un punto cualquiera del espacio y su proyección en el plano imagen:

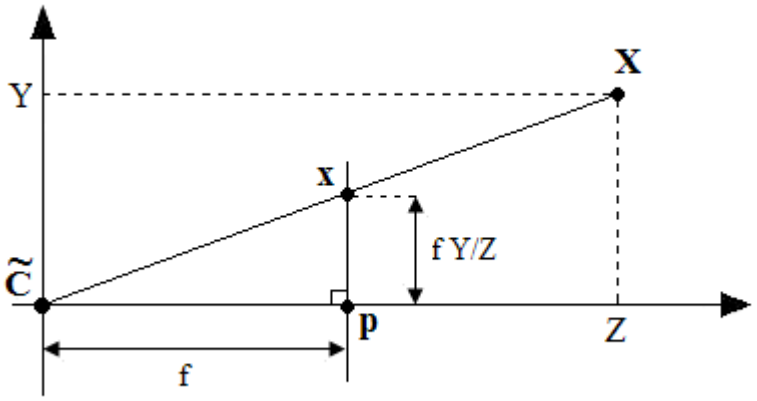


Figura 3.2: **Modelado de la cámara.** *Detalle del modelo pinhole.*

$$(X_c, Y_c, Z_c)^T \quad \longmapsto \quad (x, y)^T = (fX_c/Z, fY_c/Z)^T \tag{3.1}$$

Utilizando coordenadas homogéneas, podemos expresar la relación anterior de forma matricial:

$$\begin{pmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{pmatrix} \quad \longmapsto \quad \begin{pmatrix} fX_c \\ fY_c \\ Z_c \end{pmatrix} = \begin{bmatrix} f & & & 0 \\ & f & & 0 \\ & & 1 & 0 \end{bmatrix} \begin{pmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{pmatrix} \tag{3.2}$$

La expresión anterior, abreviada, queda de la siguiente forma:

$$\mathrm{P} = diag(f, f, 1)\,[I \mid \mathbf{0}] \qquad \Longrightarrow \qquad \mathbf{x} = \mathrm{P}\mathbf{X_c} \tag{3.3}$$

donde:

- $diag(f, f, 1)$ $\rightarrow$ es una matriz diagonal.
- $[I \mid \mathbf{0}]$ $\rightarrow$ representa una matriz identidad de 3x3 concatenada con un vector columna nulo de dimensión 3.

- $\mathbf{X_c}$ → es el vector columna, de dimensión 4, que representa las coordenadas homogéneas del punto del espacio respecto al sistema de referencia de la cámara.
- $\mathbf{x}$ → es el vector columna, de dimensión 3, que representan las coordenadas homogéneas del punto de la imagen.

A la matriz P de 3x4 se la denomina **matriz homogénea de proyección de la cámara**.

3.1.3. Cambio del sistema de referencia

Hasta ahora, hemos supuesto que el origen del sistema de coordenadas se encuentra en el centro de proyección de la cámara. Sin embargo, en la mayoría de las aplicaciones prácticas, interesa tomar otro sistema de referencia.

Como la matriz P depende del sistema tomado, tenemos que determinar cómo varía ésta en función del marco de referencia que se tome.

3.1.3.1. Cambio del sistema de referencia del objeto al de la cámara

Es normal que los puntos del espacio estén referidos respecto a otro sistema de referencia distinto del de la cámara, tal y como se muestra en la figura 3.3.

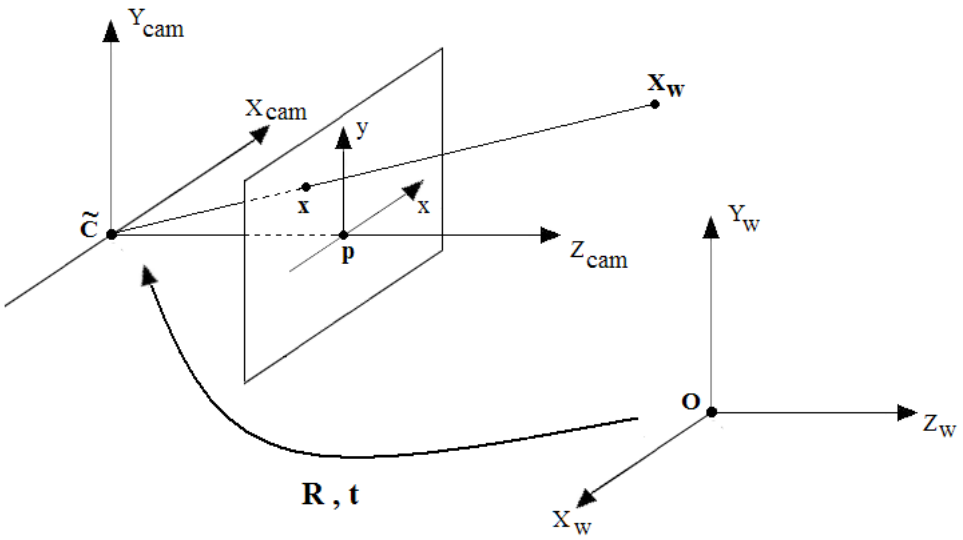


Figura 3.3: **Modelado de la cámara.** *Transformación entre el sistema de referencia del espacio y el de la cámara.*

Ambos sistemas están relacionados a través de una rotación y una traslación de la siguiente manera:

$$\mathbf{X_c} = R(\mathbf{X_w} - \widetilde{\mathbf{C}}) \tag{3.4}$$

donde:

- $\mathbf{X_w} \rightarrow$ es el vector columna de 3 dimensiones que representa las coordenadas no homogéneas del punto respecto al sistema de referencia del espacio.
- $\mathbf{X_c} \rightarrow$ representa las coordenadas del mismo punto respecto al sistema de referencia de la cámara.
- $\widetilde{\mathbf{C}} \rightarrow$ es el centro del sistema de referencia de la cámara expresado respecto al sistema de referencia del espacio.
- $R \rightarrow$ es la matriz de rotación de 3x3 que representa la orientación del sistema de referencia de la cámara.

Si expresamos la ecuación 3.4 en coordenadas homogéneas obtenemos la siguiente expresión:

$$\mathbf{X_c} = \begin{bmatrix} R & -R\widetilde{\mathbf{C}} \\ 0 & 1 \end{bmatrix} \begin{pmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{pmatrix} = \begin{bmatrix} R & -R\widetilde{\mathbf{C}} \\ 0 & 1 \end{bmatrix} \mathbf{X_w} \tag{3.5}$$

En ocasiones no se conoce de forma explicita el punto de proyección central de la cámara, y la transformación se expresa en función de una matriz de rotación y una de traslación:

$$\mathbf{X_c} = R\mathbf{X_w} + \mathbf{t} \qquad \Longrightarrow \qquad \mathbf{X_c} = [R|\mathbf{t}]\mathbf{X_w} \tag{3.6}$$

siendo:

$$\mathbf{t} = -R\widetilde{\mathbf{C}}$$

3.1.3.2. Cambio de coordenadas en el plano imagen

En la expresión 3.2 se toma el punto principal como origen de coordenadas del plano imagen, sobre el cual se referencia los puntos proyectados. Sin embargo, tal y como se muestra en la figura 3.4, el origen de coordenadas de la imagen no suele coincidir con el punto principal (se suele tomar como origen alguna de las esquinas de la imagen), por lo que tenemos que introducir un offset:

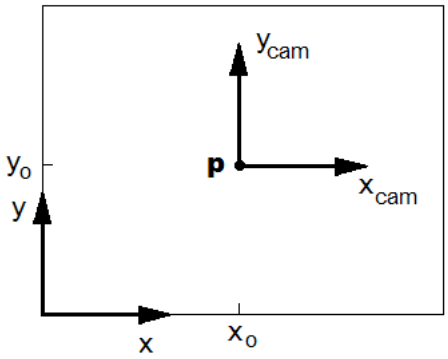


Figura 3.4: **Modelado de la cámara.** *Sistemas de coordenadas de la cámara* (x_{cam}, y_{cam}) *y de la imagen* (x, y). *El punto principal es* $\mathbf{p}$*, cuyas coordenadas imagen son* (x_o, y_o).

$$(X_c, Y_c, Z_c)^T \quad \longmapsto \quad (fX_c/Z + p_x, fY_c/Z + p_y)^T \tag{3.7}$$

donde $(p_x, p_y)^T$ son las coordenadas del punto principal referidas al sistema de referen-cia de la imagen.

De esta forma, la relación entre un punto del espacio y su proyección es la siguiente:

$$\begin{pmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{pmatrix} \quad \longmapsto \quad \begin{pmatrix} fX_c + Z_c p_x \\ fY_c + Z_c p_y \\ Z_c \end{pmatrix} = \begin{bmatrix} f & & p_x & 0 \\ & f & p_y & 0 \\ & & 1 & 0 \end{bmatrix} \begin{pmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{pmatrix} \tag{3.8}$$

Si definimos la **matriz de calibración de la cámara**, K, como:

$$K = \begin{bmatrix} f & & p_x \\ & f & p_y \\ & & 1 \end{bmatrix} \tag{3.9}$$

podemos expresar de forma abreviada la ecuación 3.8 de la siguiente forma:

$$\mathbf{x} = K[I|\mathbf{0}]\mathbf{X_c} \tag{3.10}$$

Hasta ahora, hemos supuesto que los ejes del sistema de referencia de la imagen tenían la misma escala. Sin embargo, existen sensores de adquisición en los que los píxeles no son cuadrados (como ocurre con muchas cámaras CCD), de forma que el factor de escala de cada eje es diferente.

Además, en ocasiones, interesa expresar las coordenadas de la imagen en unidades distintas a las de medida, por ejemplo, en píxeles. De esta forma, tenemos que multiplicar la matriz de calibración por un factor de escala a la vez que realizamos la transformación de unas unidades a otras:

$$K = \begin{bmatrix} m_x & & \\ & m_y & \\ & & 1 \end{bmatrix} \begin{bmatrix} f & & p_x \\ & f & p_y \\ & & 1 \end{bmatrix} \quad \Longrightarrow \quad K = \begin{bmatrix} \alpha_x & & x_o \\ & \alpha_y & y_o \\ & & 1 \end{bmatrix} \tag{3.11}$$

donde:

- $m_x \rightarrow$ es el número de píxeles por unidad de distancia, en la dirección del eje x.
- $m_y \rightarrow$ representa el número de píxeles por unidad de distancia, en la dirección del eje y.
- $\alpha_x = f m_x \rightarrow$ representa la distancia focal de la cámara en píxeles, en la dirección del eje x.
- $\alpha_y = f m_y \rightarrow$ es la distancia focal, en píxeles, en la dirección y.
- $\widetilde{\mathbf{x}}_\mathbf{o} = (x_o, y_o) \rightarrow$ es el punto principal, expresado en píxeles, cuyas coordenadas son $x_o = m_x p_x$ e $y_o = m_y p_y$.

Por último, cabe la posibilidad de que los ejes no sean perpendiculares. En este caso, tendremos que añadir un factor que refleje la deformación en función del ángulo que forman (“skew”).

Finalmente, la matriz de calibración toma la siguiente forma:

$$K = \begin{bmatrix} \alpha_x & s & x_o \\ & \alpha_y & y_o \\ & & 1 \end{bmatrix} \tag{3.12}$$

donde:

- s es el parámetro de “no perpendicularidad” o “skew". En la mayoría de las cámaras, los ejes serán perpendiculares y por tanto $s = 0$.

3.1.4. Expresión general de la matriz de proyección

El objetivo que se persigue es obtener la expresión matemática que permita relacionar un punto del espacio, cuyas coordenadas estén enmarcadas en cualquier sistema de referencia (no necesariamente el sistema de referencia de la cámara) con su proyección en el plano focal referidas al sistema de referencia de la imagen.

Uniendo las expresiones 3.6 y 3.10, se obtiene la expresión general:

$$\left.\begin{array}{l} \mathbf{X_c} = [R|\mathbf{t}]\mathbf{X_w} \\ \\ \mathbf{x} = K[I|\mathbf{0}]\mathbf{X_c} \end{array}\right\} \quad \Longrightarrow \quad \mathbf{x} = K[R|\mathbf{t}]\mathbf{X_w} \tag{3.13}$$

Por tanto, la matriz de proyección de la cámara, queda de la siguiente manera:

$$P = K[R|\mathbf{t}] \quad \Longrightarrow \quad P = \begin{bmatrix} \alpha_x & s & x_o \\ & \alpha_y & y_o \\ & & 1 \end{bmatrix} \begin{bmatrix} R_{11} & R_{12} & R_{13} & t_x \\ R_{21} & R_{22} & R_{23} & t_y \\ R_{31} & R_{32} & R_{33} & t_z \end{bmatrix} \tag{3.14}$$

Una cámara con una matriz de proyección de esta forma se denomina **cámara de proyección finita**. Esta matriz tiene dimensiones 3x4. Por tanto, tiene 11 grados de libertad, a falta de definir un factor de escala. Además, tal y como se muestra en la expresión anterior, podemos expresar dicha matriz en función de la matriz de calibración de la cámara y una matriz de rotación y traslación:

- La matriz de calibración está formada por 5 parámetros, $(\alpha_x,\ \alpha_y,\ x_o,\ y_o,\ s)$. Estos parámetros son propios del sistema de adquisición de la cámara y se denominan **parámetros intrínsecos de la cámara**.

- La otra matriz tiene 12 parámetros, 9 de ellos provienen de la matriz rotación ($R_{11} \ldots R_{33}$) y los otros 3 de la matriz de traslación ($T_{x,y,z}$). Estos 12 parámetros están relacionados con la orientación del sistema de referencia de la cámara respecto al del espacio y por tanto, son externos a la cámara (dependen del marco de referencia definido). Son los **parámetros extrinsecos de la cámara**.

3.1.5. Calibración de la cámara

La calibración de las cámaras es un proceso de estimación de los parámetros intrínsecos y extrínsecos de la cámara. Este proceso se puede dividir en:

- Estimación de la matriz P.
- Estimación de los parámetros intrínsecos y extrínsecos a partir de la matriz P obtenida. En ciertas aplicaciones sólo interesa conocer la matriz de proyección, por lo que este paso no es necesario.

El proceso de obtención de la matriz de calibración es el siguiente:

1. Se conocen una serie de puntos y sus correspondientes proyecciones en el plano imagen, $\mathbf{X_w}^i \longleftrightarrow \mathbf{x}^i$. La relación de correspondencia entre ambos puntos viene dada por la matriz de proyección P:

$$\mathbf{x}^i = P\mathbf{X_w^i} \qquad \Longrightarrow \qquad \mathbf{x}^i = \begin{bmatrix} \mathbf{P}^{1T}\,\mathbf{X_w}^i \\ \mathbf{P}^{2T}\,\mathbf{X_w}^i \\ \mathbf{P}^{3T}\,\mathbf{X_w}^i \end{bmatrix} \tag{3.15}$$

   donde:

   - $\mathbf{P}^{jT} \rightarrow$ es un vector fila de dimensión 4 que se corresponde con la fija j de la matriz P.

2. Hay que tener en cuenta, que la ecuación anterior relaciona vectores en coordenadas homogéneas. Por tanto, los vectores $\mathbf{x}^i$ y $P\mathbf{X_w}^i$ tienen la misma dirección pero difieren en magnitud por un factor no nulo. De esta forma, para poder obtener una solución de la matriz P, tenemos que realizar el producto vectorial de ambos vectores para eliminar la indeterminación de la escala.

   Siendo $\mathbf{x}^i = (x^i, y^i, z^i)$, el producto vectorial se puede expresar de forma explícita de la siguiente forma:

$$\mathbf{x}^i \times P\mathbf{X_w}^i = \mathbf{0} \qquad \Longrightarrow \qquad \begin{pmatrix} y^i\,\mathbf{P}^{3T}\,\mathbf{X_w}^i - z^i\,\mathbf{P}^{2T}\,\mathbf{X_w}^i \\ z^i\,\mathbf{P}^{1T}\,\mathbf{X_w}^i - x^i\,\mathbf{P}^{3T}\,\mathbf{X_w}^i \\ x^i\,\mathbf{P}^{2T}\,\mathbf{X_w}^i - y^i\,\mathbf{P}^{1T}\,\mathbf{X_w}^i \end{pmatrix} = \mathbf{0} \tag{3.16}$$

Puesto que $\mathbf{P}^{jT}\ {\mathbf{X}_\mathbf{w}}^{i} = {\mathbf{X}_\mathbf{w}}^{iT}\ \mathbf{P}^{j}$ para $j = 1, \ldots, 3$, de la expresión 3.16 obtenemos un sistema de tres ecuaciones donde los elementos de la matriz P son las incógnitas:

$$\begin{bmatrix} \mathbf{0}^T & -z^i\ {\mathbf{X}_\mathbf{w}}^{iT} & y^i\ {\mathbf{X}_\mathbf{w}}^{iT} \\ z^i\ {\mathbf{X}_\mathbf{w}}^{iT} & \mathbf{0}^T & -x^i\ {\mathbf{X}_\mathbf{w}}^{iT} \\ -y^i\ {\mathbf{X}_\mathbf{w}}^{iT} & x^i\ {\mathbf{X}_\mathbf{w}}^{iT} & \mathbf{0}^T \end{bmatrix} \begin{pmatrix} \mathbf{P}^1 \\ \mathbf{P}^2 \\ \mathbf{P}^3 \end{pmatrix} = \mathbf{0} \tag{3.17}$$

De estas tres ecuaciones, sólo dos son linealmente independientes. Por tanto, de cada par de puntos, se obtienen dos ecuaciones.

Si se toman las dos primeras ecuaciones de 3.17, el sistema de ecuaciones final para un conjunto de n puntos, expresado de forma matricial, queda de la siguiente forma:

$$\begin{bmatrix} \mathbf{0}^T & -z^1\ {\mathbf{X}_\mathbf{w}}^{1T} & y^1\ {\mathbf{X}_\mathbf{w}}^{1T} \\ z^1\ {\mathbf{X}_\mathbf{w}}^{1T} & \mathbf{0}^T & -x^1\ {\mathbf{X}_\mathbf{w}}^{1T} \\ \vdots & \vdots & \vdots \\ \mathbf{0}^T & -z^n\ {\mathbf{X}_\mathbf{w}}^{nT} & y^n\ {\mathbf{X}_\mathbf{w}}^{nT} \\ z^n\ {\mathbf{X}_\mathbf{w}}^{nT} & \mathbf{0}^T & -x^n\ {\mathbf{X}_\mathbf{w}}^{nT} \end{bmatrix} \begin{pmatrix} \mathbf{P}^1 \\ \mathbf{P}^2 \\ \mathbf{P}^3 \end{pmatrix} = \mathbf{0} \tag{3.18}$$

El sistema anterior es de la forma $A\mathbf{p} = \mathbf{0}$, donde A es una matriz de dimensiones $2n \times 12$.

La matriz P tiene 12 elementos, pero si ignoramos el factor de escala, tiene 11 grados de libertad. Por tanto, necesitamos 11 ecuaciones linealmente independientes para poder estimar la matriz P. Puesto que cada correspondencia $\mathbf{X_w^i} \longleftrightarrow \mathbf{x^i}$ proporciona 2 ecuaciones, como mínimo se necesitan conocer 6 puntos con sus correspondientes proyecciones en el plano imagen.

Si se conociese con exactitud la posición del par de puntos de cada correspondencia, únicamente necesitaríamos el mínimo número de ecuaciones para obtener la solución exacta de la matriz P. Sin embargo, los datos no son precisos debido a imprecisiones en la medida de los puntos del espacio como en sus proyecciones, a presencia de ruido, etc. Por tanto, se suele trabajar con sistemas sobredeterminados, donde $n \geq 6$. En este caso, la solución no será exacta y habrá que estimar el valor de la matiz P. Se toma aquella solución que minimice el error algebraico.

3. Una vez que se ha obtenido una solución para la matriz P se pueden calcular los parámetros intrínsecos y extrínsecos. Las relaciones entre cada parámetro y los elementos de la matriz P son las siguientes:

$$P = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{bmatrix} = \begin{bmatrix} \alpha_x & s & x_o \\ & \alpha_y & y_o \\ & & 1 \end{bmatrix} \begin{bmatrix} R_{11} & R_{12} & R_{13} & t_x \\ R_{21} & R_{22} & R_{23} & t_y \\ R_{31} & R_{32} & R_{33} & t_z \end{bmatrix} \tag{3.19}$$

$$\begin{cases} p_{1(1,2,3)} = \alpha_x R_{1(1,2,3)} + sR_{2(1,2,3)} + x_o R_{3(1,2,3)} \\ p_{14} = \alpha_x t_x + st_y + x_o t_z \\ p_{2(1,2,3)} = \alpha_y R_{2(1,2,3)} + y_o R_{3(1,2,3)} \\ p_{24} = \alpha_y ty + y_o t_z \\ p_{3(1,2,3)} = R_{3(1,2,3)} \\ p_{34} = tz \end{cases} \tag{3.20}$$

Además, se debe de tener en cuenta las propiedades que cumple cualquier matriz de rotación ortonormal:

- La norma de cada vector fila de la matriz de rotación vale 1.
- El producto vectorial de dos vectores filas da como resultado el tercer vector fila.
- El producto escalar de un vector fila consigo mismo es 1.
- El producto escalar de dos vectores fila distintos es 0, pues son ortogonales.

Aplicando todo esto a las ecuaciones 3.20 se obtiene:

$$\begin{aligned} t_z &= p_{34} \\ R_{3(1,2,3)} &= p_{3(1,2,3)} \\ y_o &= p_{2(1,2,3)} \cdot p_{3(1,2,3)} \\ x_o &= p_{1(1,2,3)} \cdot p_{3(1,2,3)} \\ \alpha_y &= ||p_{2(1,2,3)} - y_o R_{3(1,2,3)}|| \\ R_{2(1,2,3)} &= \frac{p_{2(1,2,3)} - y_o R_{3(1,2,3)}}{\alpha_y} \\ s &= R_{2(1,2,3)} \cdot p_{1(1,2,3)} \\ \alpha_x &= ||p_{1(1,2,3)} - sR_{2(1,2,3)} - x_o R_{3(1,2,3)}|| \\ R_{1(1,2,3)} &= \frac{p_{1(1,2,3)} - sR_{2(1,2,3)} - x_o R_{3(1,2,3)}}{\alpha_x} \\ t_y &= \frac{p_{24} - y_o tz}{\alpha_y} \\ t_x &= \frac{p_{14} - st_y - x_o t_z}{\alpha_x} \end{aligned} \tag{3.21}$$

3.1.6. Cámara afín

Al tomar una imagen de un objeto, se puede observar como sufre deformaciones en la imagen. Por ejemplo, los círculos pueden aparecer como elipses. Lo mismo ocurre con los cuadrados. Esto se debe a que, en las cámaras de proyección finita, la correspondencia entre los puntos del espacio y su imagen siguen una transformación proyectiva.

En las transformaciones proyectivas, la única propiedad geométrica que se conserva es la linealidad ("straightness"). Los ángulos, distancias y relaciones entre distancias no se mantienen. Otra propiedad que no se conserva es el paralelismo entre rectas. Dos rectas siempre se intersecan en un punto exceptuando las líneas paralelas (comúnmente se dice que este tipo de rectas se cortan "en el infinito"). Sin embargo, al tomar una imagen de líneas paralelas, se puede apreciar que éstas se cortan en un punto de la imagen. Esto se debe a que en la geometría proyectiva, un punto del infinito se proyecta en un punto finito.

Si al tomar una imagen se va aumentando la distancia focal, las deformaciones proyectivas disminuyen (en la imagen, las líneas paralelas se asemejan más a líneas paralelas). Lo mismo ocurre si se toman imágenes de objetos muy alejados de la cámara, donde la distancia entre la cámara y el objeto es grande comparada con las dimensiones de éste. En estos casos, el modelo de cámara finita se puede aproximar a una **cámara afín**.

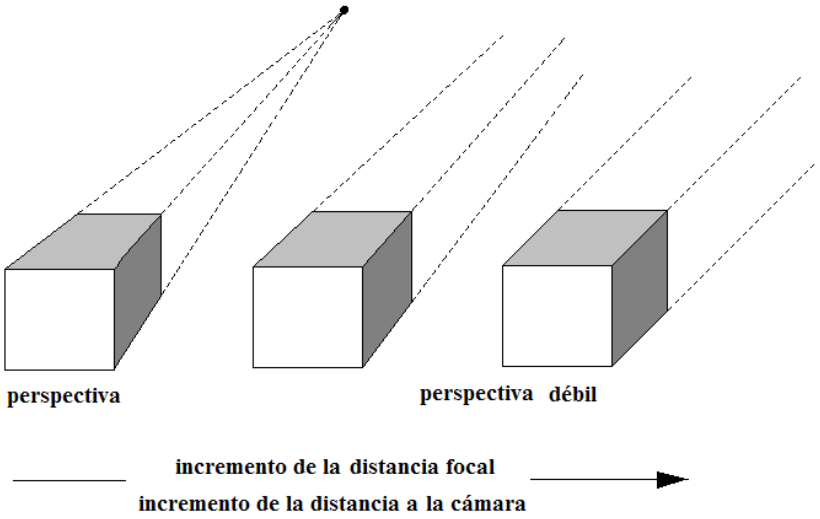


Figura 3.5: **Modelado de la cámara.** *Detalle de cómo disminuye la deformación proyectiva a medida que se aumenta la distancia focal o la distancia entre el objeto y la cámara.*

Una **cámara afín** es un tipo de cámara cuyo centro de proyección se encuentra en el infinito. La expresión general de la matriz de proyección de este tipo de cámaras es la siguiente:

$$P_A = \begin{bmatrix} \alpha_x & s & \\ & \alpha_y & \\ & & 1 \end{bmatrix} \begin{bmatrix} R_{11} & R_{12} & R_{13} & t_x \\ R_{21} & R_{22} & R_{23} & t_y \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & m_{13} & t_x \\ m_{21} & m_{22} & m_{23} & t_y \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.22)$$

La última fila de la matriz de proyección, P_A, es siempre de la forma $(0,\ 0,\ 0,\ 1)$. Esta matriz tiene 8 grados de libertad, correspondientes a los 8 elementos no nulos y distintos de la unidad.

3.1.7. Propiedades de la cámara afín

En una cámara afín, el plano en el infinito en el espacio proyectivo $\mathbb{P}^3$ se mapea con puntos en el infinito en la imagen. Esta propiedad se puede demostrar fácilmente haciendo uso de las coordenadas homogéneas:

- En coordenadas homogéneas, los puntos cuya última coordenada es nula se conocen como **puntos ideales** o **puntos en el infinito**. Por tanto, un punto ideal en el espacio 3D se denota como $(X, Y, Z, 0)^T$.
- Para demostrar que cualquier punto en el infinito se corresponde con un punto en el infinito en la imagen, basta con multiplicar dicho punto con la matriz de proyección de una cámara afín:

$$P_A \begin{pmatrix} X \\ Y \\ Z \\ 0 \end{pmatrix} = \begin{bmatrix} m_{11} & m_{12} & m_{13} & t_x \\ m_{21} & m_{22} & m_{23} & t_y \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 0 \end{pmatrix} = \begin{pmatrix} Xm_{11} + Ym_{12} + Zm_{13} \\ Xm_{21} + Ym_{22} + Zm_{23} \\ 0 \end{pmatrix} \tag{3.23}$$

- El punto de proyección que se ha obtenido tiene su última coordenada nula, y por tanto, se corresponde con un punto ideal en $\mathbb{P}^2$.

En el modelo de cámara finita, el plano imagen es el plano formado por todos los puntos que se proyectan en el infinito. Por tanto, en una cámara afín, el plano imagen se encuentra en el infinito. Es más, puesto que el centro de proyección está contenido en el plano imagen, éste se encuentra también en el infinito. De todo esto, se pueden extraer las siguientes conclusiones:

1. Cualquier modelo de cámara proyectiva cuyo plano principal se encuentre en el infinito es una cámara afín.
2. Las líneas paralelas en $\mathbb{P}^3$ se proyectan en la imagen como líneas paralelas. Esto se debe a que en el espacio 3D, las líneas paralelas se cortan en el infinito, y cualquier punto en el infinito se proyecta también en el infinito en $\mathbb{P}^2$.

3.2. Transformaciones proyectivas en 2D

La geometría 2D proyectiva estudia las propiedades del plano proyectivo $\mathbb{P}^2$ que son invariantes bajo un grupo de transformaciones conocidas como **transformaciones proyectivas** u **homografías** [Hartley and Zisserman, 2004].

Se dice que una correspondencia $h : \mathbb{P}^2 \rightarrow \mathbb{P}^2$ es una proyectividad si existe una matriz no singular H, de dimensiones 3×3, tal que, para cualquier punto en $\mathbb{P}^2$, representado por el vector $\mathbf{x}$, se cumple que $h(\mathbf{x}) = H\mathbf{x}$. Por tanto, una proyectividad es una correspondencia lineal entre vectores expresados en coordenadas homogéneas de 3 dimensiones:

$$\mathbf{x}' = H\mathbf{x} \qquad \Longrightarrow \qquad \begin{pmatrix} x_1' \\ x_2' \\ x_3' \end{pmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \tag{3.24}$$

Puesto que H es una matriz no singular, las transformaciones proyectivas son invertibles y su inversa es a su vez una proyectividad. Además, si multiplicamos la matriz H por un factor no nulo, la transformación proyectiva no varía puesto que, al estar trabajando con coordenadas homogéneas, puntos que difieren en un factor de escala no nulo son equivalentes. Por tanto, si normalizamos la matriz H de forma que $h_{33} = 1$, dicha matriz tendrá 8 grados de libertad en el caso más general, a falta de definir el factor de escala.

3.2.1. Aplicaciones de las transformaciones proyectivas 2D

Un ejemplo de proyectividad es el que se muestra en la figura 3.6.

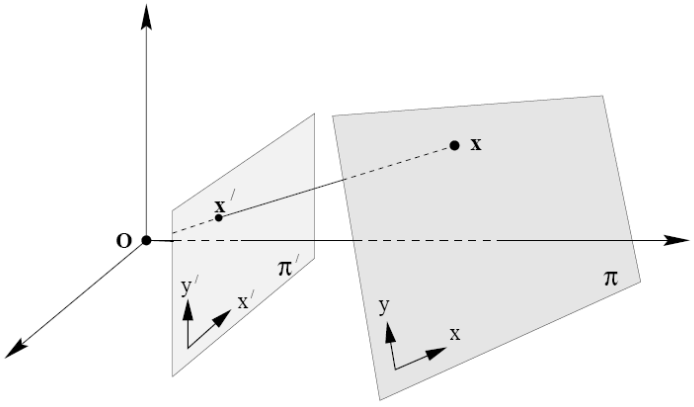


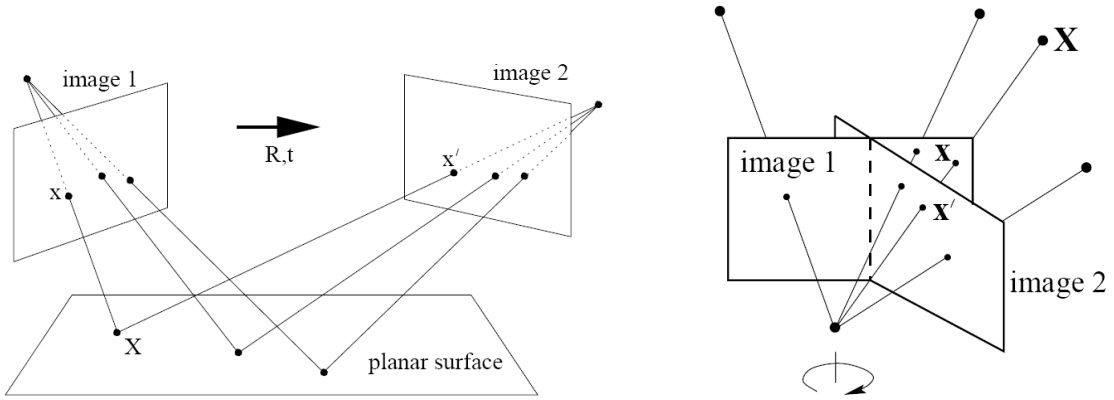
Figura 3.6: **Transformaciones proyectivas 2D.** *La proyección central genera una correspondencia entre los puntos de un plano y los puntos de otro plano.*

La proyección de rayos que pasan por un mismo punto (proyección central) en dos planos distintos definen una homografía, es decir, existe una correspondencia entre los puntos de proyección de ambos planos generados por el mismo rayo. Si se define un sistema de coordenadas para cada plano y se expresan los puntos en coordenadas homogéneas, la relación entre los puntos generados por dicha proyección central viene dada por la siguiente expresión:

$$\mathbf{x}' = H\mathbf{x} \tag{3.25}$$

donde H es la matriz de 3×3 no singular de homografía.

En las siguientes figuras, se muestran más ejemplos de transformaciones proyectivas. Si se toman dos imágenes de un mismo plano en el espacio tridimensional (con diferentes centros de proyección), tal y como se muestra en la figura 3.7(a), existe una relación directa entre ambas imagenes. Dicha relación viene dada por la expresión 3.25. De la misma forma, existe una transformación proyectiva entre imágenes con el mismo centro óptico en las que se ha aplicado una rotación a la cámara respecto de su centro o se ha variado la distancia focal entre una imagen y la otra, tal y como se muestra en la figura 3.7(b).



(a) (b)

Figura 3.7: **Transformaciones proyectivas 2D.** *(a) Transformación proyectiva inducida por la proyección de un plano en dos imagenes distintas. (b) Transformación proyectiva entre dos imagenes con el mismo centro óptico pero con distancias focales diferente o posiciones distintas de la cámara (rotaciones de ésta sobre su centro óptico).*

3.2.2. Jerarquía de las transformaciones proyectivas

Las transformaciones proyectivas forman un grupo denominado **grupo lineal proyectivo**. Dentro de este grupo, encontramos varios subgrupos que definen una jerarquía. Una forma de caracterizar dichos subgrupos es mediante sus invariantes, es decir, describiendo las transformaciones en términos de los elementos o medidas que permanecen invariantes tras aplicar la transformación.

La jerarquía de los distintos subgrupos es la que se enumera a continuación, comenzando por la transformación más simple hasta llegar al caso más general:

1. **Isometría.**
2. **Similitud.**
3. **Transformación Afín.**
4. **Transformación Proyectiva** u **Homografía.**

A continuación, se va a describir las propiedades de las transformaciones afines y proyectivas, pues son los dos tipos que se van a utilizar a lo largo de todo el proyecto.

3.2.2.1. Transformación Afín

Una **Transformación Afín** (o simplemente una **afinidad**) se compone de una transformación lineal no singular y una traslación:

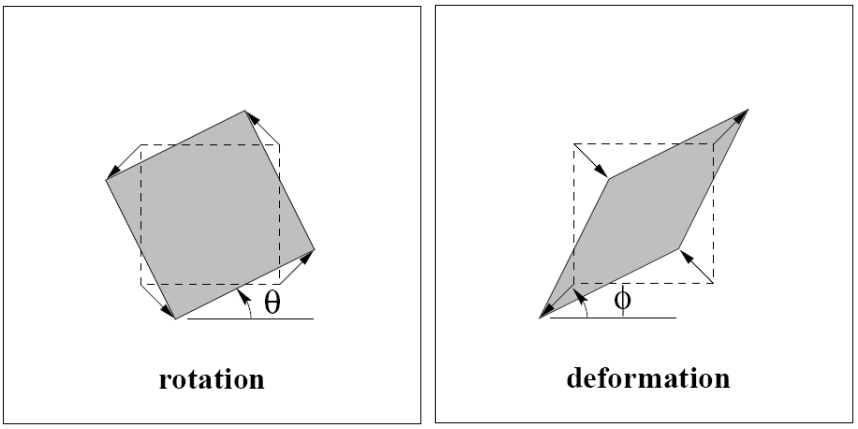
$$\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \begin{bmatrix} a_{11} & a_{12} & t_x \\ a_{21} & a_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} \tag{3.26}$$

Como se puede observar, una transformación afín tiene 6 grados de libertad correspondientes a los 6 elementos de la matriz que define dicha transformación. La expresión anterior abreviada queda de la siguiente forma:

$$\mathbf{x}' = H_A\mathbf{x} \qquad \Longrightarrow \qquad H_A = \begin{bmatrix} A & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \tag{3.27}$$

donde A es una matriz de 2×2 no singular.

En la figura 3.8 se muestran los efectos de dicha matriz A en la transformación afín.



(a) (b)

Figura 3.8: **Transformaciones proyectivas 2D.** *(a) Rotación* $R(\theta)$. *(b) Deformación* $R(-\phi)DR(\phi)$. *Las direcciones en las que se aplica el escalado en la deformación son ortogonales.*

La matriz A se puede descomponer en varias transformaciones fundamentales: 3 rotaciones y un escalado no isotrópico:

$$A = R(\theta)R(-\phi)DR(\phi) \tag{3.28}$$

donde $R(\theta)$ y $R(\phi)$ son las matrices de rotación de un ángulo θ y ϕ respectivamente y D es la matriz diagonal de escalado:

$$D = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}$$

Por tanto, la matriz A es una concatenación de una rotación de los ejes un ángulo ϕ seguida de un escalado de dichos ejes rotados x e y por un factor λ_1 y λ_2 respectivamente (el ángulo ϕ especifica la dirección de escalado). A continuación, se deshace la rotación inicial (los ejes se rotan un ángulo $-\phi$) y se aplica de nuevo una rotación de un ángulo θ.

Invariantes. Puesto que la afinidad incorpora un escalado no isotrópico, al aplicar la transformación no se conserva ni la relación entre longitudes de líneas ni los ángulos que forman. Sin embargo, sí se mantienen invariantes las siguientes medidas:

- **Líneas paralelas** - Si consideramos 2 líneas paralelas, éstas se cortan en un punto situado en el infinito de la forma $(x_1, x_2, 0)^T$. Bajo una transformación afín, este punto se mapea con otro punto en el infinito. Por tanto, las líneas paralelas seguirán siendo paralelas bajo una transformación afín.

- **Relación de longitud entre segmentos de líneas paralelas** - El escalado de un segmento de línea depende únicamente del ángulo que forman la dirección de la línea y la dirección de escalado. Supongamos que una línea forma un ángulo α con la dirección del eje x de escalado, por tanto, el factor de escala que se aplica a dicho segmento será:

$$\sqrt{\lambda_1^2 cos^2\alpha + \lambda_2^2 sen^2\alpha}$$

  Esta magnitud de escalado es común a todas las líneas con la misma dirección y por tanto, este término se cancela al obtener las relaciones entre las longitudes de segmentos paralelos.

- **Relación de áreas** - Este invariante se deduce directamente de la expresión 3.28. Las rotaciones y traslaciones no afectan al área, aunque no ocurre lo mismo con el escalado. Bajo una transformación afín, cualquier área se escala por un factor $\lambda_1\lambda_2$, que es igual al $det(A)$. Por tanto, al calcular la relación entre áreas de distintos elementos, el factor de escalado desaparece.

3.2.2.2. Transformación Proyectiva u Homografía

Una **Transformación Proyectiva** viene definida por la expresión 3.24. Es la transformación lineal y no singular más general. Utilizando una expresión abreviada de 3.24, se obtiene:

$$\mathbf{x}' = H_p\mathbf{x} \qquad \Longrightarrow \qquad H_p = \begin{bmatrix} A & \mathbf{t} \\ \mathbf{v}^T & \upsilon \end{bmatrix} \tag{3.29}$$

donde el vector $\mathbf{v}$ es de la forma $\mathbf{v} = (\upsilon_1, \upsilon_2)^T$. Normalizamos la matriz H_p de forma que $\upsilon = 1$, la matriz estará compuesta por 8 elementos distintos de la unidad. Por tanto, H_p tiene 8 grados de libertad a falta de definir el factor de escala.

Invariantes. Uno de los invariantes más importantes de las transformaciones proyectivas es la relación cruzada de 4 puntos colineales. También se mantiene la propiedad de colinealidad, es decir, una recta se proyecta en otra recta.

Una diferencia clara e importante respecto a las transformaciones afines es que las líneas paralelas dejan de proyectarse como líneas paralelas. Esto se debe a que bajo una transformación proyectiva, un punto en el infinito $(x_1, x_2, 0)^T$ se corresponde con un punto

finito, y por tanto, las líneas paralelas se proyectan como rectas que se cortan en un punto finito:

$$\begin{bmatrix} A & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \\ 0 \end{pmatrix} = \begin{pmatrix} A \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \\ \upsilon_1 x_1 + \upsilon_2 x_2 \end{pmatrix} \tag{3.30}$$

3.2.3. Estimación de la matriz de transformación

Consideremos un conjunto de N puntos de una imagen de los cuales se conocen sus correspondencias en una segunda imagen: $\mathbf{x_i} \longleftrightarrow \mathbf{x'_i}$. El objetivo es determinar la matriz H que relaciona los dos conjuntos de puntos entre sí:

$$\mathbf{x'_i} = H\mathbf{x_i}$$

Lo primero que hay que determinar es el número mínimo de puntos necesarios para estimar la matriz H. Este número depende del tipo de transformación proyectiva pues se necesitan tantas ecuaciones como grados de libertad tenga la matriz H. Además, de cada par de correspondencias $\mathbf{x_i} \longleftrightarrow \mathbf{x'_i}$ se obtienen 2 ecuaciones linealmente independientes (un punto 2D tiene 2 grados de libertad, uno por la componente x y otro por la y). Por tanto, el número mínimo de puntos necesarios es:

- **Afinidad**: 6 grados de libertad $\longrightarrow$ 3 puntos.
- **Homografía**: 8 grados de libertad $\longrightarrow$ 4 puntos.

Si se conociese con exactitud la posición del par de puntos de cada correspondencia, únicamente se necesitaría el número mínimo de ecuaciones para obtener la solución exacta de la matriz H. Sin embargo, los datos no son precisos (discretización de la imagen, presencia de ruido, etc) por lo que se suele trabajar con sistemas sobredimensionados. En este caso, la solución no será exacta y habrá que estimar el valor de la matiz H, tomando aquella solución que minimice el error algebraico.

El proceso de obtención de la matriz H, para el caso más general (8 grados de libertad) es el siguiente:

1. Se conoce un conjunto de N correspondencias entre puntos de dos imágenes distintas: $\mathbf{X_w}^i \longleftrightarrow \mathbf{x}^i$. La relación de correspondencia entre cada par de puntos viene dada por la matriz de homografía H:

$$\mathbf{x'_i} = H\mathbf{x_i} \qquad \Longrightarrow \qquad \mathbf{x'_i} = \begin{bmatrix} \mathbf{h}^{1T}\,\mathbf{x_i} \\ \mathbf{h}^{2T}\,\mathbf{x_i} \\ \mathbf{h}^{3T}\,\mathbf{x_i} \end{bmatrix} \tag{3.31}$$

donde:

- $\mathbf{h}^{jT} \rightarrow$ es un vector fila de dimensión 3 que se corresponde con la fija j de la matriz H.

2. Hay que tener en cuenta, que la ecuación anterior relaciona vectores en coordenadas homogéneas. Por tanto, los vectores $\mathbf{x'_i}$ y $H\mathbf{x_i}$ tienen la misma dirección pero difieren en magnitud por un factor no nulo. De esta forma, para poder obtener una solución de la matriz H, tenemos que realizar el producto vectorial de ambos vectores para eliminar la indeterminación de la escala.

   Siendo $\mathbf{x'_i} = (x'_i, y'_i, z'_i)$, el producto vectorial se puede expresar de forma explicita de la siguiente manera:

$$\mathbf{x'_i} \times H\mathbf{x_i} = \mathbf{0} \qquad \Longrightarrow \qquad \begin{pmatrix} y'_i\, \mathbf{h}^{3T}\, \mathbf{x_i} \; - \; z'_i\, \mathbf{h}^{2T}\, \mathbf{x_i} \\ z'_i\, \mathbf{h}^{1T}\, \mathbf{x_i} \; - \; x'_i\, \mathbf{h}^{3T}\, \mathbf{x_i} \\ x'_i\, \mathbf{h}^{2T}\, \mathbf{x_i} \; - \; y'_i\, \mathbf{h}^{1T}\, \mathbf{x_i} \end{pmatrix} = \mathbf{0} \tag{3.32}$$

   Puesto que $\mathbf{h}^{jT}\, \mathbf{x_i} = \mathbf{x_i}^T\, \mathbf{h}^j$ para $j = 1, \ldots, 3$, de la expresión 3.32 obtenemos un sistema de tres ecuaciones donde los elementos de la matriz H son las incógnitas:

$$\begin{bmatrix} \mathbf{0}^T & -z'_i\, \mathbf{x_i}^T & y'_i\, \mathbf{x_i}^T \\ z'_i\, \mathbf{x_i}^T & \mathbf{0}^T & -x'_i\, \mathbf{x_i}^T \\ -y'_i\, \mathbf{x_i}^T & x^i\, \mathbf{X_w}^{iT} & \mathbf{0}^T \end{bmatrix} \begin{pmatrix} \mathbf{h}^1 \\ \mathbf{h}^2 \\ \mathbf{h}^3 \end{pmatrix} = \mathbf{0} \tag{3.33}$$

   De estas tres ecuaciones, sólo dos son linealmente independientes. Por tanto, de cada par de puntos, se obtienen dos ecuaciones. Si se toman las dos primeras ecuaciones de 3.33, el sistema final para un conjunto de n puntos, expresado de forma matricial, queda de la siguiente forma:

$$\begin{bmatrix} \mathbf{0}^T & -z'_1\, \mathbf{x_1}^T & y'_1\, \mathbf{x_1}^T \\ z'_1\, \mathbf{x_1}^T & \mathbf{0}^T & -x'_1\, \mathbf{x_1}^T \\ \vdots & \vdots & \vdots \\ \mathbf{0}^T & -z'_n\, \mathbf{x_n}^T & y'_n\, \mathbf{x_n}^T \\ z'_n\, \mathbf{x_n}^T & \mathbf{0}^T & -x'_n\, \mathbf{x_n}^T \end{bmatrix} \begin{pmatrix} \mathbf{h}^1 \\ \mathbf{h}^2 \\ \mathbf{h}^3 \end{pmatrix} = \mathbf{0} \tag{3.34}$$

   El sistema anterior es de la forma $A\mathbf{h} = \mathbf{0}$, donde A es una matriz de dimensiones $2n \times 8$.

3.3. Matriz fundamental y geometría de múltiples cámaras

Como se vio en la sección 3.1 de este capítulo, la formación de las imágenes consiste en una representación bidimensional del mundo 3D, perdiéndose la información de profundidad. Por tanto, un punto cualquiera del espacio $\mathbf{X}_i = (x, y, z, 1)$ se proyecta en un punto $\mathbf{m}_i = (x', y', 1)$ del plano imagen de la siguiente forma:

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \lambda K\big(R \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} + T\big) \qquad\qquad P = K[R|T]$$

donde P es la matriz de proyección de la cámara. El valor de la constante λ se calcula a partir de la coordenada z del punto $3D$.

Por tanto, conociendo la matriz P se puede obtener el punto de proyección en el plano imagen de cualquier punto del espacio tridimensional. Sin embargo, a partir del punto de proyección $\mathbf{m}_i$ es imposible determinar $\mathbf{X}_i$ pues faltaría la información de profundidad, λ (el punto $\mathbf{X}_i$ no es el único que se proyecta en $\mathbf{m}_i$, sino que todos los puntos que definen el rayo de proyección de $\mathbf{X}_i$ se proyectan en $\mathbf{m}_i$).

Para poder obtener las coordenadas $3D$ de un punto se necesitan conocer la proyección de dicho punto, al menos, en dos cámaras situadas en distintas posiciones. Además, ambas proyecciones estarán relacionadas a través de la **matriz fundamental**.
La utilización de dos cámaras para extraer la información tridimensional del entorno se denomina **Visión Estereoscópica**.

3.3.1. Geometría epipolar

Supongamos que tenemos dos cámaras situadas en distintas posiciones, cuyos centros de proyección son $\mathbf{C}$ y $\mathbf{C}'$. Tomamos desde cada cámara una imagen de un punto $\mathbf{X}$ del espacio, siendo $\mathbf{x}$ las coordenadas del punto proyectado en la primera imagen y $\mathbf{x}'$ las de la segunda. La **geometría epipolar** describe la relación geométrica entre ambas proyecciones del mismo punto. En la figura 3.9 se muestra un esquema de esta situación.

Se va a definir una serie de términos importantes en la geometría epipolar:

- **Baseline** - Es la línea base que une los centro de las cámaras.
- **Epipolo** - Es el punto de intersección entre el plano imagen y la línea base. En concreto, es la proyección de cada uno de los centros de una cámara en el plano imagen de la otra cámara.

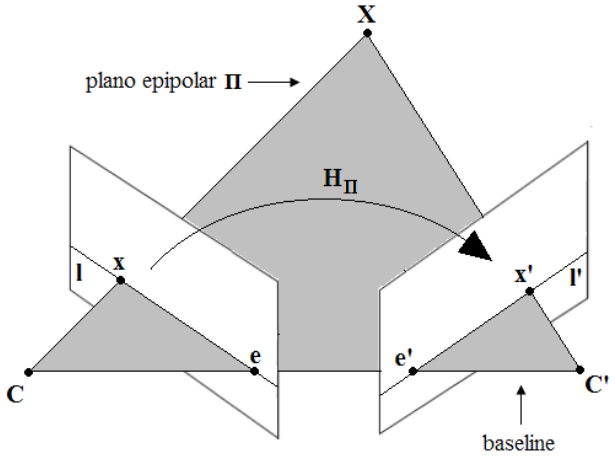
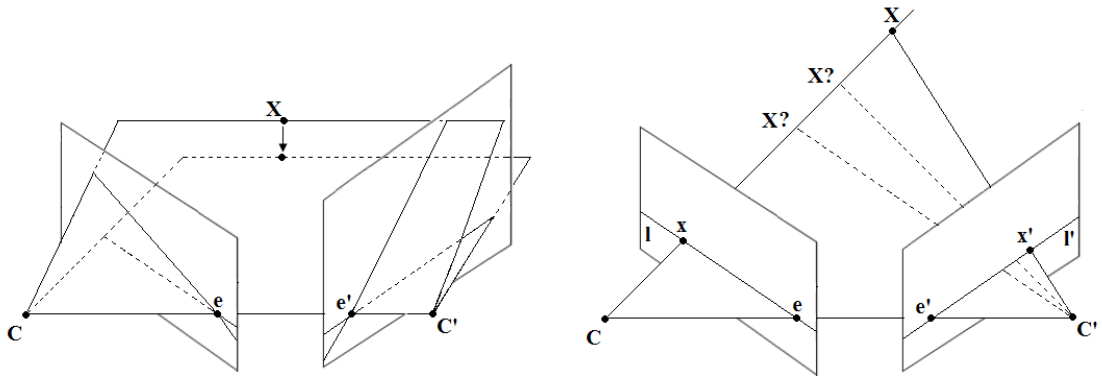


Figura 3.9: **Matriz fundamental y geometría de múltiples cámaras.** *Geometría epipolar.*

- **Plano epipolar** - Es el plano que contiene a la línea base. Se puede ver en la figura 3.10(a) que a medida que el punto $\mathbf{X}$ del espacio varíe, el plano epipolar cambia de posición (rota en torno a la línea base) Toda esta familia de posibles planos epipolares forman un **"epipolar pencil"**.

- **Linea epipolar** - Es la línea de intersección entre el plano imagen de cada cámara y el plano epipolar. Además, la línea epipolar contiene al epipolo.



(a) (b)

Figura 3.10: **Matriz fundamental y geometría de múltiples cámaras.** *Geometría epipolar. (a) A medida que la posición del punto tridimensional* $\mathbf{X}$ *varía, el plano epipolar rota en torno a la línea "baseline". (b) El punto 3D del espacio y el centro* $\mathbf{C}$ *de la primera cámara definen el rayo de proyección del punto* $\mathbf{x}$*. La proyección de este rayo en la segunda imagen se corresponde con la línea* $\mathbf{l}'$*. La proyección del punto 3D en la segunda imagen debe estar contenido en esta línea.*

Se puede apreciar en la figura 3.9 que tanto el punto 3D, como sus proyecciones $\mathbf{x}$ y $\mathbf{x}'$ son coplanares (los tres puntos pertenecen al plano epipolar). Lo mismo ocurre con los rayos de proyección y la línea "baseline".

Supongamos que únicamente conocemos la proyección $\mathbf{x}$ del punto 3D en la primera imagen, tal y como se muestra en la figura 3.10(b). La proyección de este mismo punto en la segunda cámara no podrá estar situado en cualquier punto de la imagen sino que su posición queda delimitada por las siguientes restricciones:

- La posición del plano epipolar se determina con el rayo de proyección de la primera cámara y la línea "baseline", pues ambas deben estar contenidas en el plano.
- El rayo de proyección que genera el punto $\mathbf{x}'$ en la segunda cámara también debe pertenecer al plano epipolar. El plano epipolar y el plano imagen de la segunda cámara se intersecan en la recta $\mathbf{l}'$, por tanto, $\mathbf{x}'$ pertenecerá a dicha recta. Además, $\mathbf{l}'$ se corresponde con la proyección en la segunda imagen del rayo de proyección de la primera imagen. A esta recta se la denomina línea epipolar de $\mathbf{x}$.

3.3.2. Matriz Fundamental

La **matriz fundamental**, denotada por F, permite establecer la relación matemática entre un mismo punto proyectado en las dos imágenes.

Para obtener la matriz fundamental tenemos que tener en cuenta las características que presenta la geometría epipolar:

- Dado un par de imágenes, cualquier punto $\mathbf{x}$ en la primera imagen tiene asociada una línea epipolar en la segunda imagen, $\mathbf{l}'$.
- Cualquier punto $\mathbf{x}'$ de la segunda imagen que esté relacionado con $\mathbf{x}$ debe pertenecer a la línea epipolar $\mathbf{l}'$.
- La línea epipolar es la proyección en la segunda imagen del rayo que genera $\mathbf{x}$ en la primera imagen.

Por tanto, hay que encontrar la correspondencia entre un punto en una de las imágenes y su correspondiente línea epipolar en la otra imagen. Dicha correspondencia queda definida por la matriz fundamental F:

$$\mathbf{x} \mapsto \mathbf{l}' \qquad \Longrightarrow \qquad \mathbf{l}' = F\mathbf{x} \tag{3.35}$$

donde la matriz fundamental, a falta de definir el factor de escala, es de la forma:

$$F = \begin{pmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & 1 \end{pmatrix}$$

A partir de la correspondencia establecida en la ecuación 3.35, se puede obtener la relación existente entre $\mathbf{x}$ y $\mathbf{x}'$:

$$\mathbf{x} \longleftrightarrow \mathbf{x}' \qquad \Longrightarrow \qquad \mathbf{x}'^T F \mathbf{x} = 0 \tag{3.36}$$

La ecuación anterior se obtiene a partir del siguiente razonamiento:

- El punto $\mathbf{x}$ es la proyección de un punto tridimensional en una primera imagen. Si el punto $\mathbf{x}'$ se corresponde con la proyección del mismo punto en una segunda imagen, este punto pertenecerá a la epipolar $\mathbf{l}' = F\mathbf{x}$ correspondiente al punto $\mathbf{x}$.
- Si un punto cualquiera $\mathbf{x}$ pertenece a una recta $\mathbf{l}$ se cumple que $\mathbf{x}^T\mathbf{l} = 0$. Por tanto, si aplicamos esta condición a $\mathbf{l}'$ y $\mathbf{x}'$ se obtiene:

$$\mathbf{x}'^T\mathbf{l}' = \mathbf{x}'^T F\mathbf{x} = 0 \qquad \Longrightarrow \qquad \mathbf{x}'^T F\mathbf{x} = 0$$

3.3.3. Cálculo de la matriz fundamental

La matriz fundamental queda definida por la ecuación 3.36:

$$\mathbf{x}'^T F\mathbf{x} = 0$$

Dado un conjunto n de correspondencias $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$ entre dos imágenes, se puede calcular la matriz F. En particular, para un par de correspondencias de la forma:

$$\mathbf{x} = (x, y, 1)^T$$

$$\mathbf{x}' = (x', y', 1)^T$$

Si se desarrolla la ecuación se obtiene:

$$x'xf_{11} + x'yf_{12} + x'f_{13} + y'xf_{21} + y'yf_{22} + y'f_{23} + xf_{31} + yf_{32} + f_{33} = 0 \tag{3.37}$$

Por tanto, de cada par de correspondencias se obtiene una ecuación linealmente independiente. Como la matriz fundamental tiene 8 grados de libertad, al menos se necesitan 8 correspondencias para obtener una solución. Utilizando la notación matricial para la ecuación 3.37, para un conjunto de n correspondencias se obtiene:

$$\begin{pmatrix} x'_1x_1 & x'_1y_1 & x'_1 & y'_1x_1 & y'_1y_1 & y'_1 & x_1 & y_1 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x'_nx_n & x'_ny_n & x'_n & y'_nx_n & y'_ny_n & y'_n & x_n & y_n & 1 \end{pmatrix} \mathbf{f} = 0 \tag{3.38}$$

donde:

$$\mathbf{f} = (f_{11},\ f_{12},\ f_{13},\ f_{21},\ f_{22},\ f_{23},\ f_{31},\ f_{32},\ 1)$$

El sistema de ecuaciones anterior es de la forma $A\mathbf{x} = 0$. Si el rango de la matriz A es 8 y las coordenadas de los puntos son correctas, la solución es única. Sin embargo, debido a presencia de ruido e imprecisiones de medida, los datos no son exactos por lo que el rango de A debe ser mayor que 8 (sistema sobredeterminado). En este caso, se obtiene la solución de mínimos cuadrados utilizando la descomposición SVD de la matriz A.

3.3.4. Visión Estéreo y reconstrucción 3D

Al tomar una imagen de un punto del espacio tridimensional $\mathbf{X} = (X, Y, Z)^T$ se pierde la información de profundidad. La imagen $\mathbf{x}$ del punto tridimensional se forma a partir de la intersección del rayo de proyección (que une el punto $\mathbf{X}$ con el centro de la cámara) y el plano imagen. Por tanto, si te toma cualquier otro punto del rayo de proyección, genera el mismo punto en la imagen. Por esta razón, a partir del punto proyectado (conociendo también la matriz de la cámara P), no es posible volver a obtener las coordenadas del punto 3D. Necesitamos más información para resolver la indeterminación de la distancia entre la cámara y el punto 3D. En la figura 3.11 se muestra un esquema aclaratorio.

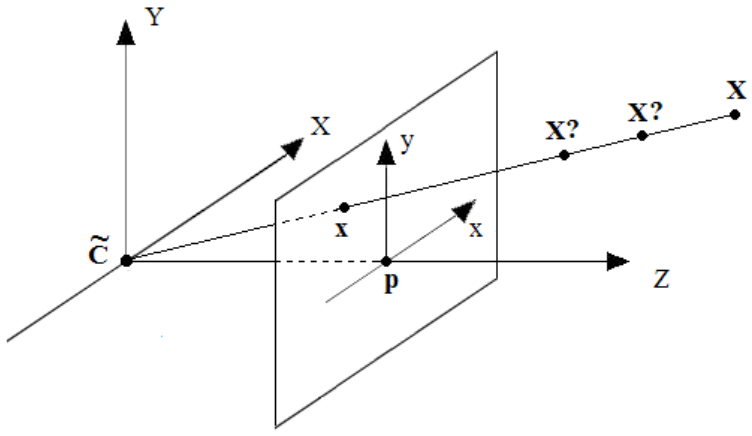


Figura 3.11: **Matriz fundamental y geometría de múltiples cámaras.** *Proceso de formación de una imagen. El punto* $\mathbf{X}$ *se proyecta en el punto* $\mathbf{x}$*. Se puede apreciar que cualquier otro punto perteneciente al rayo de proyección genera el mismo punto imagen*

Si ahora se toma otra imagen del mismo punto con una segunda cámara situada en otro lugar, de la que también se conoce su matriz de proyección P', se obtiene otro punto $\mathbf{x}'$ en el plano imagen de la segunda cámara. Al igual que pasaba antes, es imposible conocer el punto del espacio 3D a partir de su proyección pues existen infinitas soluciones. Sin embargo, juntando las informaciones que proporcionan cada una de las imágenes, se puede determinar la posición exacta de $\mathbf{X}$. Como se puede ver en la figura 6.7, el punto tridimensional $\mathbf{X}$ se corresponde con el punto de intersección entre los rayos de proyección de cada una de las cámaras.

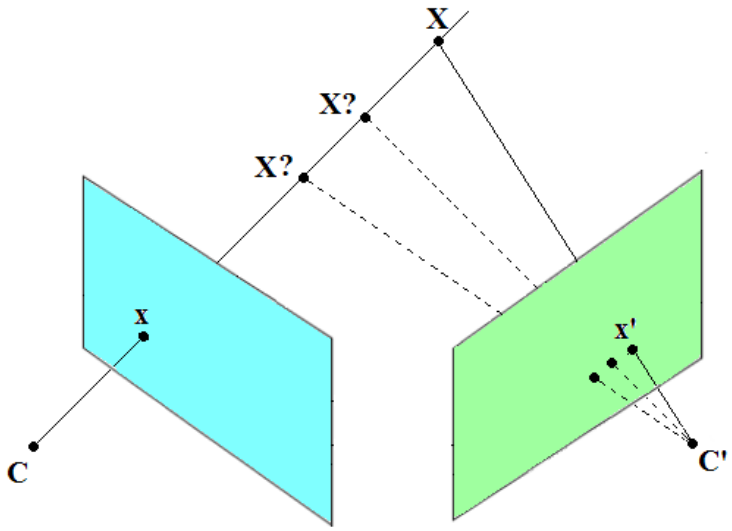


Figura 3.12: **Matriz fundamental y geometría de múltiples cámaras.** *A partir de la información aportada por dos imágenes del mismo punto tomadas desde posiciones distintas se puede obtener las coordenadas 3D del punto del espacio*

La utilización de dos cámaras para extraer la información tridimensional del entorno se denomina **Visión Estereoscópica**.

En resumen, se tiene dos cámaras situadas en lugares distintos del espacio, de las que se conoce sus matrices de proyección P y P'. Un punto $\mathbf{X}$ del espacio se proyecta en cada una de las cámaras de la siguiente forma:

$$\begin{array}{lcl} \text{Camara 1} & \longrightarrow & \mathbf{x} = P\mathbf{X} \\ \text{Camara 2} & \longrightarrow & \mathbf{x}' = P'\mathbf{X} \end{array} \tag{3.39}$$

Si se conocen el par de correspondencias $\mathbf{x} \leftrightarrow \mathbf{x}'$, se puede calcular $\mathbf{X}$. El par $\mathbf{x} \leftrightarrow \mathbf{x}'$ se puede obtener a través de la matriz fundamental F sabiendo que ambas proyecciones se relacionan entre sí de la siguiente manera:

$$\mathbf{x}'^T F \mathbf{x} = 0$$

Debido a los posibles errores en las medidas de $\mathbf{x}$ y $\mathbf{x}'$ e imprecisiones a la hora de obtener las matrices de proyección de cada cámara y la matriz fundamental, no habrá un punto $\mathbf{X}$ que satisfaga exactamente las ecuaciones 3.39, sino que se obtendrá un valor estimado del punto $\hat{\mathbf{X}}$ que minimice el error.

Para estimar el punto $\mathbf{X}$ se va a utilizar el método lineal de triangulación. Se conocen los puntos de proyección de $\mathbf{X}$ en cada una de las imágenes (ecuación 3.39). Hay que tener en cuenta, que cada ecuación de 3.39 relaciona vectores en coordenadas homogéneas. Por tanto, los vectores $\mathbf{x}$ y $P\mathbf{X}$ tienen la misma dirección pero difieren en magnitud por un factor no nulo. Lo mismo ocurre con $\mathbf{x}'$ y $P'\mathbf{X}$. De esta forma, para poder obtener una solución del punto tridimensional, hay que realizar el producto vectorial de ambos vectores para eliminar la indeterminación de la escala. Por ejemplo, para la primera cámara el producto vectorial $\mathbf{x} \times P\mathbf{X}$ genera las siguientes ecuaciones:

$$\begin{aligned} x(\mathbf{p}^{3T}\mathbf{X}) - (\mathbf{p}^{1T}\mathbf{X}) &= 0 \\ y(\mathbf{p}^{3T}\mathbf{X}) - (\mathbf{p}^{2T}\mathbf{X}) &= 0 \\ x(\mathbf{p}^{2T}\mathbf{X}) - y(\mathbf{p}^{1T}\mathbf{X}) &= 0 \end{aligned} \tag{3.40}$$

donde $\mathbf{p}^{iT}$ se corresponde con la fila i de la matriz P. Tomando de cada cámara dos ecuaciones de la expresión anterior (por ejemplo, las dos primeras) se construye el sistema de ecuaciones final. Dicho sistema es del tipo $A\mathbf{x} = 0$, donde la matriz A es de la forma:

$$A = \begin{bmatrix} x\mathbf{p}^{3T} - \mathbf{p}^{1T} \\ y\mathbf{p}^{3T} - \mathbf{p}^{2T} \\ x'\mathbf{p}'^{3T} - \mathbf{p}'^{1T} \\ y'\mathbf{p}'^{3T} - \mathbf{p}'^{2T} \end{bmatrix} \tag{3.41}$$

El sistema anterior es redundante pues únicamente serían necesarias 3 ecuaciones linealmente independientes. Una forma de resolver el sistema, es utilizar la descomposición SVD de la matriz A.

Capítulo 4

Detección de puntos de interés en la imagen

El reconocimiento de objetos por visión artificial y la capacidad de entender el entorno a través de imágenes se han convertido en tareas de gran importancia y que están presentes en infinidad de aplicaciones relacionadas con el campo de la medicina, la industria y el transporte, en defensa, seguridad y vigilancia, en entretenimiento, data mining, en la interacción hombre-máquina, en robótica, etc.

Uno de los objetivos a la hora de implementar un método de reconocimiento de objetos es que éste sea invariante, en el mayor grado posible, a ciertas situaciones que se dan en este tipo de aplicaciones:

- En la mayoría de los casos, las escenas a analizar no son conocidas a priori.
- Los objetos de la escena pueden presentar distintas apariencias en función de:
  - La escala de observación.
  - La orientación del objeto: rotaciones y traslaciones.
  - Deformaciones proyectivas.
- Se pueden producir modificaciones en la iluminación en la escena, lo que se traduce en cambios de contraste y brillo.
- Las escenas pueden tener un alto grado de complejidad, compuestas por numerosos objetos y estos pueden sufrir distintos grados de oclusiones.
- Presencia de ruido.

En este proyecto, se propone utilizar un método basado en características invariantes para resolver el problema del reconocimiento de objetos. En concreto, se utiliza el método **SIFT** (Scale Invariant Feature Transform) propuesto por Lowe [Lowe, 2004]. Este método utiliza descriptores de características locales que presentan las siguientes propiedades:

- Invarianza completa a cambios de escala y orientación.
- Buen comportamiento frente a transformaciones afines y cambios de perspectiva inferiores a 60^o para objetos planares.
- Invarianza a cambios de iluminación.
- Presentan gran inmunidad al ruido.
- La cantidad de descriptores SIFT que se generan en una imagen es muy alta, lo que permite el reconocimiento de objetos que se encuentren parcialmente ocultos.
- Los descriptores SIFT son bastante distintivos entre ellos de forma que cada descriptor presenta una gran probabilidad de correspondencia correcta, incluso en grandes bases de datos formadas por numerosos descriptores.
- Para objetos tridimensionales, presenta una respuesta buena para cambios de perspectiva inferiores a 20^o.

Los pasos necesarios para obtener los descriptores SIFT son los siguientes:

1. **Detección de extremos en el espacio de escala** - El primer paso es la búsqueda de puntos en toda la imagen que presenten invarianza a cambios de escala. Cada punto encontrado, será un posible punto de interés.
2. **Localización de los puntos de interés** - Para cada candidato, se determina su posición exacta en la imagen y su escala. Los puntos de interés se seleccionan en base a su estabilidad.
3. **Asignación de orientación** - Se asignan una o más orientaciones a cada punto de interés utilizando las direcciones de los gradientes locales. A partir de aquí, toda la información es relativa a la orientación, escala y localización de cada punto, con lo que se consigue invarianza frente a estas transformaciones.
4. **Generar el descriptor** - Para generar el descriptor se toma la información que proporcionan los gradientes locales. Se transforma toda esta información en una representación que sea robusta frente a cambios de apariencia de los objetos y de iluminación.

En este capítulo, se va a describir con más detalle los distintos pasos a seguir en la obtención de los descriptores SIFT y el proceso de correspondencia inicial entre la imagen y los patrones generados.

4.1. Detección de puntos de interés en el espacio de escala

El primer paso para obtener los descriptores SIFT consiste en identificar puntos de la imagen invariantes frente a cambios de escala. Para ello, se buscan puntos que presenten las mismas características en todas las escalas de observación utilizando **espacios de escala** [Witkin, 1983].

Koenderink [Koenderink, 1984] y Lindeberg [Lindeberg, 1994] demostraron que la única base para generar dichos espacios era la función Gaussiana. Por tanto, el espacio de escala de una imagen se obtiene convolucionando dicha imagen con una función Gaussiana de escala variable:

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y) \tag{4.1}$$

donde:

- $L(x, y, \sigma)$ $\rightarrow$ es el espacio de escala resultante.
- $I(x, y)$ $\rightarrow$ es la imagen de la cual se desea obtener su espacio de escala.
- $G(x, y, \sigma)$ $\rightarrow$ es la función Gaussiana, de escala variable, con la siguiente forma:

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

Para detectar los puntos de interés de forma eficiente, Lowe [Lowe, 1999] propuso utilizar los extremos locales del espacio de escala formado por diferencias de Gaussianas. En este caso, se toman dos Gaussianas con escalas consecutivas que difieren en una constante k, y su diferencia se convoluciona con la imagen:

$$\begin{aligned} D(x, y, \sigma) &= \big(G(x, y, k\sigma) - G(x, y, \sigma)\big) * I(x, y) \\ &= L(x, y, k\sigma) - L(x, y, \sigma) \end{aligned} \tag{4.2}$$

Existe una serie de razones por las cuales se utiliza este tipo de función:

- La Laplaciana de la Gaussiana normalizada en escala, $\sigma^2\nabla^2 G$, se puede aproximar a la función diferencias de Gaussianas.
- Lindeberg [Lindeberg, 1994] demostró que la normalización de la Laplaciana con el factor σ^2 era necesario para conseguir una invarianza real frente a la escala.
- Mikolajczyk [Mikolajczyk, 2002] demostró que los puntos característicos de la imagen más estables se correspondían con los máximos y mínimos de $\sigma^2\nabla^2 G$, comparado con otras funciones como el gradiente, el Hessiano o el detector Harris de esquinas.

La relación entre la función $D(x, y, \sigma)$ y $\sigma^2\nabla^2 G$ es la siguiente:

$$\sigma\nabla^2 G = \frac{\partial G}{\partial \sigma} \approx \frac{G(x, y, k\sigma) - G(x, y, \sigma)}{k\sigma - \sigma} \tag{4.3}$$

Por tanto, la función diferencia de Gaussianas se puede aproximar a la siguiente expresión:

$$G(x, y, k\sigma) - G(x, y, \sigma) \approx (k - 1)\sigma^2\nabla^2 G \tag{4.4}$$

Se puede ver que en la aproximación anterior está presente el factor de normalización σ^2 necesario para conseguir la invarianza frente a cambios de escala. El factor $(k-1)$ de la ecuación anterior aparece en todas las escalas, por lo que no influye a la hora de localizar los extremos de la función.

Una forma eficiente para obtener $D(x, y, \sigma)$ es la que se muestra en la figura 4.1. Las imágenes convolucionadas se agrupan en octavas (por ejemplo, cada octava se corresponde con el doble de la σ anterior) y el valor de k se selecciona de forma que haya un número s fijo de imágenes por octava. Por tanto, $k = 2^{1/s}$.

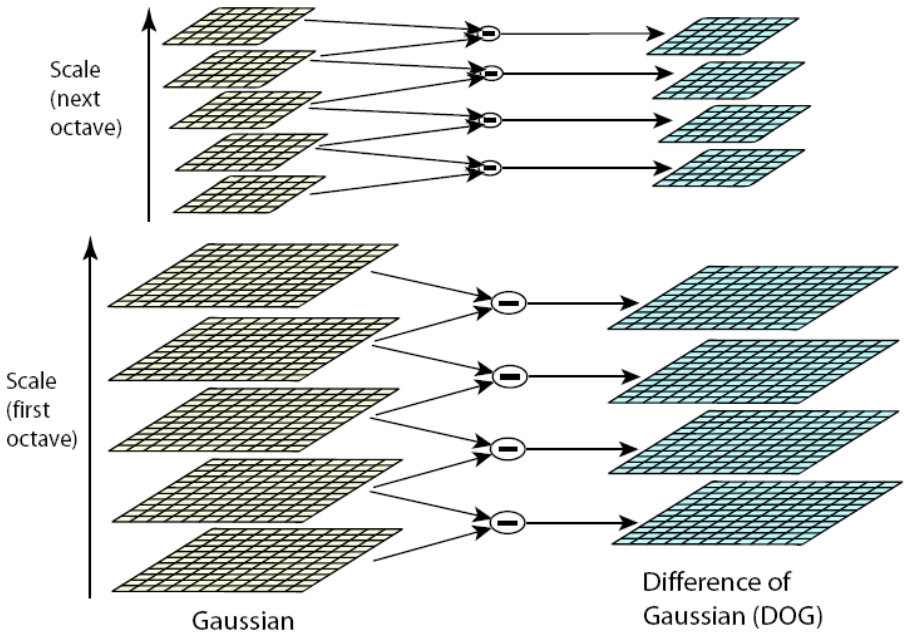


Figura 4.1: **Puntos de interés en el espacio de escala.** *En cada octava, la imagen se convoluciona con las diferentes Gaussianas. Las imágenes adyacentes se restan formando las imágenes correspondientes a la diferencia de Gaussianas. Despues de cada octava, la imagen se submuestrea por un factor igual a 2 y se repite el proceso*

4.1.1. Detección de extremos locales

Para detectar los máximos y mínimos locales de $D(x, y, \sigma)$, cada punto de la imagen se compara con sus ocho vecinos y con los nueve vecinos de las imágenes correspondientes a la escala superior e inferior, tal y como se muestra en la figura 4.2. El punto será seleccionado si su valor es mayor que los del resto de sus vecinos, o menor que el resto. Aunque a primera vista, la carga computacional puede parecer alta, la mayor parte de los puntos de las imágenes se descartan en las primeras comparaciones.

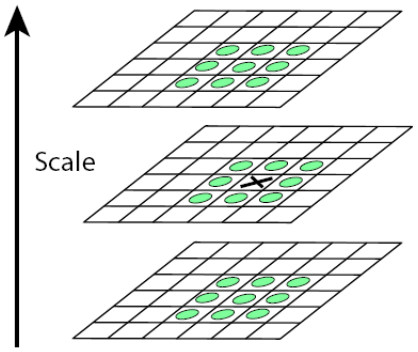


Figura 4.2: **Puntos de interés en el espacio de escala.** *Los máximos y mínimos de las diferencias de Gaussianas se obtienen comparando cada píxel (marcado con una X) con los 26 vecinos dentro de una región de dimensión* 3×3 *(marcados con un círculo).*

Una cuestión importante es determinar la frecuencia de muestreo en la imagen y en la escala que permiten detectar de forma fiable los extremos locales. Además, dichos extremos presentan una mayor inestabilidad ante pequeñas perturbaciones si se encuentran muy próximos entre ellos.

- **Frecuencia de muestreo en escala** - Lowe [Lowe, 2004] realizó una serie de experimentos sobre una colección de imágenes para evaluar el número óptimo de escalas por octava y así determinar la frecuencia que maximizaba la estabilidad de los extremos locales detectados.

  De los resultados obtenidos se observa que la mayor repetitividad a la hora de detectar los extremos se produce al utilizar 3 escalas distintas en cada octava. También concluye que los resultados obtenidos no mejoran al incrementar el número de muestras de escala por octava. Esto se debe a que a pesar de que el número de extremos locales es mayor, su estabilidad es menor y su detección empeora.

- **Frecuencia de muestreo en el dominio espacial** - Hay que determinar también el valor del suavizado inicial, σ. Tras realizar varios experimentos, se obtuvo que la repetitividad en la detección de extremos se incrementaba a medida que aumentaba σ, a la vez que disminuía la eficiencia. Por tanto, Lowe [Lowe, 2004] propone utilizar $\sigma = 1{,}6$, basándose en los resultados de dichos experimentos.

Al aplicar a la imagen un suavizado previo a la detección de extremos, se descartan las altas frecuencias. Para no perder información, la imagen se expande para generar un número de muestras mayor. Por tanto, mediante interpolación líneal, se duplica el tamaño de la imagen antes de empezar a construir el espacio de escala.

4.1.2. Localización exacta de los puntos de interés

Una vez que se detectan todos los extremos locales, hay que determinar qué candidatos serán validos como puntos de interés:

- Se obtiene la posición exacta de los extremos locales utilizando la información de los puntos de alrededor. Para ello, se utiliza el desarrollo de Taylor de segundo orden de la función $D(x, y, \sigma)$ evaluada en las cercanias del extremo local:

$$D(\mathbf{x}) = D + \frac{\partial D^T}{\partial \mathbf{x}}\mathbf{x} + \frac{1}{2}\mathbf{x^T}\frac{\partial^2 D}{\partial \mathbf{x}^2}\mathbf{x} \tag{4.5}$$

  donde D y sus derivadas se evaluan en el punto del extremo local siendo $\mathbf{x} = (x, y, \sigma)$ el offset. La posición exacta del máximo, $\hat{\mathbf{x}}$, se obtiene derivando la expresión anterior e igualandola a cero:

$$\hat{\mathbf{x}} = -\frac{\partial^2 D^{-1}}{\partial \mathbf{x}^2}\frac{\partial D}{\partial \mathbf{x}} \tag{4.6}$$

- Una vez que se ha obtenido la posición del extremo, se evalúa el valor del mismo, $D(\hat{\mathbf{x}})$. Si el extremo no supera un cierto umbral ($|D(\hat{\mathbf{x}})| < 0{,}03$) se descarta. Esto se debe a que los extremos con bajo contraste son más sensibles al ruido.

- Para obtener puntos de interés estables, no basta con eliminar los extremos con bajo contraste. La función diferencia de Gaussianas presenta una fuerte respuesta a lo largo de los bordes, incluso cuando el extremo está poco definido. En estos puntos, la curvatura principal es grande a lo largo del borde pero pequeña en la dirección perpendicular.

  Las curvaturas principales se obtienen de la matriz Hessiana, pues son proporcionales a los autovalores de ésta:

$$\mathbf{H} = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix} \tag{4.7}$$

Puesto que sólo interesa conocer la relación entre autovalores, se puede evitar el cálculo explícito de los mismos. Si β es el autovalor de menor magnitud y α el autovalor mayor, de forma que $\alpha = r\beta$, obtenemos:

$$Tr(\mathbf{H}) = D_{xx} + D_{yy} = \alpha + \beta \tag{4.8}$$

$$Det(\mathbf{H}) = D_{xx}D_{yy} - (D_{xy})^2 = \alpha\beta \tag{4.9}$$

$$\frac{Tr(\mathbf{H})^2}{Det(\mathbf{H})} = \frac{(\alpha+\beta)^2}{\alpha\beta} = \frac{(r\beta+\beta)^2}{r\beta^2} = \frac{(r+1)^2}{r} \tag{4.10}$$

La expresión 4.10 sólo depende de la relación entre autovalores, r. Por tanto, basta fijar un valor para r (Lowe [Lowe, 2004] utiliza $r = 10$) y se descartan todos los puntos que cumplan:

$$\frac{Tr(\mathbf{H})^2}{Det(\mathbf{H})} > \frac{(r+1)^2}{r} \tag{4.11}$$

4.1.3. Obtención de la orientación

Una vez que se han descartado los candidatos no válidos, se le asigna a cada punto de interés una orientación basada en las propiedades locales de la imagen en torno a ese punto.

La escala final de cada punto se utiliza para seleccionar la imagen suavizada con la función Gaussiana de escala más cercana, $L(x, y, \sigma)$. Para dicha imagen y a esa determinada escala, se calcula la magnitud y orientación del gradiente utilizando la diferencia de píxeles:

$$m(x,y) = \sqrt{\big(L(x+1,y) - L(x-1,y)\big)^2 + \big(L(x,y+1) - L(x,y-1)\big)^2} \tag{4.12}$$

$$\theta(x,y,) = \tan^{-1}\left(\frac{L(x,y+1) - L(x,y-1)}{L(x+1,y) - L(x-1,y)}\right) \tag{4.13}$$

siendo:

- $m(x,y) \rightarrow$ magnitud del gradiente.
- $\theta(x,y) \rightarrow$ orientación del gradiente.

A continuación, se calculan los gradientes de las muestras situadas en torno al punto de interés. En concreto, se toma una región de 16×16 muestras. Para conseguir invarianza a rotaciones, las coordenadas y orientaciones de cada gradiente se expresan respecto al punto característico y su orientación.

La magnitud también se pondera con una función Gaussiana centrada en el punto de interés y con una desviación típica igual a $1{,}5$ veces la escala de dicho punto. El propósito del uso de esta ventana Gaussiana es el de dar menos énfasis a los gradientes situados lejos del centro.

Todo este proceso se muestra figura 4.3.

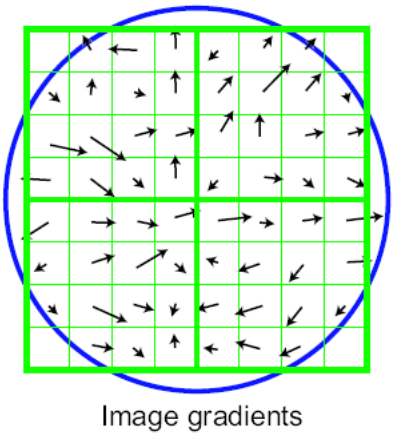
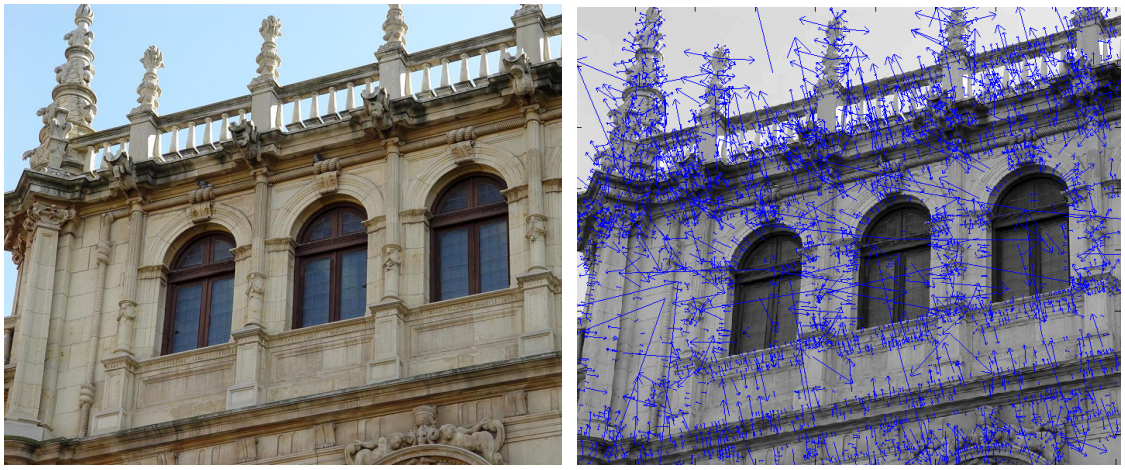


Figura 4.3: **Puntos de interés en el espacio de escala.** *Primero se calcula la orientación y magnitud del gradiente de cada muestra situada en torno al punto de interés. Además, se utiliza una ventana de tipo Gaussiana (indicada con el circulo) para ponderar la magnitud de los mismos.*

El siguiente paso es crear un histograma de orientaciones con todos estos gradientes ponderados. Los picos de este histograma se corresponden con las direcciones dominantes de los gradientes locales. Por tanto, para calcular la orientación final del punto de interés, se toma el mayor de los picos y todos los que estén en torno al 80 % de éste. En el caso en que haya múltiples picos de magnitud similar, se crean varios puntos de interés con la misma localización pero orientaciones diferentes.

En las siguientes imágenes, se muestran algunos ejemplos en los que se ha obtenido todos los puntos característicos de una imagen. Las flechas de color azul representan la orientación y magnitud del gradiente en cada punto. Se puede apreciar que el número de puntos de interés encontrados en la imagen es muy elevado.



(a) (b)

Figura 4.4: **Puntos de interés en el espacio de escala.** *(a) Imagen original de la que se quiere obtener los puntos de interés. El tamaño de la imagen es de* 900×750 *píxeles. (b) Resultado tras aplicar el algoritmo SIFT. Las flechas de color azul muestran la orientación y magnitud de los gradientes de cada punto de interés. En total se han detectado 4515 puntos*

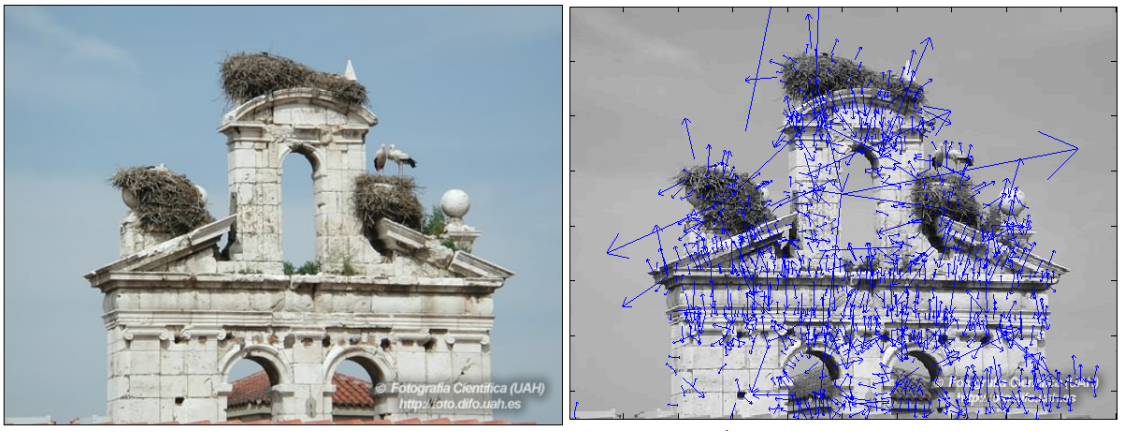


Figura 4.5: **Puntos de interés en el espacio de escala.** *(a) Imagen original de la que se quiere obtener los puntos de interés. El tamaño de la imagen es de* 500×375 *píxeles. (b) Resultado tras aplicar el algoritmo SIFT. Las flechas de color azul muestran la orientación y magnitud de los gradientes de cada punto de interés. En total se han detectado 1195 puntos*

4.2. Método SIFT

Hasta ahora, lo único que se ha hecho es obtener la localización exacta de cada punto de interés en la imagen , junto con su orientación y escala. Además, debido a la utilización de un sistema local de coordenadas para describir la región en torno a cada punto de interés, se consigue invarianza frente a rotaciones y traslaciones.

El siguiente paso es generar los descriptores que sean invariantes, en el mayor grado posible, al resto de transformaciones que pueden sufrir los objetos y las imágenes, como cambios de iluminación y perspectiva de los objetos. Además, es importante que los descriptores sean representativos para que, a la hora de comparar descriptores, se detecten claramente las correspondencias erroneas con una probabilidad elevada.

4.2.1. Descripción de los descriptores SIFT

Para calcular el descriptor de cada punto de interés, se parte del array de 16×16 muestras con sus correspondientes gradientes, tal y como se muestra en la figura 4.6.

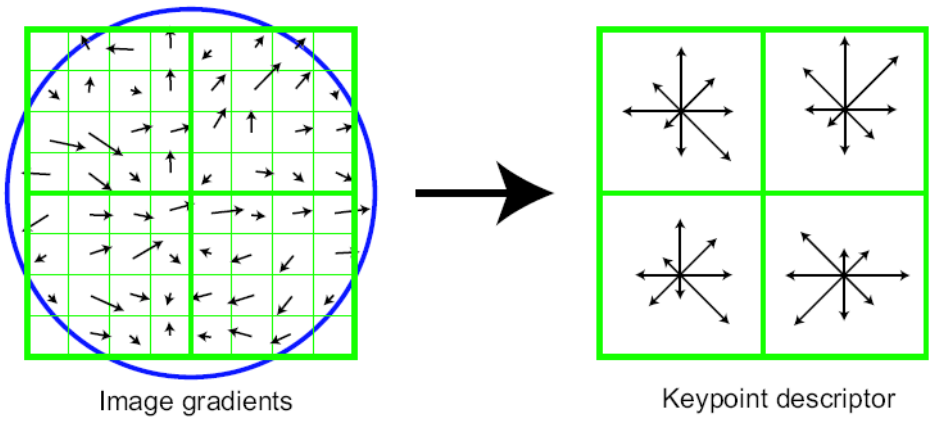


Figura 4.6: **Método SIFT.** *Se calculan los gradientes en una región de* 16×16 *muestras en torno al punto de interés, tal y como se muestra en la izquierda de la figura. A continuación, se divide esta región en zonas de dimensión* 4×4 *y se calculan los histogramas de orientación de cada zona. Cada histograma está formado por 8 elementos. Para simplificar la figura, sólo se ha representado una región de* 8×8 *muestras y por tanto, un array de histogramas de* 2×2

Se divide este array en regiones de 4×4 y se construye un histograma de orientación de cada región. Además, cada histograma estará formado por 8 elementos que se corresponden cada uno con una orientación distinta. Por tanto, al final se obtiene un array de 4×4 histogramas.

A continuación, se construye el descriptor. Éste se corresponde con un vector que contiene los valores de los histogramas de orientación. Puesto que cada punto de interés tiene asociado un array de 4×4 histogramas de 8 orientaciones, cada descriptor será un vector de $4 \times 4 \times 8 = 128$ elementos.

Cuando se produce un cambio en el contraste de la imagen, el valor de cada píxel se multiplica por una cierta constante, y por tanto, el gradiente también se ve afectado por dicha constante. Para eliminar los efectos que producen los cambios de iluminación en el descriptor, basta con normalizar el vector. Los cambios de brillo, sin embargo, no afectan al descriptor. Al modificarse el brillo, se suma a cada píxel una constante, pero esta constante no influye en el descriptor debido a que el gradiente se calcula mediante diferencias de píxeles.

De esta forma, se consigue invarianza frente a cambios de iluminación homogénea. Sin embargo, no pasa lo mismo para cambios no lineales, como ocurre en los casos de saturación de la camara o en cambios de iluminación que afecta de forma distinta a cada superficie de los objetos tridimensionales. Estos efectos pueden producir grandes cambios en las magnitudes de los gradientes, aunque no afectan tanto a las orientaciones de los mismos. Para reducir estos efectos y eliminar los gradientes de gran magnitud, cada vector normalizado se umbraliza de forma que cada elemento no sobrepase un cierto valor (fijado experimentalmente en $0{,}2$), normalizando de nuevo el vector. Además, a la hora de buscar correspondencias entre los vectores de características, se dará más importancia a la orientación de los mismos que a su magnitud.

4.2.2. Matching mediante descriptores de alta dimensionalidad

Hasta ahora, sólo se ha realizado una descripción de los descriptores SIFT y de su algoritmo de obtención. Como se ha comentado en la introducción del capítulo, para resolver el problema de detección de objetos, se van a utilizar modelos geométricos y de apariencia. Una vez obtenidos los modelos de apariencia, se buscan correspondencias entre los puntos de dichos modelos y los puntos de interés encontrados en la imagen a analizar.

Para realizar la correspondencia inicial, se compara cada descriptor de la imagen con todos los descriptores de los patrones almacenados en la base de datos. Se toma como posible candidato el descriptor más cercano en semejanza al de la imagen. Para evaluar esta semenjanza, se debería comparar los 128 elementos de cada descriptor entre sí y tomar como candidato aquel con mínima distancia Euclídea entre los elementos de ambos vectores. Sin embargo, este planteamiento es muy costoso pues los descriptores tienen alta dimensionalidad.

Para solucionar este problema y mejorar la eficiencia a la hora de comparar descriptores, se va a utilizar el producto escalar. A partir del producto escalar, se obtiene el ángulo que forman los descriptores (los descriptores están normalizados de forma que su modulo

es la unidad). De forma que, cuanto menor sea su ángulo, serán más semejantes.

Sin embargo, no vale únicamente con calcular para cada descriptor de la imagen el ángulo que forma con los descriptores de la base de datos y tomar el de menor ángulo pues, existirá un alto porcentaje de descriptores que no tendrán una correspondencia asociada. Estos descriptores se corresponden con puntos característicos que no fueron detectados en el proceso de entrenamiento previo o pertenecen al fondo de la escena. Por tanto, es útil buscar un método que permita descartar estos puntos de interés.

La utilización de un umbral global de distancia que permita determinar si una correspondencia es válida no funciona correctamente. Lowe [Lowe, 2004] propone un método más efectivo para conseguir un "matching" más fiable. Este consiste en evaluar la relación entre la distancia del primer vecino y la del segundo más cercano a un descriptor de la imagen. De esta forma, se considera que la correspondencia es válida si la relación entre distancias no supera un cierto umbral, es decir, si el segundo vecino está suficientemente alejado del primero. La justificación de este método es bastante intuitiva: si la correspondencia entre un descriptor de la imagen y otro de un patrón es correcta, ambos tendrán que ser muy similares y por tanto el ángulo que forman será muy pequeño comparado con el resto de descriptores del patrón. Sin embargo, para una falsa correspondencia del descriptor, la probabilidad de encontrar más descriptores con distancias similares es muy alta debido al alto número de descriptores que general el método SIFT.

Experimentalmente, Lowe propone usar un umbral comprendido entre $0{,}6$ y $0{,}8$, pues es dentro de ese rango donde el porcentaje de falsas correspondencias descartadas es mayor y el número de "inliers" también descartados es menor. En la figura 4.7 se muestra cómo varía el número de correspondencias iniciales que genera el método en función del umbral utilizado, y cuantos de estos se corresponden realmente con "inliers" y cuantos son falsas correspondencias.

A medida que disminuye el umbral, el sistema es más restrictivo pues la distancia entre el primer vecino y el segundo tiene que ser mayor para poder considerar que es una correspondencia correcta. En estos casos, el número final de "outliers" es muy bajo en comparación con el número resultante de "inliers". A medida que se aumenta el umbral, tanto el número de "inliers" como el de "outliers" aumenta y la tasa crecimiento de falsas correspondencias es mayor a partir de un umbral de $0{,}75$ en comparación con la de los "inliers" (sigue un crecimiento exponencial).

De todo esto, se concluye que el umbral óptimo de trabajo para nuestro caso concreto está en torno a $0{,}8$ y $0{,}85$, pues es el intervalo en el cual el número de "inliers" es mayor. Aunque para este rango de valores, el número de falsas correspondencias es significativo, esto no presenta problemas pues, como se verá en capítulos posteriores, se va a utilizar distintos algoritmos robustos que permiten descartar dichos "outliers" incluso en los casos de un alto porcentaje de ellos. Lo importante en este caso, es detectar inicialmente la mayor cantidad de "inliers".

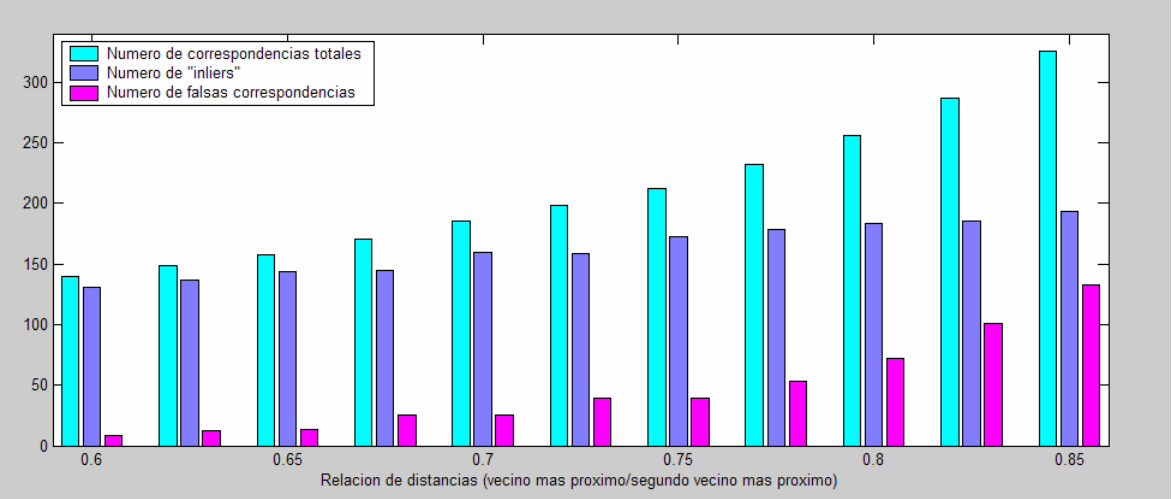


Figura 4.7: **Método SIFT.** *En esta gráfica se muestra la variación del totales en función del umbral aplicado. Así mismo, se muestra el número de puntos del total de las corrrespondencias que realmente son "inliers" y cuales se corresponden con falsas correspondencias.*

Al trabajar con descriptores de alta dimensionalidad, la utilización de otros métodos de búsqueda como el k-d tree [Friedman et al., 1977] no mejoran en eficiencia respecto al método exhaustivo anterior. Por tanto, Lowe también propone como mejora el uso del algoritmo
Best-Bin-First (BBF) [Beis and Lowe, 1997]. Sin embargo, los resultados que se obtienen con el método de búsqueda exhaustiva es bueno, por tanto, se va a utilizar dicho método.

Por último, hay que remarcar que el objetivo de este proyecto no es implementar el algoritmo SIFT, sino utilizar estos descriptores para una aplicación concreta, el reconocimiento de objetos mediante imágenes.
Para obtener los descriptores SIFT de las imágenes se va a utilizar un programa desarrollado por David Lowe y que se puede descargar desde su página personal:

`http://www.cs.ubc.ca/~lowe/keypoints/`

El programa esta compuesto por un archivo binario compilado que se puede ejecutar bajo Windows o bajo Linux, y una serie de archivos de Matlab que se encargan de realizar el "matching" inicial utilizando el método exhaustivo y la visualización de los descriptores y las correspondencias. Estos archivos de Matlab son los siguientes:

- `sift.m` - Este programa se encarga de obtener los puntos característicos de una imagen junto con sus descriptores. El programa devuelve las siguientes variables como parámetros de salida:

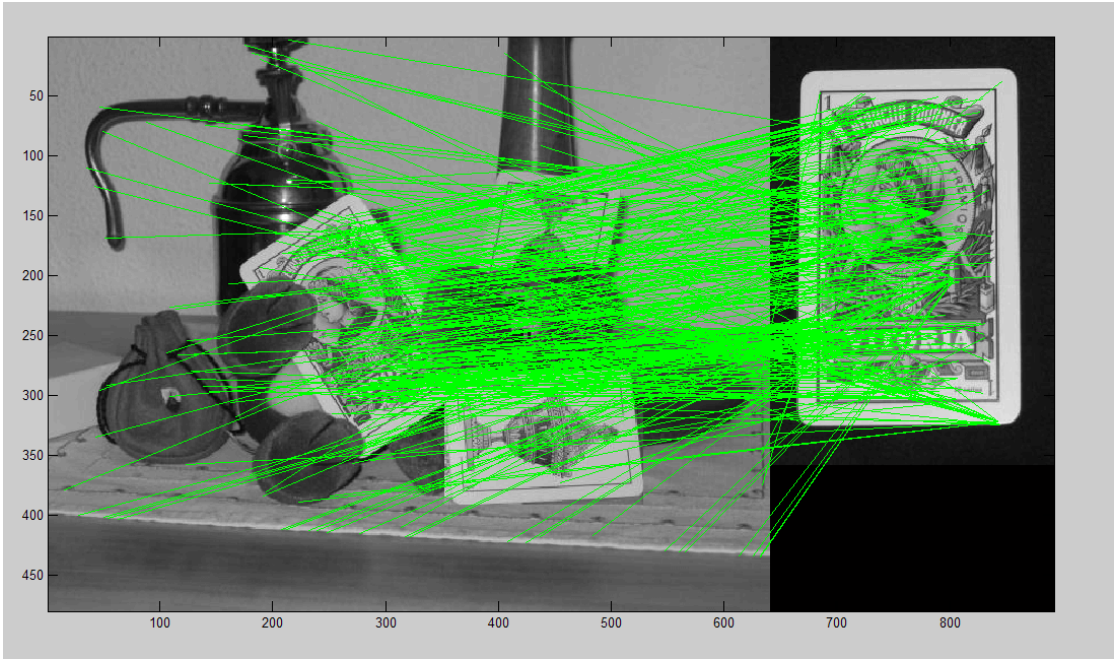
  - `image`: es el array de la imagen en formato double.
  - `descriptors`: es una matriz de dimensión $K \times 128$, donde cada fila se corresponde con el descriptor SIFT de uno de los K puntos característicos encontrados en `image`. Los descriptores están normalizados a 1.
  - `locs`: Matriz de dimensión $K \times 4$. Cada fila de la matriz contiene toda la información referente a la localización de cada unos de los K descriptores en la imagen: posición en el eje x de la imagen (filas), posición en el eje y (columnas), escala del descriptor y orientación del mismo.

- `match.m`: Este programa se encarga de realizar el emparejamiento de puntos entre una imagen cualquiera y la imagen del patrón, una vez que se han obtenido los descriptores de cada imagen. Esta función devuelve dos arrays con los indices de las correspondencias realizadas:

  - `matchPatron`: array donde se almacenan los índices que ocupan los descriptores emparejados del patrón en la matriz `descriptors` del patrón.
  - `matchImagen`: al igual que antes, se almacenan los indices correspondientes a los descripotre de la imagen. De esta forma, el descriptor que indexa `matchPatron(i)` está emparejado con el descriptor indexado por `matchImagen(i)`.

En las figuras 4.8 y 4.9 se muestran dos ejemplos del “matching” inicial utilizando un umbral de $0{,}85$.

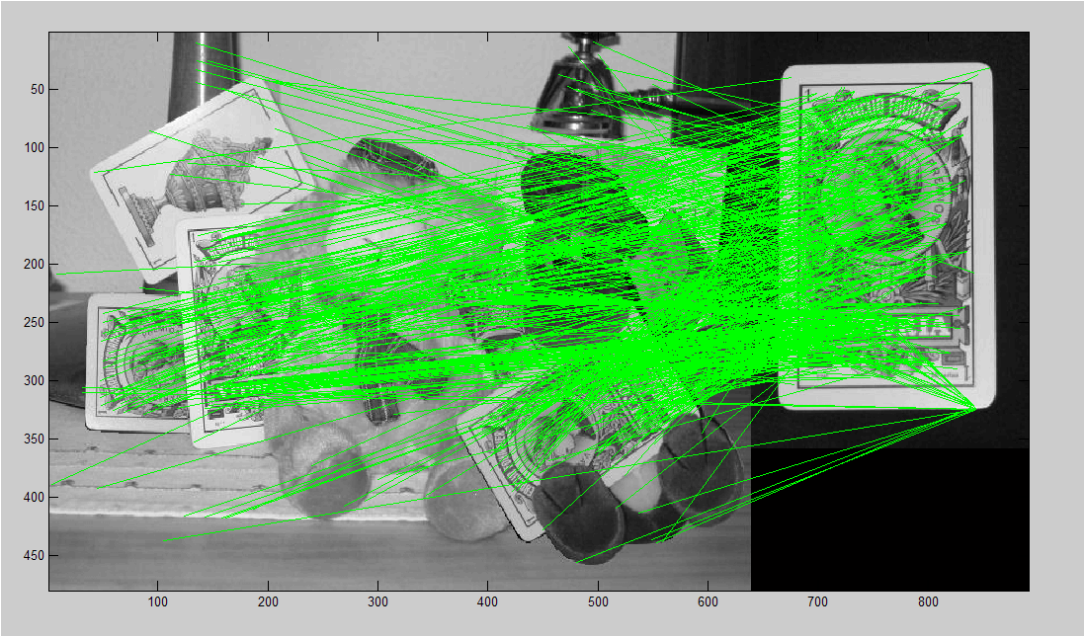
(a)



(b)

Figura 4.8: **Método SIFT.** *(a) Imagen original en la que se quiere buscar todas los naipes correspondientes al as de oros. (b) Resultado tras aplicar el "matching" inicial del método SIFT. Las líneas de color verde unen los puntos emparejados entre la imagen y el patrón. En total se han emparejado 326 puntos*

(a)



(b)

Figura 4.9: **Método SIFT.** *(a) Imagen original en la que se quiere buscar todas los naipes correspondientes al as de oros. (b) Resultado tras aplicar el "matching" inicial del método SIFT. Las líneas de color verde unen los puntos emparejados entre la imagen y el patrón. En total se han emparejado 524 puntos*

Una de las desventajas que tiene la utilización del programa desarrollado es que el tiempo de ejecución necesario para obtener los descriptores es muy alto, lo que hace que sea imposible aplicar el sistema completo de detección en tiempo real. En la gráfica 4.10, se muestra el tiempo de ejecución en función del número final de descriptores encontrados utilizando imágenes de 640×480. Si el tamaño de las imágenes es mayor, el tiempo crece exponencialmente.

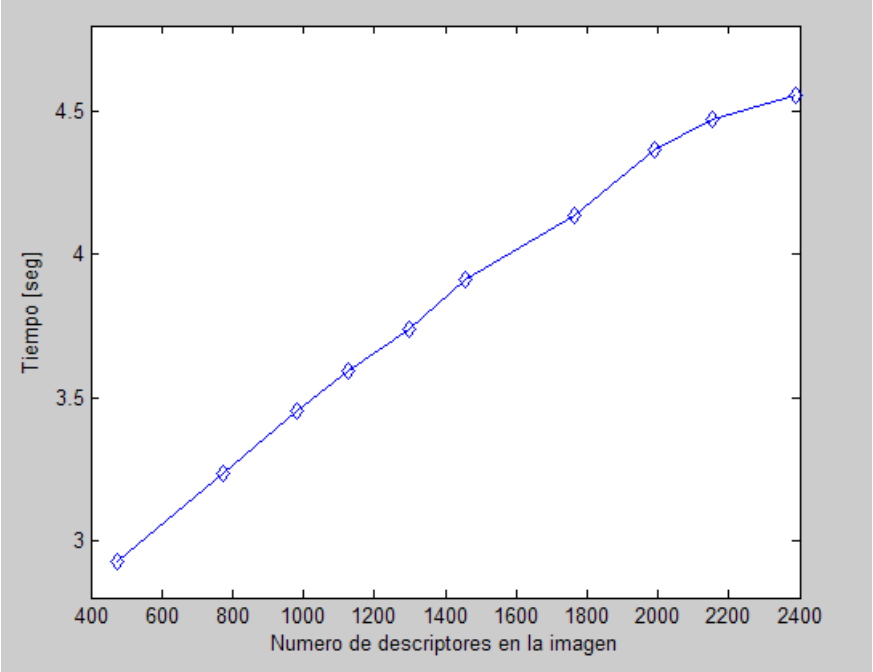


Figura 4.10: **Método SIFT.** *Esta gráfica muestra el tiempo de ejecución del programa en función del número de descriptores que encuentra en la imagen. Se han utilizado imágenes de* 640×480.

Capítulo 5

Estimación mediante algoritmos robustos

El método SIFT descrito en el capítulo anterior, permite obtener una correspondencia entre puntos de una imagen y de un patrón, con alta probabilidad de que cada emparejamiento sea correcto. Sin embargo, como se pudo ver en los ejemplos de las figuras 4.8 y 4.9, un cierto porcentaje de estas correspondencias serán "outliers", pudiendo sobrepasar incluso el 50 %. Por tanto, es necesario utilizar algún método que permita separar los "inliers" de los "outliers", incluso en los casos en los que el número de falsas correspondencias sea muy alto.

Por otro lado, este "matching" inicial sólo da información de la localización en las imagenes de los puntos emparejados. A partir de esta información, tenemos que ser capaces de determinar si hay en la imagen más de una repetición del mismo objeto, la localización y orientación de cada uno de ellos y agrupar los "inliers" que pertenezcan a cada una de las repeticiones.

Para conseguir todo esto, se propone utilizar los siguientes métodos de estimación robusta:

- **Transformada de Hough.**
- **Método RANSAC (Random Sample Consensous).**

En este capítulo, se va a realizar una breve descripción general de cada método y en capítulo posteriores, se detallará cómo se aplican estos dos métodos a nuestro problema concreto y se hará una comparación entre ambos.

5.1. Transformada Hough

La **Transformada de Hough** es una técnica que permite localizar en las imágenes conjuntos de puntos con características concretas. Este método fue desarrollado por Hough en 1962 [Hough, 1962] y patentada por IBM, aunque la versión de la transformada que se utiliza habitualmente fue desarrollada en 1971 por Peter Hart y Richard Duda [Duda and Hart, 1971].

La Transformada de Hough permite detectar formas geométricas o conjuntos de puntos con características concretas utilizando una definición parametrizada de las mismas. Básicamente, el método consiste en definir una correspondencia entre los puntos de la imagen y el espacio de parámetros. Inicialmente, la Transformada de Hough se utilizaba para identificar formas geométricas básicas como líneas, círculos y elipses. Sin embargo, esta transformada se puede utilizar para identificar modelos con características concretas, siempre que se conozca una definición paramétrica que relacione los puntos del modelo entre ellos.

Posteriormente, su uso se ha extendido a la detección de patrones arbitrarios, cuya geometría carece de una definición paramétrica directa. En este caso, la transformada recibe el nombre de **Transformada de Hough Generalizada** [Ballard and Brown, 1982].

Para entender el funcionamiento de la Transformada de Hough, primero se va a describir el proceso de detección de líneas rectas debido a su simplicidad, aunque, como ya se verá en el Capítulo **??**, en este proyecto se va a utilizar la Transformada de Hough para estimar la homografía entre los patrones y las imágenes.

5.1.1. Detección de rectas mediante la Transformada de Hough

Se desea detectar las rectas que mejor se ajusten a un conjunto de puntos existente en una imagen. Se asume que mediante alguna operación de procesado de la imagen (por ejemplo, mediante técnicas de detección de bordes), se ha obtenido un conjunto de N puntos con alta probabilidad de pertenecer a alguna recta. El proceso es el siguiente:

- Se busca un espacio alternativo para definir las rectas, de forma que cada recta se represente como un punto en dicho espacio.
- Se calculan todas las posibles líneas rectas que pasan por cada candidato. Cada punto "vota" por una serie de rectas a las cuales puede pertenecer.
- Aquellos puntos que en el espacio alternativo reciban un número elevado de "votos" se corresponden con líneas en la imagen.

La ecuación que define cualquier recta contenida en un plano es de la siguiente forma:

$$y = ax + b$$

donde a es la pendiente de la recta y b la ordenada en el origen.

Dado un punto del espacio $\mathbf{x_i}$, existen infinitas rectas que pasan por él y todas ellas cumplen la condición $y_i = ax_i + b$ para distintos valores de los parámetros a y b, tal y como se muestra en la figura 5.1.

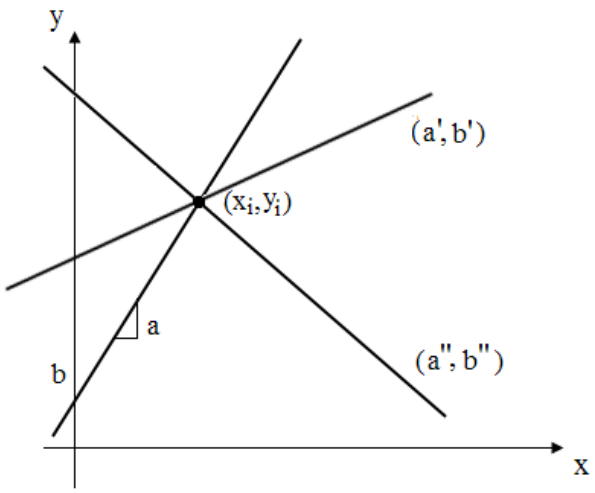


Figura 5.1: **Transformada de Hough.** *Conjunto infinito de líneas que pasan por un mismo punto de la imagen*

Si fijamos el punto (x_i, y_i), la ecuación de la recta se puede expresar en función de las coordenadas de dicho punto y de uno de los parámetros:

$$b = -ax_i + y_i$$

obteniéndose así un nuevo espacio en función de los parámetros a y b, denominado **espacio de parámetros**.

Por cada punto (x_i, y_i) que pasa por una recta en el plano imagen, se obtiene una recta r_i en el espacio de parámetros:

$$y = ax + b \quad \rightarrow \quad \begin{cases} (x_1, y_1) & \rightarrow & r_1 : b = -ax_1 + y_1 \\ (x_2, y_2) & \rightarrow & r_2 : b = -ax_2 + y_2 \\ (x_3, y_3) & \rightarrow & r_3 : b = -ax_3 + y_3 \\ \quad \ldots & & \\ (x_n, y_n) & \rightarrow & r_n : b = -ax_n + y_n \end{cases}$$

Tal y como se muestra en la figura 5.2, todos los candidatos que pertenezcan a la misma recta $y = a'x + b'$ generarán rectas en el espacio de parámetros que se cortan en un mismo punto de coordenadas, (a', b').

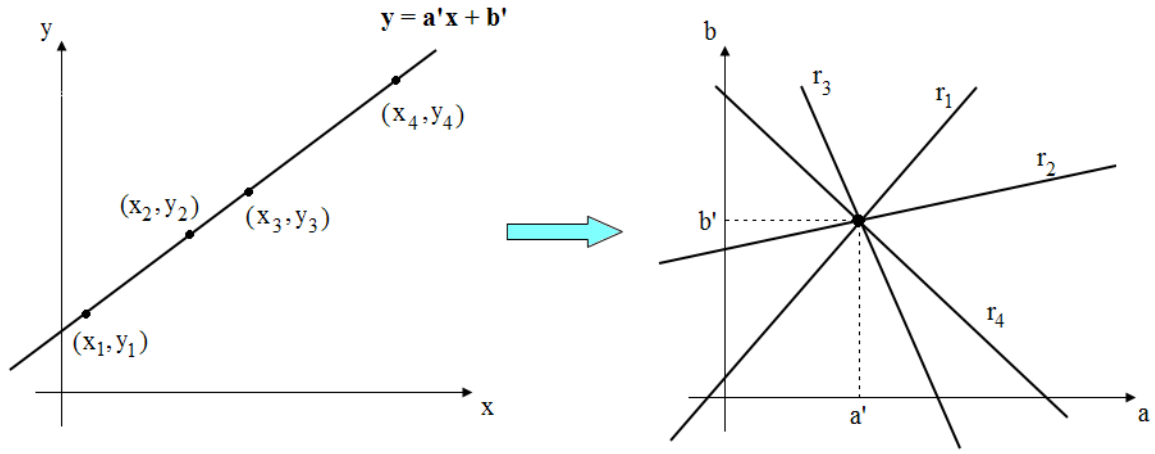


Figura 5.2: **Transformada de Hough.** *Los puntos que pertenecen a la misma recta generan rectas en el espacio de parámetros que se cortan en un mismo punto* (a', b').

La relación entre los puntos de la imagen y el espacio de parámetros se resume en el siguiente algoritmo:

Algoritmo 1 Detección de rectas utilizando la Transformada de Hough

1: Subdividir y cuantificar los parámetros a y b en intervalos dentro del rango de variación permitido: (a_{min}, a_{max}) y (b_{min}, b_{max}).

2: Crear un array de acumulación $H(a, b)$. Inicialmente, todas las celdas del array se ponen a cero. Este acumulador será un histograma bidimensional.

3: Para cada candidato de la imagen, (x_i, y_i), se obtienen los valores discretos de (a, b), fijando distintos valores para uno de los parámetros y calculando el segundo parámetro en función de estos valores y las coordenadas del punto:

$$b = -ax_i + y_i \quad \rightarrow \quad \begin{cases} a = a_1 & \rightarrow b_1 = -a_1 x_i + y_i \\ a = a_2 & \rightarrow b_2 = -a_2 x_i + y_i \\ \quad \ldots & \\ a = a_m & \rightarrow b_m = -a_m x_i + y_i \end{cases} \quad , \forall a \in (a_{min}, a_{max})$$

4: Incrementar todas las celdas de $H(a, b)$ que resulten de la ecuación anterior.

$$H(a_j, b_j) = H(a_j, b_j) + 1$$

5: Cada máximo local del acumulador nos informan de la presencia de una recta en la imagen. Si tenemos un máximo en $H(a_j, b_j)$, los parámetros de la recta en el plano imagen serán (a_j, b_j). El valor del máximo aporta información sobre el número de puntos pertenecientes a la misma recta.

Uno de los problemas que presenta este método al utilizar la representación cartesiana de la recta es que tanto la pendiente de la recta como la ordena en el origen no son parámetros acotados (a medida que las rectas se acercan a posiciones verticales, ambos parámetros tienden al infinito).

Peter Hart y Richard Duda [Duda and Hart, 1971] propusieron una parametrización alternativa que eliminaba este problema. Para ello se utiliza la representación polar de una recta:

$$\rho = x\cos\theta + y\sin\theta$$

donde θ es el ángulo que forma la normal a la recta con el eje de abscisas y ρ es la distancia desde el origen, tal y como se muestra en la figura 5.3. A θ y ρ se los denomina **parámetros normales** de la recta.

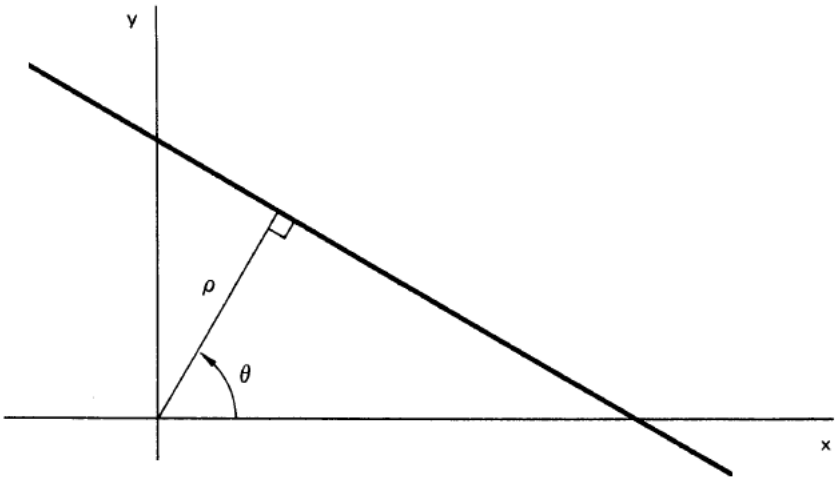


Figura 5.3: **Transformada de Hough.** *Representación polar de una línea recta.*

Puesto que las funciones seno y coseno son periódicas, los valores que puede tomar el parámetro θ se restringe al intervalo $[0, \pi)$, de esta forma, los parámetros normales de la recta serán únicos:

$$\rho = x\cos\theta + y\sin\theta \qquad \text{donde:} \quad 0 \leq \rho \leq (x_{max}^2 + y_{max}^2)^{-1/2}$$

$$0 \leq \theta \leq \pi$$

Con esta restricción, cualquier línea del plano x-y se corresponde con un único punto en el plano θ-ρ. La forma de construir el acumulador en el plano θ-ρ es similar al Algoritmo 1. La única diferencia es que, esta vez, cada punto (x_i, y_i) en la imagen genera curvas sinusoidales en el espacio de parámetros, tal y como se muestra en la figura 5.4. Por tanto, todos los puntos (x_i, y_i) que pertenezcan a una recta de la forma $\rho' = x\cos\theta' + y\sin\theta'$, generarán curvas que se cortan en un mismo punto del espacio de parámetros, (θ', ρ').

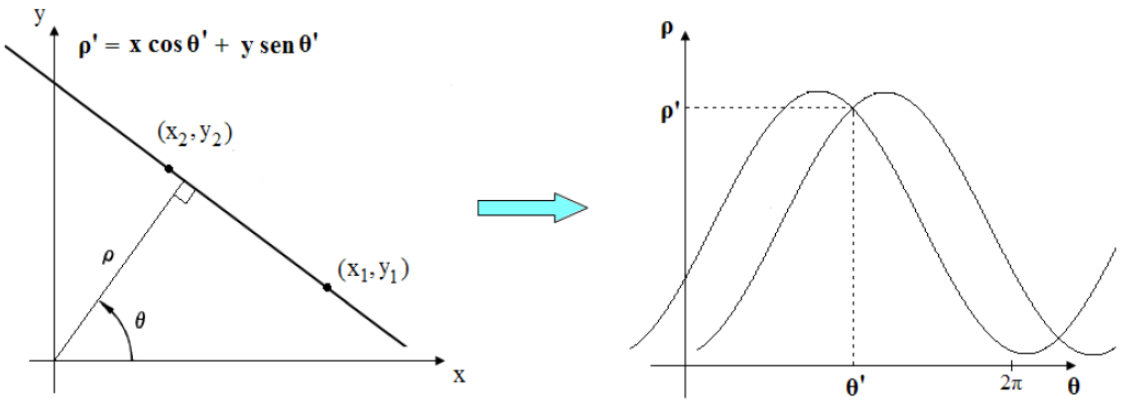


Figura 5.4: **Transformada de Hough.** *Los puntos que pertenecen a la misma recta generan curvas sinusoidales en el espacio de parámetros que se cortan en un mismo punto* (θ', ρ').

En la figura 5.6 puede ver un ejemplo práctico de la detección de rectas utilizando la Transformada de Hough:

Como se puede observar en el ejemplo anterior, con la Transformada de Hough somos capaces de detectar varias rectas en la misma imagen. Esta es una ventaja que ofrece este método: no es necesario conocer de antemano el número de clases presentes en la imagen.

De la misma forma que se aplica la Transformada de Hough para detectar rectas en las imágenes, se puede aplicar a cualquier figura geométrica de la que se conozca su parametrización. Es más, esta transformada se puede utilizar para detectar modelos de puntos con características concretas. Basta con conocer la relación entre dichos puntos mediante una definición paramétrica. Las fases para detectar un modelo general utilizando la Transformada de Hough es la siguiente:

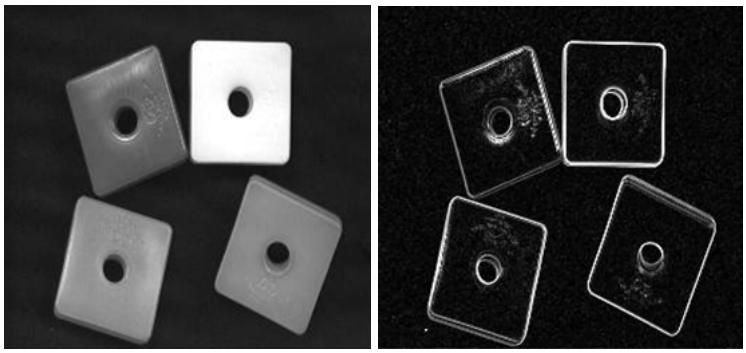
- Se desea detectar un modelo del cual se conoce la definición paramétrica que relaciona los puntos que pertenecen a dicho modelo:

  $$\mathcal{S} = f(\mathbf{\Phi})$$

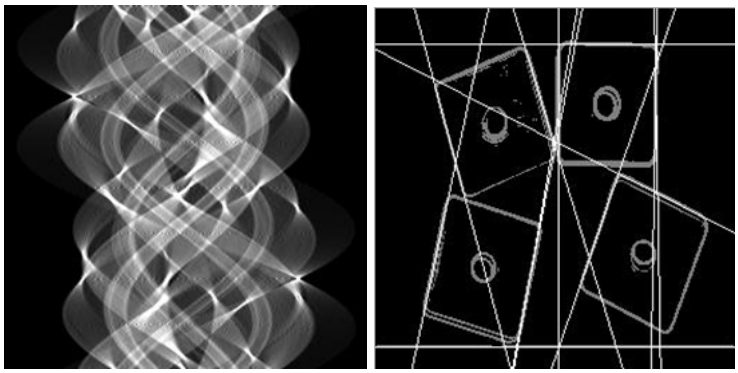
  siendo ϕ el vector de párametros:

  $$\mathbf{\Phi} = (\phi_1, \phi_2, \phi_3, \ldots, \phi_m)$$

- En la imagen se obtiene un conjunto de N puntos con alta probabilidad de pertenecer a dicho modelo.



(a) (b)



(c) (d)

Figura 5.5: **Transformada de Hough.** *(a) Imagen original. (b) Imagen binarizada de los bordes. (c) Array de acumulación. (d) Resultado de la detección de rectas.*

- A partir de la definición paramétrica del modelo, se busca una expresión que permita calcular el valor de uno de los parámetros en función del resto de parámetros y de las coordenadas de los puntos:

$$\phi_i = f(x, y, \phi_1, \ldots, \phi_{i-1}, \phi_{i+1}, \ldots, \phi_m)$$

- A partir de aquí, el algoritmo que se aplica es similar al Algortimo 1:

Algoritmo 2 Detección de un modelo general utilizando la Transformada de Hough

1: Se cuantifica y subdivide el espacio de parámetros $\mathbf{\Phi} = (\phi_1, \phi_2, \ldots, \phi_m)$ en intervalos dentro del rango de variación permitido: $\phi_i \in (\phi_{i_{min}}, \phi_{i_{max}})$ para $i = 1, \ldots, m$.

2: Se crea el array de acumulación $H(\phi_1, \ldots, \phi_m)$. Inicialmente, todas las celdas del array se ponen a cero.

3: Para cada candidato de la imagen, (x_j, y_j), se obtienen los valores discretos de $(\phi_1, \ldots, \phi_m)$, calculando el valor de uno de los parámetros en función de todas las combinaciones de valores que pueden tomar el resto de parámetros y de las coordenadas del punto:

$$\phi_i = f(x, y, \phi_1, \ldots, \phi_m) \rightarrow \begin{cases} \phi_i = \phi_{i_1} \rightarrow \phi_{i_1} = f(x_j, y_j, \phi_{1_1}, \ldots, \phi_{m_1}) \\ \phi_i = \phi_{i_2} \rightarrow \phi_{i_2} = f(x_j, y_j, \phi_{1_2}, \ldots, \phi_{m_2}) \\ \quad \ldots \\ \phi_i = \phi_{i_k} \rightarrow \phi_{i_k} = f(x_j, y_j, \phi_{1_k}, \ldots, \phi_{m_k}) \end{cases}$$

$$\forall \phi_i \in (\phi_{i_{min}}, \phi_{i_{max}})$$

4: Incrementar todas las celdas de $H(\phi_1, \ldots, \phi_m)$ que resulten de la ecuación anterior.

$$H(\phi_{1_j}, \ldots, \phi_{m_j}) = H(\phi_{1_j}, \ldots, \phi_{m_j}) + 1$$

5: Cada máximo local del acumulador nos informan de la presencia de un modelo en la imagen. Si tenemos un máximo en $H(\phi_{1_j}, \ldots, \phi_{m_j})$, los parámetros del modelo en la imagen serán $(\phi_{1_j}, \ldots, \phi_{m_j})$. El valor del máximo da información sobre el número de puntos que pertenecen a ese modelo. Si además se lleva un registro de los puntos que "votan" a cada celda del acumulador, se puede determinar qué puntos pertenecen al modelo.

La Transformada de Hough es un método de estimación robusto y eficaz, puesto que permite, por un lado, agrupar cada punto candidato con su modelo correspondiente y descartar los que no pertenecen a ningún modelo, incluso en los casos de un alto porcentaje de "outliers", y por otro lado, permite detectar varios modelos sin la necesidad de conocer de antemano el número de estos que están presentes en la imagen. A pesar de todo esto, hay que tener en cuenta la complejidad de este método.

Hay que tener presente el incremento notable del número de operaciones y el crecimiento exponencial de las dimensiones del acumulador a medida que aumenta el número de parámetros, lo que hace que este método sea ineficiente para modelos con un alto número de parámetros.

También hay que tener en cuenta el rango de valores entre los que puede estar cada parámetro y la cuantificación de los mismos. Si el rango es amplio, el número de operaciones a realizar es mayor. Lo mismo ocurre al aumentar la resolución de los parámetros. Esto se debe a que este método se basa en calcular el valor de un parámetro en función del resto de parámetros, utilizando todas las combinaciones posibles de valores permitidos.

Además, el éxito de esta técnica depende también de la cuantificación de los parámetros. Si tenemos poca resolución, se obtendrán máximos muy pronunciados y fáciles de detectar en el acumulador. Sin embargo, perdemos precisión a la hora de estimar los valores de los parámetros pertenecientes a cada modelo detectado. Por el contrario, si aumentamos la resolución, los valores de los parámetros serán más fiables mientras que en el acumulador tendremos picos menos definidos.

El último paso de los Algoritmos 1 y 2 consiste en encontrar máximos locales en el acumulador. La búsqueda de un máximo absoluto no es válida en nuestro caso pues no necesariamente habrá un único modelo en la imagen, o incluso, es posible que no haya ninguno. Cada máximo local que se encuentre en el acumulador se corresponderá con un nuevo modelo detectado. Además, la busqueda de máximos locales debe ser una búsqueda controlada:

- En general, los máximos no serán picos abruptos, sino que se asemejarán a distribuciones Gaussianas con desviaciones más o menos pequeñas. Esto se debe a:
  - Presencia de ruido.
  - El rango total de valores que pueden tomar cada parámetro se subdivide en un número finito de intervalos. Por tanto, al calcular el valor de un parámetro en función del resto de parámetros, sólo tenemos en cuenta un número finito de combinaciones y además, el resultado tendrá que ser cuantificado.
- Varios candidatos pueden “votar” por una misma celda del acumulador, aunque esa combinación de parámetros no se corresponda con ningún modelo presente en la imagen, generando un máximo (máximos espurios, de baja amplitud y generalmente, con desviaciones grandes).

Por tanto, hay que buscar un método que descarte los máximos espurios, permita encontrar los máximos locales del acumulador y que a su vez, evalue la probabilidad de que dicho máximo se corresponda con un modelo en la imagen. En este proyecto, se propone utilizar el algoritmo **K-medias** con algunas modificaciones. A continuación se realiza una breve descripción del algoritmo K-medias. Las modificaciones realizadas se describirán en el Capítulo **??**.

5.1.2. Algoritmo K-medias

La agrupación de elementos es de suma importancia en los algoritmos de aprendizaje no supervisado. Lo algoritmos de **"clustering"** (agrupamiento) permiten clasificar o agrupar objetos en función de ciertos atributos. Por tanto, cada clase estará formada por elementos "similares" entre sí y que a su vez son distintos al resto de elementos que forman las otras clases. Una forma de medir esta "similitud" es minimizando la distancia de cada objeto al centroide de la clase a la que pertenece, de esta forma, se minimiza la varianza de cada clase. En la figura 5.6 se muestra un ejemplo gráfico con 4 clases distintas:

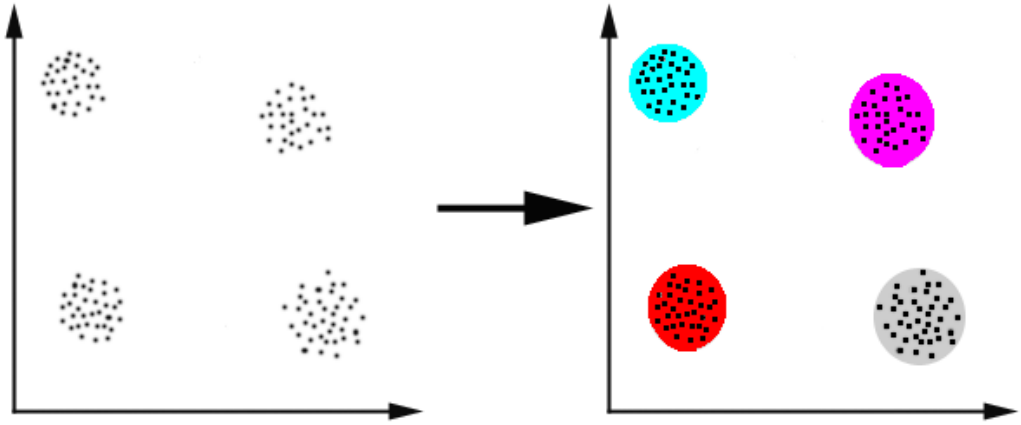


Figura 5.6: **Algoritmo K-medias.** *Ejemplo de clasificación de puntos en 4 clases distintas.*

El algoritmo **K-medias** es un algoritmo de "clustering" que permite agrupar un conjunto de N elementos en clases distintas no solapadas (cada elemento sólo puede pertenecer a una clase), conociéndose de antemano el número total de clases existentes, K. Para medir las distancias de los elementos a los centroides de cada clase se pueden usar varias alternativas. La más simple es utilizar la **distancia Euclídea**. Sin embargo, cuando los elementos estan ponderados y las clases se pueden aproximar a distribuciones Gaussianas (como será nuestro caso) se suele utilizar la **distancia de Mahalanobis**:

$$d(\mathbf{X_j}, \mathbf{C_i}) = (\mathbf{X_j} - \mathbf{C_i})^T \Sigma^{-1} (\mathbf{X_j} - \mathbf{C_i})$$

donde $\mathbf{C_i}$ y $\mathbf{X_j}$ son las coordenadas del centroide de la clase y las del punto respectivamente. Σ es la matriz de covarianza.

El proceso de clasificación utilizando el algoritmo K-medias es simple:

- Se comienza el algoritmo tomando K puntos aleatorios que se corresponderán con los centroides iniciales de las clases. Inicialmente, todas las clases serán equiprobables.
- El siguiente paso es calcular para cada elemento, las distancias a cada centroide, y asociarlo a la clase cuyo centroide sea el más cercano (el de menor distancia). En el caso de utilizar la distancia de Mahalanobis y suponer que las clases se asemejan a distribuciones Gaussianas, las distancias se calculan teniendo en cuenta la razón de verosimilitud entre las funciones de densidad de cada clase y als probabilidades a priori:
  - Cada clase C_i se corresponde con una distribución gaussiana N-dimensional, de media μ_i y matriz de covarianza Σ_i:

$$f(\mathbf{x}/C_i) = \frac{1}{(2\pi)^{N/2}(|\Sigma_i|)^{1/2}} \exp\Big[-\frac{1}{2}(\mathbf{x}-\mu_\mathbf{i})\Sigma_i^{-1}(\mathbf{x}-\mu_\mathbf{i})^T\Big]$$

  - El criterio que permite determinar si un punto pertenece a una clase u a otra es el siguiente:

$$\frac{f(\mathbf{x}/C_i)}{f(\mathbf{x}/C_j)} \underset{C_i}{\overset{C_j}{\lessgtr}} \frac{P_j}{P_i}$$

    donde P_i y P_j son las probabilidades a priori de cada clase. Desarrollando la expresión anterior elevada al cuadrado y tomando logaritmos se obtiene:

$$2{\cdot}\ln\left(\frac{P_i}{|\Sigma_i|^{1/2}}\right)-\Big[(\mathbf{x}-\mu_\mathbf{i})\Sigma_i^{-1}(\mathbf{x}-\mu_\mathbf{i})^T\Big] \underset{C_j}{\overset{C_i}{\lessgtr}} 2{\cdot}\ln\left(\frac{P_j}{|\Sigma_j|^{1/2}}\right)-\Big[(\mathbf{x}-\mu_\mathbf{j})\Sigma_j^{-1}(\mathbf{x}-\mu_\mathbf{j})^T\Big]$$

  - Por tanto, las distancias a los centroides se definen de la siguiente manera:

$$d_i = \Big[(\mathbf{x}-\mu_\mathbf{i})\Sigma_i^{-1}(\mathbf{x}-\mu_\mathbf{i})^T\Big] - 2\cdot\ln\left(\frac{P_i}{|\Sigma_i|^{1/2}}\right)$$

- Una vez que se han agrupado todos los elementos, se recalculan los K centroides con los puntos pertenecientes a cada clase. En el caso de utilizar la distancia de Mahalanobis, los nuevos centroides se corresponderán con la media de la distribución de cada clase. También se tiene que calcular las matrices de covarianza .
- Se vuelve a recalcular las distancias entre cada elemento y los nuevos centroides, reasignándolos de nuevo a la clase más cercana. El proceso se repite hasta que la variación entre los centroides recalculados y los centroides anteriores no supere un cierto umbral.

A continuación, se detalla el algoritmo completo usando la distancia de Mahalanobis:

Algoritmo 3 Clasificación de puntos utilizando K-medias con distancia de Mahalanobis

1: **Punto de partida:** Tenemos un conjunto de N elementos:

$$\mathcal{X} = \{\mathbf{X_i} \in \mathbb{R}^M | i = 1, \ldots, N\}$$

Cada elemento está ponderado por un valor normalizado: w_i

2: **Inicialización**:
Se define el número de clases distintas K, todas ellas equiprobables ($P_i = \frac{1}{K}$).
Se escogen los centroides: μ_j aleatorio para $j = 1, \ldots, K$.
Se inicializan las matrices de covarianza: $\Sigma_j = \mathbf{I_{M \times M}}$
Fijar umbral de variación máxima: $\epsilon_{distancia}$

3: **repeat**

4: Para cada punto se calcula las distancias de Mahalanobis a cada centroide:

$$d_j^i \;=\; d(\mathbf{C_i}, \mathbf{X_j}) = \left[(\mathbf{X_j} - \mu_\mathbf{i})\Sigma^{-1}(\mathbf{X_j} - \mu_\mathbf{i})^T\right] - 2\ln\left(\frac{P_i}{|\Sigma_i|^{1/2}}\right) \qquad \begin{matrix} i = 1, \ldots, K \\ j = 1, \ldots, N \end{matrix}$$

generando un array de distancias por cada centroide:

$$\begin{aligned} \mathcal{D}_{\mathbf{C_1},\mathcal{X}} &= (d_1^1, d_2^1, \ldots, d_N^1) \\ &\ldots \\ \mathcal{D}_{\mathbf{C_K},\mathcal{X}} &= (d_1^K, d_2^K, \ldots, d_N^K) \end{aligned}$$

5: Cada punto se asocia a la clase cuya distancia al centroide sea menor:

$$\begin{aligned} &\text{Elementos de } C_1 \quad \rightarrow \quad \mathcal{X}_1 = (X_1^1, \ldots, X_{N_1}^1) \\ &\qquad \ldots \\ &\text{Elementos de } C_K \quad \rightarrow \quad \mathcal{X}_K = (X_1^K, \ldots, X_{N_k}^K) \end{aligned}$$

6: Se recalculan los centriodes (medias), las matrices de covarianza y las probabilidades de cada clase C_i en función de los elementos que se han asociado a cada una.

$$\mu_{i_{nuevo}} \;=\; \mathrm{E}[\mathcal{X}_i] = \frac{1}{N}\sum_{j=1}^{N_i} w_j^i \mathbf{X}_j^i \qquad \text{donde:} \quad N = \sum_{j=1}^{N_i} w_j^i$$

$$\Sigma_i \;=\; \mathrm{E}[(\mathbf{X}_i - \mu_{i_{nuevo}})^T(\mathbf{X}_i - \mu_{i_{nuevo}})]$$

$$P_i = \sum_{j=1}^{N_i} w_j^i$$

7: **until** $d(\mu_{i_{nuevo}}, \mu_i) < \epsilon_{distancia} \quad \forall i$

A continuación se muestra un ejemplo del funcionamiento del algoritmo de K-medias. En la figura 5.7 se muestran 3 funciones Gaussianas con diferentes medias y varianzas. Cada una de estas funciones se corresponde con una clase de datos distintas. El objetivo es separar cada clase, obteniendo a su vez la varianza y media de cada una de ellas.

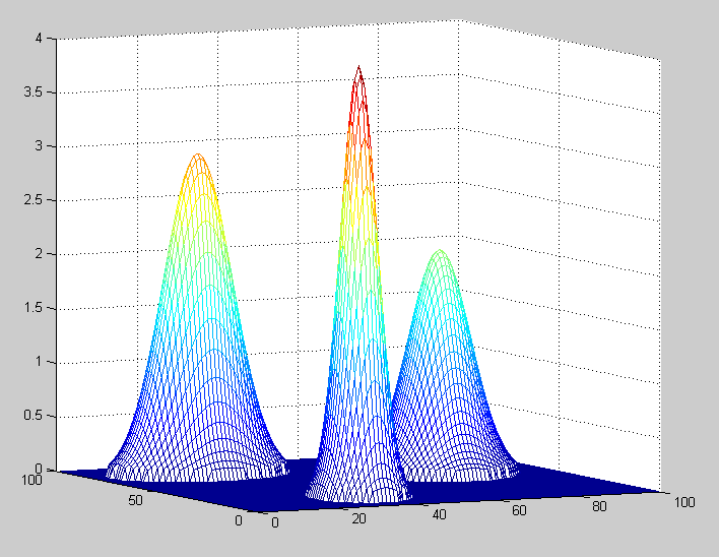


Figura 5.7: **Algoritmo K-medias.** *Ejemplo de clasificación de 3 clases distintas. Cada clase se corresponde con una de las funciones gaussianas.*

En las figuras 5.8, 5.9,5.10 y 5.11se muestran las iteraciones del algoritmo. Para cada iteración, se muestran dos figuras: en la figura de arriba se ha representado los 3 centros iniciales de cada iteración (de un color distinto cada uno), junto con los puntos asociados a cada uno tras calcular las distancias de Mahalanobis. En la de abajo, se muestra los nuevos centros (indicados con círculos) junto con los centros iniciales (con un rombo). Como se puede ver, en la cuarta iteración se alcanza una solución, y ésta es bastante satisfactoria. En la tabla 7.1.1 se muestra los valores de las centros y varianzas de cada clase junto con la solución obtenida tras aplicar el algoritmos K-medias:

	Datos Reales		Soluciones	
	Centro	Varianza	Centro	Varianza
Clase 1	$\mathbf{C_1} = (20, 35)$	$\Sigma_1 = \begin{bmatrix} 25 & 0 \\ 0 & 15 \end{bmatrix}$	$\tilde{\mathbf{C}}_\mathbf{1} = (20, 35)$	$\Sigma_1 = \begin{bmatrix} 22{,}6995 & 0 \\ 0 & 13{,}6514 \end{bmatrix}$
Clase 2	$\mathbf{C_2} = (80, 25)$	$\Sigma_2 = \begin{bmatrix} 35 & 0 \\ 0 & 65 \end{bmatrix}$	$\tilde{\mathbf{C}}_\mathbf{2} = (80, 25)$	$\Sigma_2 = \begin{bmatrix} 30{,}8963 & 0 \\ 0 & 57{,}1320 \end{bmatrix}$
Clase 3	$\mathbf{C_1} = (60, 75)$	$\Sigma_3 = \begin{bmatrix} 70 & 5 \\ 10 & 50 \end{bmatrix}$	$\tilde{\mathbf{C}}_\mathbf{3} = (60, 75)$	$\Sigma_3 = \begin{bmatrix} 59{,}4120 & 6{,}3021 \\ 6{,}3021 & 42{,}4352 \end{bmatrix}$

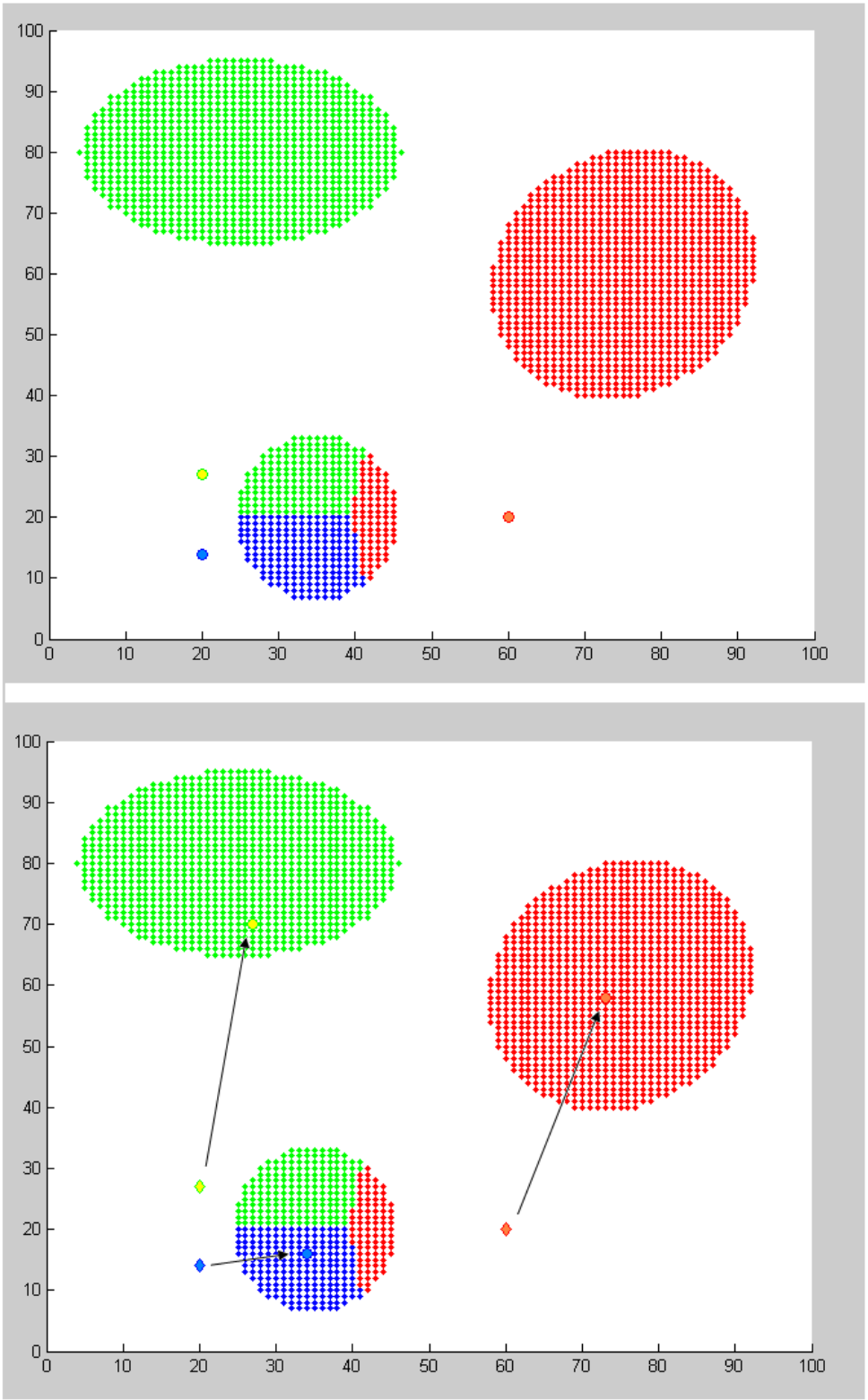


Figura 5.8: **Algoritmo K-medias.** *Primera iteración del algoritmo K-medias.*

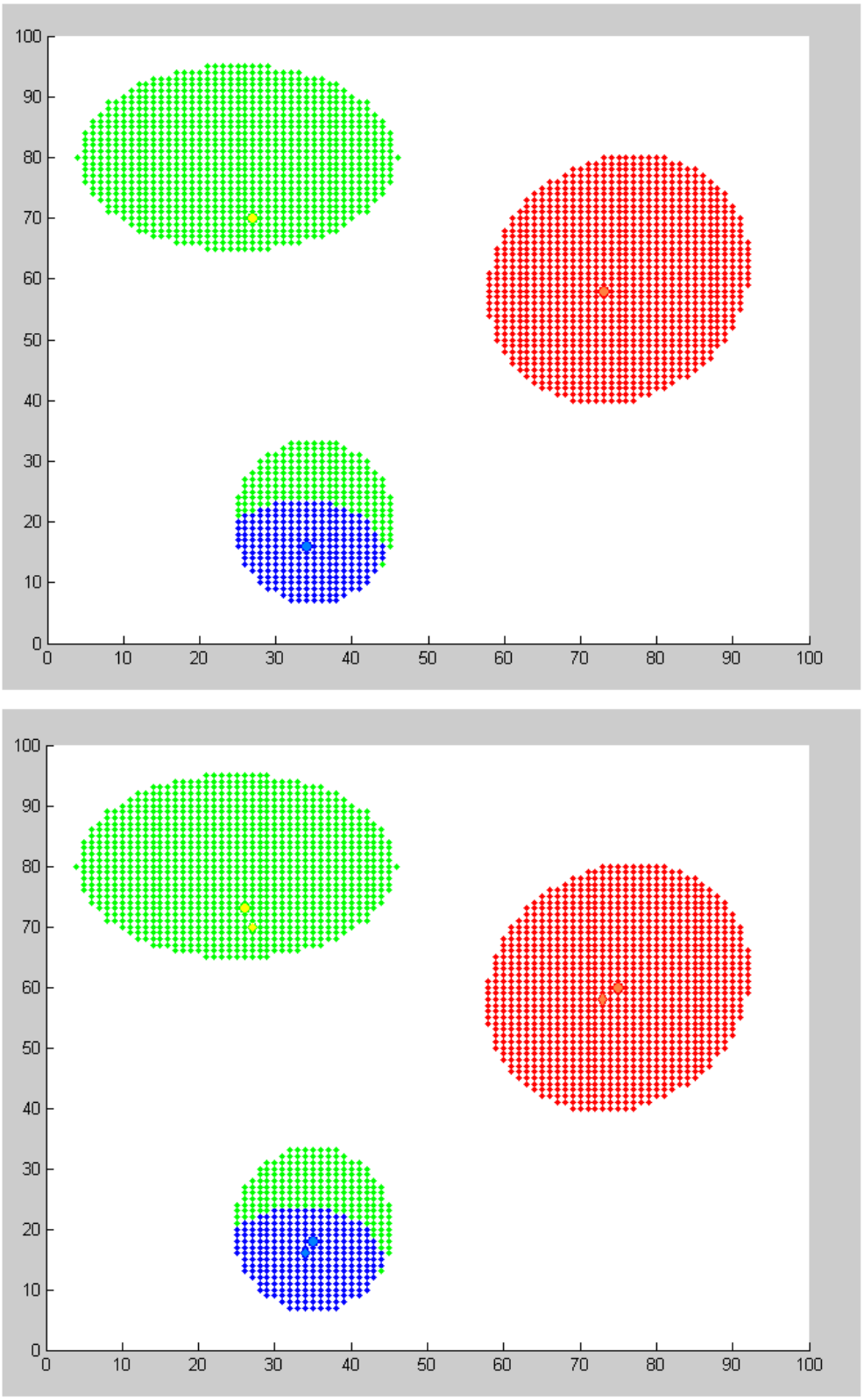


Figura 5.9: **Algoritmo K-medias.** *Segunda iteración del algoritmo K-medias.*

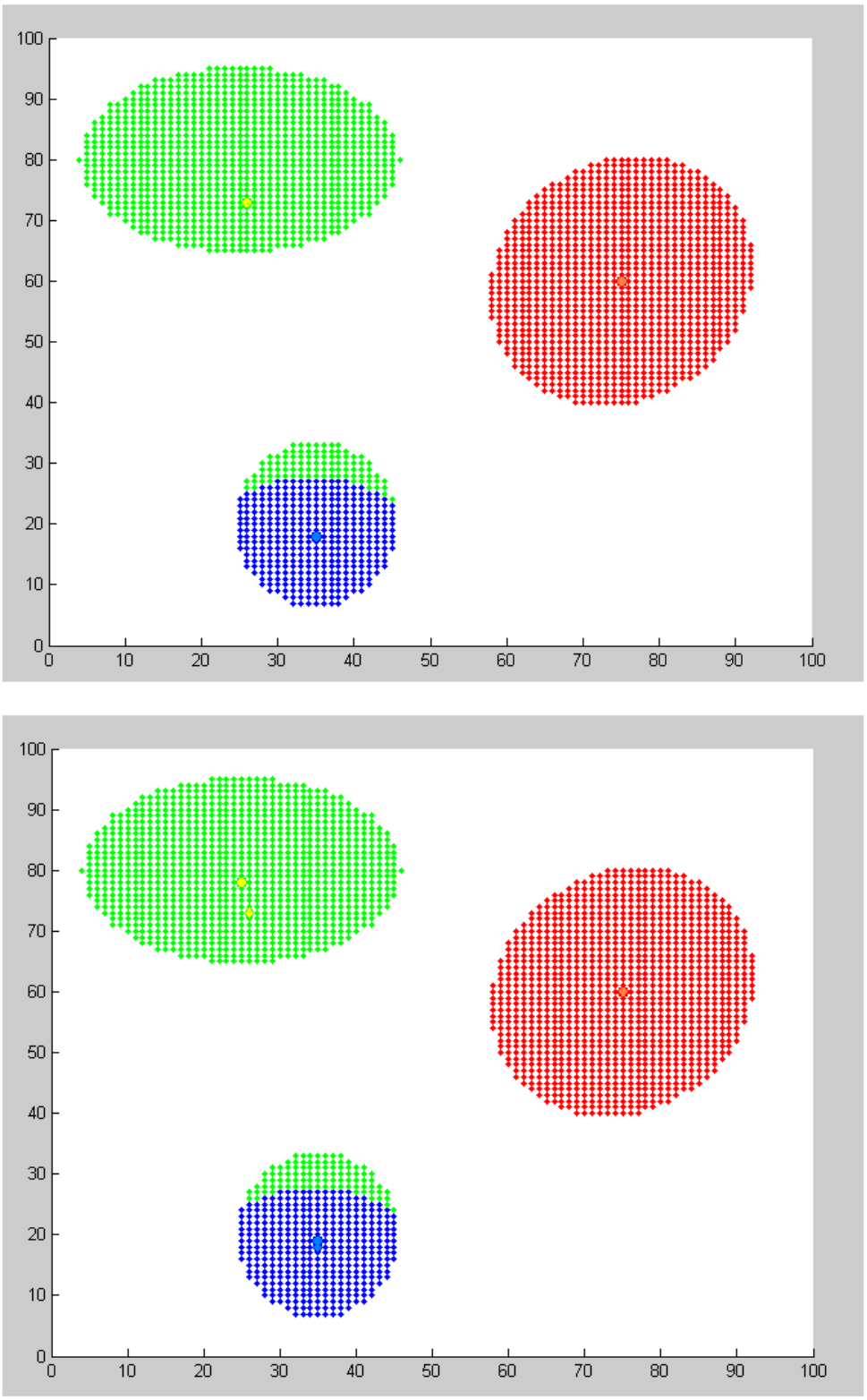


Figura 5.10: **Algoritmo K-medias.** *Tercera iteración del algoritmo K-medias.*

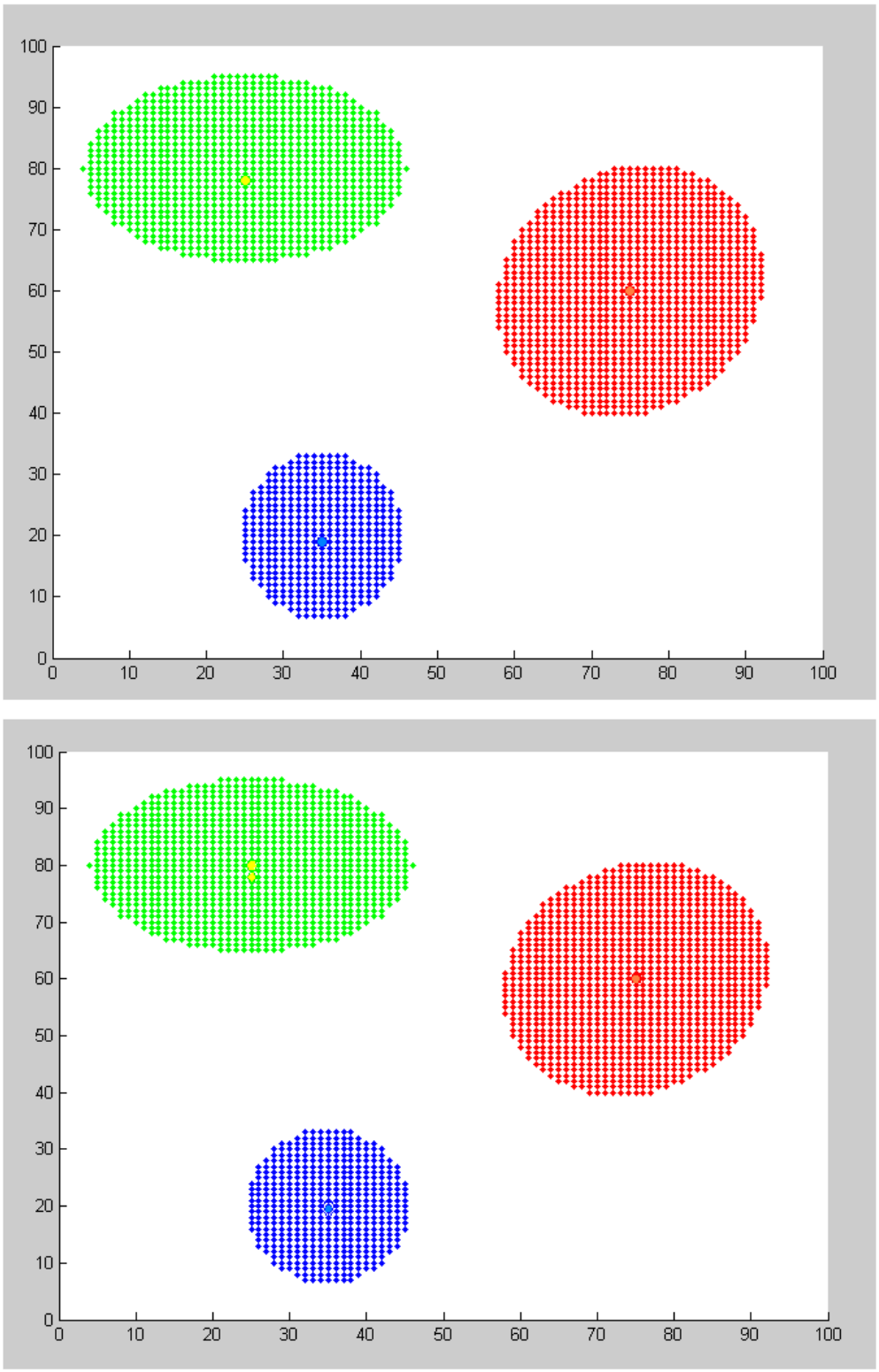


Figura 5.11: **Algoritmo K-medias.** *Cuarta iteración del algoritmo K-medias.*

5.2. Método RANSAC (Random Sample Consensous)

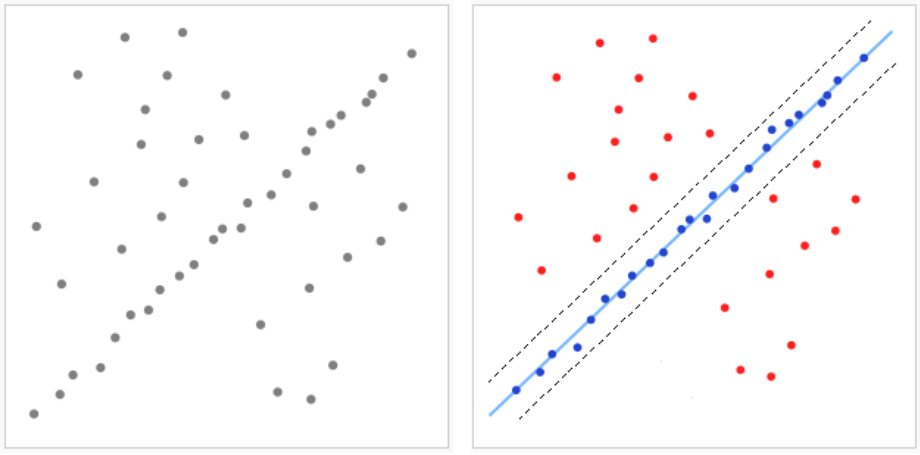
El método **RANSAC** (Random Sample Consensus) [Fischler and Bolles, 1981] [Hartley and Zisserman, 20 permite el ajuste robusto de un modelo a un conjunto de datos, incluso en los casos en que el porcentaje de "outliers" es elevado.

El procedimiento de este método es contrario al de los algoritmos de estimación tradicionales: en vez de usar la mayor cantidad de datos posibles para obtener una solución inicial y luego intentar eliminar los datos erróneos, RANSAC utiliza el menor conjunto de datos que permita calcular una solución inicial, aumentando dicho conjunto con datos consistentes que se adapten al modelo obtenido.

Para entender el algoritmo de RANSAC, se va a utilizar de nuevo el ejemplo sencillo de detección de una recta que mejor se adapte a un conjunto de puntos.

5.2.1. Detección de una recta utilizando RANSAC

El problema que se desea resolver es el que se muestra en la figura 5.12. Dado un conjunto de N puntos 2D, se quiere encontrar la recta que mejor se ajuste a los puntos eliminando los posibles "outliers". Para ello se impone la condición de que un punto será válido si su desviación respecto a la recta es menor de un umbral de distancia t.



(a) (b)

Figura 5.12: **Método RANSAC.** *(a) Conjunto de N puntos 2D. (b) Solución de la recta obtenida utilizando el método RANSAC.*

La ecuación de una recta cualquiera es de la forma:

$$y = ax + b$$

Al tener dos grados de libertad, basta con conocer dos puntos para definir la recta que pasa por ambos. El funcionamiento de RANSAC es simple:

1. Se escogen de forma aleatoria 2 puntos. Con estos dos puntos, queda definida la recta que los contiene.

2. Se determina el subconjunto de puntos que se adaptan a la recta anterior y cuya distancia a la misma no sobrepasa un cierto umbral t. A este conjunto de puntos se los denomina consenso y son los "inliers" de la recta. Además, se le asocia una puntuación a dicha recta en función del número de "inliers".

3. Este proceso se repite un número determinado de veces. La recta que haya recibido más puntuación es la solución final (intuitivamente, las rectas definidas por "outliers" recibirán puntuaciones bajas).

5.2.2. Parámetros del método RANSAC

Para utilizar el método de RANSAC, necesitamos definir una serie de parámetros:

- **Número de muestras** $\mathtt{s}$ - Hay que determinar el número de puntos mínimo que permiten definir el modelo. Por ejemplo, en el caso de la recta, $s = 2$.

- **Umbral de distancia** $\mathtt{t}$ - Este umbral permite determinar, con una probabilidad α, si un punto es un "inlier". Para calcular este umbral, es necesario conocer la distribución de probabilidad de la distancia de un "inlier" al modelo. Aunque en la práctica, lo normal es determinar este umbral de forma empírica puesto que en muchos casos se desconoce esta distribución.

- **Número de iteraciones** $\mathbf{N^{it}}$ - Lo ideal sería poder probar todas las combinaciones de puntos posibles para obtener el mejor modelo. Sin embargo, esta práctica, en la mayoría de los casos, es irrealizable e innecesaria. A pesar de ello, el número de iteraciones N^{it} debe ser lo suficientemente grande para garantizar que con una probabilidad p, al menos una de las combinaciones de puntos esté libre de "outliers".

  Supongamos que w es la probabilidad de que cualquier dato sea un "inlier", y por tanto, $\epsilon = 1 - w$ es la probabilidad de sea un "outlier". Entonces, hay que realizar al menos N^{it} iteraciones, donde $(1 - w^s)^{N^{it}} = 1 - p$. Despejando se obtiene:

  $$N^{it} = \frac{log(1-p)}{log(1-(1-\epsilon)^s)}$$

  Normalmente, se toma un valor de $p = 0{,}99$.

En muchos casos, la proporción de "outliers", ϵ, es desconocida y el número de iteraciones se calcula de forma adaptativa. En estos casos, el algoritmo se inicializa suponiendo el peor de los casos estimado para ϵ y su valor se va actualizando a medida que se encuentran conjuntos mayores de "inliers". La forma de realizar el cálculo adaptativo de N^{it} es el siguiente:

Algoritmo 4 Cálculo adaptativo de N^{it}

1: $N^{it} = \infty$, sample_count = 0.
2: **while** $N^{it} >$ sample_count **do**
3: Elegir un conjunto s de puntos y obtener los "inliers".
4: Calcular ϵ:
$$\epsilon = 1 - \frac{\text{número de "inliers"}}{\text{número total de puntos}}$$
5: Recalcular N^{it}:
$$N^{it} = \frac{log(1-p)}{log(1-(1-\epsilon)^s)}$$
6: sample_count = sample_count + 1
7: **end while**

- **Umbral de consenso T** - El algoritmo finaliza si el tamaño del conjunto del consenso (formado por los "inliers") es mayor que el número de "inliers" esperados para una cierta proporción de "outliers" ϵ y un conjunto de n elementos en total:

$$T = (1 - \epsilon)n$$

5.2.3. Algoritmo general de RANSAC

Como se ha visto en el apartado 5.2.1, el algoritmo de RANSAC se puede aplicar en la detección de líneas rectas. Es más, este algoritmo se puede utilizar para detectar cualquier modelo matemático junto con su vector de parámetros $\mathbf{\Phi}$, siempre que se conozca la definición paramétrica del mismo, $f(y, \mathbf{\Phi}) = 0$.

Se conocen un conjunto de M puntos:

$$Y = \{y_i | i = 1, \ldots, M\}$$

Además, existe una función:

$$\mathbf{\Phi} = g(Y^s)$$

que permite calcular los parámetros buscados a partir de Y^s, que es un subconjunto de elementos de Y. Este subconjunto está formado por s elementos que es número mínimo necesario para calcular los parámetros del modelo.

El algoritmo clásico de RANSAC consta de las siguientes fases:

Algoritmo 5 Algoritmo de RANSAC clásico

$N^{it} = \infty$, $N = 1$, $N_{best}^{inlier} = 0$, $p = 0{,}99$.

while $N < min(N^{it}, N_{max}^{it})$ **do**
  Se selecciona de forma aleatoria un subconjunto de s elementos de Y, Y^s.
  Se calcula $\mathbf{\Phi} = g(Y^s)$.
  Se obtiene el conjunto de N^{inlier} puntos de Y que son los "inliers" del modelo anterior. Todos los puntos cuya desviación al modelo obtenido este por debajo del umbral t serán "inliers".

$$Y^{inlier} = \{y_i| \, |f(y_i, \mathbf{Phi}) < t|\}$$

  if $N^{inlier} > N_{best}^{inlier}$ **then**
    $\mathbf{\Phi}_{best} = \mathbf{\Phi}, \quad \epsilon = 1 - (N^{inlier}/M)$

    $N^{it} = \frac{log(1-p)}{log(1-(1-\epsilon)^s)}$

  end if
  $N = N + 1$
end while

5.2.4. Algoritmo RANSAC para múltiples objetos

El Algoritmo 5 vale únicamente para el caso de un único modelo a encontrar. Sin embargo, en nuestro caso, tenemos un número no definido de modelos, y por tanto, hay que modificar el algoritmo anterior para que se pueda aplicar para el caso de múltiples objetos:

Algoritmo 6 Algoritmo de RANSAC para múltiples modelos

while $N_{best}^{inlier} > N_{min}^{inlier}$ **do**
  Aplicar el Algoritmo 5 sobre Y.
  $Y = \{Y - Y^{inlier}\} \qquad \bar{\mathbf{\Phi}} = [\bar{\mathbf{\Phi}}; \mathbf{\Phi}_{best}]$.
end while
En el vector $\bar{\mathbf{\Phi}}$ se obtiene el conjunto de modelos.

Capítulo 6

Detección de objetos planares a partir de una imagen patrón

El objetivo general de este proyecto es la detección de objetos. Partiendo de una imagen, se desea determinar si un objeto, del cual se ha generado un patrón con anterioridad (modelo de apariencia), está o no presente en la imagen.

Los descriptores SIFT proporcionan buenos resultados al trabajar con objetos planares, incluso aplicando a los mismos cambios de perspectiva de hasta 60^{o}. Por tanto, como primera toma de contacto con dichos descriptores, esta parte del proyecto se va a centrar en la detección de objetos planares. Posteriormente, en el capítulo **??**, se generalizará su uso para objetos tridimensionales.

Como se explicó en el capítulo 4, con el método SIFT se obtienen un conjunto de descriptores tanto en la imagen como en el patrón y se buscan correspondencias entre ellos. Aunque el método SIFT garantiza que estas correspondencias son correctas con una probabilidad bastante alta, la presencia de un gran número de “outliers” es inevitable. Por tanto, para poder detectar objetos en la imagen hay que descartar estas falsas correspondencias. Además, no sólo interesa conocer los “matching” correctos, sino ser capaces de agrupar dichos “inliers” en función del objeto al que pertenezcan en la imagen (para el caso de que haya más de un objeto igual en la imagen) y a su vez, determinar la transformación sufrida por cada uno.

En este capítulo se van a proponer varios métodos que permiten descartar “outliers” y obtener las matrices de transformación para cada objeto detectado.

6.1. Descripción general del problema

Primero se va a realizar una descripción general de de la situación y del problema que se desea resolver.

- **Estructura del conjunto de cámaras** - Se tiene un conjunto de M cámaras calibradas, repartidas en el espacio de trabajo, de la forma:

  $$\mathcal{C} = \{P_i | i = 1, \ldots, M\}$$

  descritas por su matriz de calibración $P_i \in \mathbb{R}^{3\times4}$.

  Se va a suponer que el modelo de cada cámara es un modelo "pin-hole" ideal sin distorsión. El conjunto de cámaras se encuentra referenciado a un origen común de coordenadas en el espacio O. Por sencillez, este proyecto se va centrar en resolver el problema de detección de objetos utilizando una única cámara.

- **Modelo geométrico del objeto** - Por otro lado, se conoce el modelo geométrico de un conjunto de N objetos:

  $$\mathcal{S} = \{X^j | j = 1, \ldots, N\}$$

  donde cada uno de ellos está descrito por un conjunto de M^j puntos de su estructura relativos a un origen de referencia local al propio objeto O^j (para objetos planares, el sistema de referencia local se va a definir de forma que el eje z coincida con el vector normal al plano):

  $$X^j = \{X_k^j \in \mathbb{R}^3 | k = 1, \ldots, M^j\}$$

  El objeto tendrá una posición y orientación desconocidas. El objetivo de este proyecto se centra en buscar la pose de dicho objeto respecto al origen de coordenadas global O.

- **Información medida** - En esta situación, tenemos un objeto en el espacio de trabajo (o varios objetos iguales repetidos) $X^j \in \mathcal{S}$ observado por las cámaras. Para una cámara cualquiera, se obtiene un conjunto de proyecciones en el plano imagen de los puntos de X^j:

  $$m^j = \{m_i^j \in \mathbb{R}^2 | i = k_1, \ldots, k_{\hat{N}^j}\}$$

El conjunto de índices $k_1, \ldots, k_{\hat{N}^j}$ representa la asociación entre los puntos de medida m^j y los puntos del modelo X^j. En general, $\hat{N}^j \leq N^j$ debido, por ejemplo, a oclusiones con el entorno o con su misma estructura opaca.

- **Posición y orientación del objeto** - La posición y orientación del objeto en el espacio queda completamente definida por una matriz de transformación homogénea $T \in \mathbb{R}^{4\times4}$ que relaciona el origen de referencia relativo al objeto O^j con el origen de coordenadas global del espacio de trabajo O:

$$T = \begin{pmatrix} R & \mathbf{t} \\ \mathbf{0} & 1 \end{pmatrix} \tag{6.1}$$

donde $R \in \mathbb{R}^{3\times3}$ representa la matriz ortogonal de rotación y $\mathbf{t} \in \mathbb{R}^{3\times1}$ es el vector de traslación. Para obtener estas dos matrices se va a utilizar la información que proporcionan las imágenes captadas por las cámaras.

Esta primera parte del proyecto se va a centrar en la detección de objetos planares. Como se explicó en el apartado 3.2.1 del capítulo 3, la proyección de un plano cualquiera en el plano imagen de una cámara induce una homografía. Por tanto, para objetos cuyos puntos X^j se encuentran en un mismo plano ($z = 0$), la pose de éste se reduce al cálculo de una homografía representada por la matriz $H \in \mathbb{R}^{3\times3}$. Dicha matriz de homografía relaciona los puntos del objeto X^j con las proyecciones de los mismo en el plano imagen de cada cámara m^j:

$$m_k^j = \lambda H \begin{pmatrix} x_k^j \\ y_k^j \\ 1 \end{pmatrix} \qquad X_k^j = \begin{pmatrix} x_k^j \\ y_k^j \\ 0 \end{pmatrix} \tag{6.2}$$

Por otro lado, los puntos del objeto y sus proyecciones se se relacionan entre si a través de la matriz de proyección de la cámara y la matriz de transformación:

$$\begin{aligned} m_k^j &= \lambda P \cdot T\, X_k^j \qquad \lambda \in \mathbb{R} \\ P \cdot T &= (\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3, \mathbf{p}_4) \qquad \mathbf{p}_i \in \mathbb{R}^{3\times1} \end{aligned} \tag{6.3}$$

donde P es la matriz de proyeción de la cámara.

Eliminando $\mathbf{p}_3$ puesto que $z_k^j = 0$ y relacionando las ecuaciones 6.2 y 6.3 se obtiene la relación directa existente entre la matriz de homografía H y la matriz de transformación del objeto:

$$H = (\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_4) \tag{6.4}$$

La suposición de objeto plano se puede extender a objetos tridimensionales, siempre y cuando el tamaño del objeto sea muy inferior a la distancia del mismo a la cámara que lo observa. Por ejemplo, esta es la situación frecuente en el espacio inteligente, y por tanto, el modelo geométrico considerado para cualquier objeto que se encuentre dentro del espacio inteligente será una homografía.

- **Modelo de apariencia** - Tenemos un conjunto de puntos que forman el modelo geométrico del objeto y un conjunto de proyecciones en el plano imagen, sin embargo, aun se desconoce la matriz de homografía que relaciona dichos puntos entre sí. Para realizar la asociación de puntos, se hace uso de un modelo de apariencia para los puntos de X^j en la imagen:

$$A^j = \{A_k^j | k = 1, \ldots, k = N^j\}$$

Dicho modelo de apariencia tiene como objetivo asignar una descripción A_k^j relacionada con el aspecto de cada punto X_k^j en el plano imagen. Por tanto, la apariencia se compone de información que se adquiere directamente de la imagen: color, textura, forma, etc. En general, esta información varía con la pose del objeto y con otros factores externos como la iluminación, por lo que hay que buscar un modelo de apariencia que sea lo más invariante posible a todos estos cambios y que a su vez sea suficiente para discriminar un punto buscado del resto.

Para obtener la correspondencia entre los puntos de la imagen y los del modelo geométrico, cada A_k^j del modelo de apariencia se compara con el equivalente medido en la imagen $\hat{A}_k^j$. Una vez que se asocia el conjunto de medidas m^j a los puntos del modelo geométrico de X^j se obtiene la matriz de homografía H. Si además se conoce la matriz de calibración de la cámara P, se puede obtener la pose T del objeto.

En este proyecto se propone utilizar los descriptores del método SIFT para generar el modelo de apariencia de los objetos debido al buen comportamiento que presentan frente a cambios de iluminación, en la pose del objeto, etc, tal y como se explico en el capítulo 4.

En este caso, el modelo de apariencia no se va a generar a partir del modelo geométrico del objeto sino que se toma una imagen del objeto planar como patrón y se localizan los descriptores SIFT en dicha imagen. Al igual que antes, la imagen patrón (modelo de apariencia) y la imagen a analizar están relacionadas mediante una homografía H. Además, el objeto planar tendrá asociado un sistema de coordenadas métricas dispuesto por conveniencia en el centro del propio objeto donde el eje z se corresponde con el vector normal al plano. Mediante el conocimiento de dos distancias métricas sobre el objeto y sus correspondientes en la imagen se construye una

homografía H_m que permite trasladar los puntos del patrón imagen al sistema métrico relativo al plano. De esta forma, el conjunto de puntos X^j se construye de forma acorde con el modelo de apariencia previamente calculado con el método SIFT.

$$\mathbf{X}_i^j = \lambda H_m \cdot H\ \mathbf{m}_i^j \qquad \longrightarrow \qquad \mathbf{X}_i^j = \begin{pmatrix} x_i^j \\ y_i^j \\ 1 \end{pmatrix} \tag{6.5}$$

Partiendo del modelo de apariencia y de una imagen cualquiera del objeto, el proposito es obtener la matriz de homografía H. Una vez que se obtiene la matriz H el cálculo de la matiz de transformación es inmediata si se conoce H_m. La matriz H_m se obtiene a partir de las relaciones de distancia entre puntos conocidos del patrón y las distancias físicas reales entre los mismos puntos del objeto. A partir de ahora, para no complicar más la notación, se va a utilizar $\mathbf{X}^j$ para representar el conjunto de los puntos que forman el modelo de apariencia (los puntos de la imagen patrón) en vez de los puntos del modelo geométrico.

Por otro lado, proceso de asociación de los descriptores SIFT se corresponde con el proceso de "matching" inicial que se explico en el apartado 4.2.2 del capítulo 4. Partimos de un conjunto de puntos en la imagen patrón de los que se ha obtenido sus correspondientes descriptores:

$$\mathbf{X}^j = \{\mathbf{X}_k^j | k = 1, \ldots, N\} \quad \rightarrow \quad A^j = \{A_k^j | k = 1, \ldots, N\}$$

donde:

$$\mathbf{X}_k^j = \begin{pmatrix} x_k^j \\ y_k^j \\ 1 \end{pmatrix}$$

De la misma manera, se obtiene otro conjunto de puntos en la imagen de la escena a analizar junto con sus descriptores correspondientes:

$$\mathcal{M} = \{\mathbf{m}_k | k = 1, \ldots, M\} \quad \rightarrow \quad \hat{A} = \{\hat{A}_k | k = 1, \ldots, M\}$$

donde:

$$\mathbf{m}_k = \begin{pmatrix} u_k \\ v_k \\ 1 \end{pmatrix}$$

Mediante el proceso de "matching", se busca una correspondencia entre los descriptores emparejando puntos del patrón y puntos de la imagen con descriptores similares. De esta forma, se obtiene un conjuto de $\hat{N}$ pares de puntos, $\mathbf{m}_k \longleftrightarrow \mathbf{X}_k^j$.

En la figura 6.1 se muestra un ejemplo de dicha correspondencia inicial:

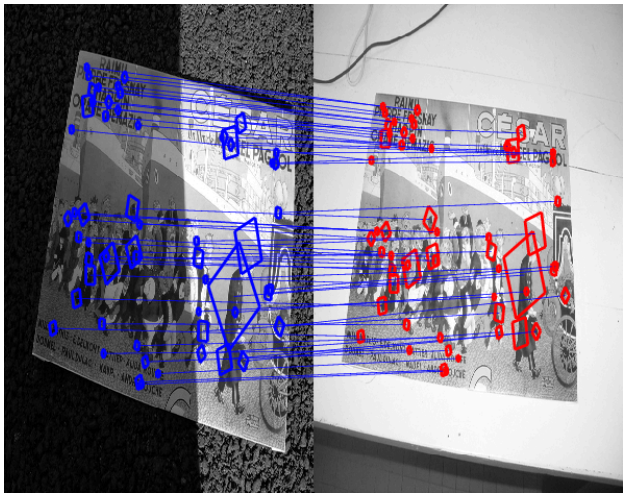


Figura 6.1: **Descripción general del problema.** *Ejemplo de correspondencia inicial*

Aunque este método de "matching" propuesto para el método SIFT es en general bastante bueno, parte de estas correspondencias iniciales serán "outliers". También hay que tener en cuenta que en la misma imagen puede haber varios objetos iguales. Para obtener una solución correcta de la matriz H es necesario eliminar las falsas correspondencias. Además, se tendrá que agrupar los "inliers" (en los casos de múltiples objetos) para obtener la matriz de homografía de cada de ellos. Por tanto, hay que utilizar algún mecanismo que permira descartar estos "outliers" de antemano y agrupar los "inliers".

En este proyecto se propone utilizar dos métodos distintos de estimación robusta que permiten descartar los "outliers", incluso cuando el porcentaje de los mismos es muy alto y al mismo tiempo, permiten determinar si hay más de una repetición del objeto en la imagen y obtener la solución de la pose de cada una. Estos dos métodos propuestos, de los cuales ya se realizó una definición general de ambos en el capítulo 5, son:

- **Transformada de Hough.**
- **RANSAC.**

En este capítulo se va a describir de forma detallada cómo se aplican ambos métodos al problema que se plantea en este proyecto, primero suponiendo que el modelo de la cámara es afín y posteriormente se generalizará el problema suponiendo una cámara proyectiva. También se describirán algunas modificaciones que se han realizado al algoritmo de RANSAC para mejorar su eficiencia.

6.2. Detección de objetos planares mediante aproximación de cámara afín

Si se toma una imagen con una cámara proyectiva aumentando la distancia focal, se puede apreciar que las deformaciones proyectivas que sufren los objetos disminuyen (en la imagen, las líneas paralelas se asemejan más a líneas paralelas). Lo mismo ocurre si se toman imágenes de objetos cada vez más alejados de la cámara. En estos casos, el modelo de cámara finita se puede aproximar a una **cámara afín**.

Como ya se explicó en el apartado 3.1.6 del capítulo 3, una de las características de la cámara afín es que las líneas paralelas en $\mathbb{P}^3$ se proyectan como líneas paralelas en el plano imagen, es decir, el paralelislo es un invariante. Debido a esta característica, la relación entre un plano en $\mathbb{P}^3$ y su proyección en el plano imagen viene dada por una **transformación afín** en vez de por una homografía. Por tanto, al aproximar el modelo de la cámara proyectiva a un modelo afín, la relación existente entre los puntos del patrón X^j y las proyecciones del objeto en la imagen será una transformación afín:

$$m_k^j = \lambda H \begin{pmatrix} x_k^j \\ y_k^j \\ 1 \end{pmatrix} \qquad X_k^j = \begin{pmatrix} x_k^j \\ y_k^j \\ 0 \end{pmatrix}$$

donde:

$$H = \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ 0 & 0 & 1 \end{pmatrix} \tag{6.6}$$

De esta forma, el problema de la obtención de la pose del objeto se simplifica pues la matriz H tiene únicamente 6 grados de libertad para el caso de afinidad (recordemos que el caso más general, el de homografía, la matriz H tiene 8 grados de libertad). Al realizar la aproximación de cámara afín, se gana en tiempo de computo, sin embargo, las soluciones de posición y orientación que se obtienen no serán las exactas. Hay que remarcar que esta aproximación sólo es valida si la distancia entre la cámara y los objetos es grande comparada con el tamaño de los mismos.

En este apartado sólo se va a describir el proceso seguido para el cálculo de la pose utilizando la Transformada de Hough y RANSAC junto con la aproximación de cámara afín. En el capítulo **??** se realizará un estudio del rango de distancias que debe haber entre la cámara y los objetos para que los errores sean despreciables y de los errores que se obtienen al suponer esta aproximación.

6.2.1. Transformada de Hough

Dado un conjunto de N correspondencias, la Transformada de Hough es el primer método que se propone para separar los "inliers" de los "outliers", agruparlos en función del objeto al que pertenezcan (en el caso en que haya más de un mismo objeto en la imagen) y así poder calcular la matriz afín que modela la transformación espacial entre los puntos del patrón $\mathbf{X} = (x, y)^T$ y los puntos de cada objeto detectado en la imagen, $\mathbf{m} = (u, v)^T$:

$$\begin{aligned} \mathbf{m} &= D\mathbf{X} + d \\ &= \begin{pmatrix} d_{x,x} & d_{x,y} \\ d_{y,x} & d_{y,y} \end{pmatrix} \mathbf{X} + \begin{pmatrix} d_x \\ d_y \end{pmatrix} \end{aligned} \tag{6.7}$$

Para utilizar la Transformada de Hough, es necesario parametrizar dicha relación con el menor número de parámetros, pues la eficiencia del algoritmo decae considerablemente a medida que aumenta el número de parámetros. La parametrización óptima se calcula mediante la descomposición QR de la matriz afín. Dicha descomposición establece que dada una matriz $A \in \mathbb{R}^{N \times N}$, existe siempre una descomposición de la forma:

$$A = Q \cdot R$$

donde R es una matriz de rotación ortonormal y Q es una matriz triangular inferior:

$$Q = \begin{pmatrix} Q_{1,1} & 0 & \dots & 0 \\ Q_{2,1} & Q_{2,2} & \ddots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ Q_{N,1} & Q_{N,2} & \dots & Q_{N,N} \end{pmatrix} \qquad R \in \mathbb{O}(r)$$

En el caso de la matriz afín, la descomposición QR es la siguiente:

$$D = \begin{pmatrix} K_x & 0 \\ s & K_y \end{pmatrix} \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \qquad d = \begin{pmatrix} d_x \\ d_y \end{pmatrix} \tag{6.8}$$

donde:

- $K_x \rightarrow$ es el factor de escala en el eje x.
- $K_y \rightarrow$ es el factor de escala en el eje y.
- $\theta \rightarrow$ es el ángulo de rotación de Euler en el eje z del sistema de referencia del objeto planar.
- $d_x \rightarrow$ es la componente del vector de desplazamiento en el eje x.
- $d_y \rightarrow$ es la componente del vector de desplazamiento en el eje y.

- $s \rightarrow$ es el parámetro que representa la deformación que se produce debido a los efectos de perspectiva en los que se rompe la ortogonalidad de los ejes. El parámetro s está relacionado con el ángulo de deformación entre la imagen y el eje ortogonal de la siguiente manera:

$$s = \tan \alpha$$

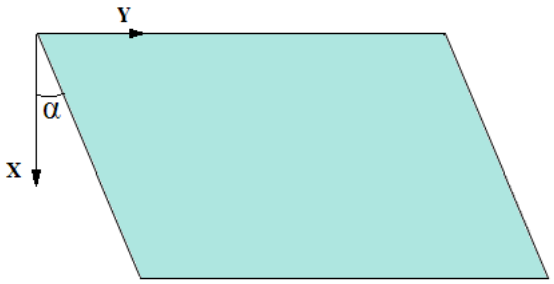


Figura 6.2: **Detección de objetos planares mediante la aproximación de cámara afín.** *Relación entre el parámetro s y el ángulo de deformación.*

Estos son los 6 párametros que se van a utilizar en la Transformada de Hough. Por tanto, el acumulador de Hough tendrá 6 dimensiones (como es lógico, coincide con los grados de libertad de la matriz afín). Desarrollando la expresión 6.8, la relación afín entre dos puntos se puede expresarse de la siguiente manera:

$$\begin{cases} u = K_x\big(x\cos(\theta) - y\sin(\theta)\big) + d_x \\ \\ v = K_y\big(x\sin(\theta) + y\cos(\theta)\big) + s\big(x\cos(\theta) - y\sin(\theta)\big) + d_y \end{cases} \tag{6.9}$$

Una vez que se han definido los parámetros que permiten describir una afinidad (las dos componentes de la escala, el factor de deformación, el ángulo de rotación y las dos componentes de traslación) se puede implementar la Transformada de Hough aplicada a la detección de objetos planares.

Con el programa de Lowe, además de extraer los descriptores de una imagen, también aporta información sobre la localización de los mismos, su factor de escala y su ángulo de rotación. Debido a que la eficiencia de la Transformada de Hough disminuye considerablemente al aumentar el número de parámetros, una primera idea para mejorar la eficiencia sería utilizar la información anterior sobre la escala y la rotación para así reducir a 4 el número de parámetros de Hough:

- Para cada pareja de puntos del "matching" inicial se puede obtener la rotación relativa entre ambos calculando la diferencia entre el ángulo de rotación del descriptor en la imagen y el del descriptor correspondiente en el patrón.
- Puesto que también se conoce la escala de los descriptores, se puede obtener el factor de escala de cada correspondencia.
- De esta forma, se podría construir un acumulador previo utilizando únicamente los factores de escala y las rotaciones relativas de todas las correspondencias iniciales. Los máximos de dicho acumulador probablemente se corresponderán con uno o varios objetos en la imagen (puede haber más de un mismo objeto con la misma escala y ángulo de rotación). Para cada pareja de valores de escala y rotación obtenidos en el acumulador anterior, se construye un segundo acumulador para los 4 parámetros restantes y así obtener la transformación afín completa.

Sin embargo, esto no se puede aplicar en todos los casos. Obtener la rotación que ha sufrido el objeto en la imagen mediante la relación entre los ángulo de los descriptores sólo es válido si la rotación real que sufre el objeto se produce en el eje z y en el resto de ejes los ángulos de rotación son pequeños. Cuando la deformación proyectiva deja de ser despreciable, esto ya no se cumple. Respecto a la escala, la información que se obtiene tampoco es válida. El programa de Lowe devuelve el parámetros σ del espacio de escala que se corresponde con un factor de escala unidimensional. Sin embargo, para calcular la transformación afín se necesita conocer las escalas independientes en cada dimensión K_x y K_y. La información que aporta σ sobre la escala sólo es válida en ausencia de deformaciones proyectivas cuando el escalado aplicado al objeto es uniforme ($K_x = K_y$). Por tanto hay que construir un acumulador de 6 dimensiones.

Para construir el acumulador se parte de un conjunto de N correspondencias entre puntos de la imagen y del patrón obtenidas en el "matching" inicial. Como se explicó en el algoritmo 2 del capítulo 5, para cada correspondencia $\mathbf{X}_i \leftrightarrow \mathbf{m}_i$ se obtienen los valores discretos de los 6 parámetros $(K_x,\ K_y,\ \theta,\ s,\ d_x,\ d_y)$, calculando el valor de uno de ellos en función de todas las combinaciones de valores posibles que puedan tomar el resto de parámetros.

Observando la ecuación 6.9, dos de los parámetros de una transformación afín son independientes entre ellos, d_x y d_y. Por tanto, se construirá el acumulador tomando como variables dependientes estos dos parámetros:

$$\begin{cases} d_x = f(K_x, \theta) & \longrightarrow \quad d_x = u - K_x\big(x\cos(\theta) - y\sin(\theta)\big) \\ d_y = f(K_y, s, \theta) & \longrightarrow \quad d_y = v - K_y\big(x\sin(\theta) + y\cos(\theta)\big) - s\big(x\cos(\theta) - y\sin(\theta)\big) \end{cases}$$

Una vez obtenido el acumulador, hay que buscar los máximos relativos del mismo. El conjunto de valores $(K_x,\ K_y,\ \theta,\ s,\ d_x,\ d_y)$ de cada máximo encontrado se corresponde con un objeto detectado en la imagen cuya transformación afín queda definida con los 6 parámetros anteriores. Si además del acumulador de Hough, se crea una estructura de datos donde se lleve un registro de los pares de puntos que "votan" por cada celda del acumulador, se podrá determinar los puntos que se corresponden con "inliers" y agruparlos en función del objeto detectado al que pertenezcan.

Es importante recalcar que no basta con buscar el máximo absoluto del acumulador pues puede haber más de un objeto en la imagen o incluso ninguno. Además, como ya se explicó en el apartado 5.1 del capítulo 5, la búsqueda de máximos relativos debe ser una busqueda controlada pues no todos los máximos del acumulador se corresponderán con un objeto detectado:

- Varios pares de puntos del "matching" inicial pueden "votar" por una misma celda del acumulador generando un máximo (máximos espurios, de baja amplitud y generalmente, con desviaciones grandes), aunque esa combinación de parámetros no se corresponda con ninguna transformación afín presente en la imagen.
- En general, los máximos no serán picos abruptos sino que se asemejarán a distribuciones gaussianas con desviaciones más o menos pequeñas (todos los "inliers" de un mismo objeto no "votan" por la misma celda sino que los votos se encuentran concentrados en un area de varias celdas vecinas). Este fenómeno se debe a la presencia de ruido y a que los valores de los parámetros son discretos.

Por tanto, hay que buscar un método que descarte los máximos espurios, permita encontrar los máximos locales del acumulador y que a su vez, evalúe la probabilidad de que dichos máximos se correspondan con un objeto en la imagen. Los máximos validos serán aquellos cuya amplitud sobrepase un cierto umbral y además presenten una desviación pequeña. Para ello se va a uilizar el algoritmo K-medias descrito también en el capítulo 5. Sin embargo, no podemos aplicar dicho algoritmo directamente pues el K-medias necesita conocer de antemano el número de clases distintas. Como en nuestro caso el número de objetos presentes en la imagen es desconocido es necesario realizar una serie de modificaciones al algoritmo para que el número de clases sea adaptativo.

6.2.1.1. Modificación del algoritmo K-medias

Hay que modificar el algoritmo K-medias para que el número de clases sea variable. El algoritmo comienza con un número de clases K fijo (en concreto, $K = 2$) y su valor se irá adaptando en función de los resultados que se van obteniendo en cada iteración del mismo.

El punto de partida es un acumulador normalizado de M dimensiones, donde cada celda toma un cierto valor w_i. La modificación de K-medias es la siguiente:

1. El primer paso es aplicar el Algoritmo 3 del capítulo **??**. De forma iterativa, el algoritmo agrupa todos los puntos del acumulador en N clases utilizando la distancia de Mahalanobis. Cada clase se corresponderá con una función Gaussiana M-dimensional cuya media y varianza final se calcula en función de los puntos pertenecientes a cada clase.

2. Ahora hay que determinar si las agrupaciones son buenas y si el número de clases N estimado es correcto. Se va considerar que una clase es válida si su función Gaussiana se adapta bien al perfil que forman los puntos pertenecientes a ella. Por tanto, si el error entre la Gaussiana y el perfil real está por debajo de un umbral mínimo, se considera que la clase es correcta.

   Sin embargo, si el error es grande, parece lógico pensar que la función Gaussiana no se adapta bien por lo que dicha clase se descarta. Ahora bien, hay que intentar estimar la causa por la que la clasificación no ha sido correcta. Supongamos los siguientes casos:

   - Supongamos que el error que se comete al intentar aproximar la función Gaussiana al perfil de los puntos es muy grande. Una justificación bastante probable a este hecho es que la clase esté compuesta por puntos que realmente pertenecen a clases distintas, tal y como se muestra en la figura 6.3. Por esta razón, el error que se comete es elevado (en este ejemplo, el maximo de la Gaussiana coincide prácticamente con valores reales muy pequeños). Por tanto, si el error supera un cierto umbral máximo, se descarta la clase y además, se aumenta en una unidad el número de clases K.

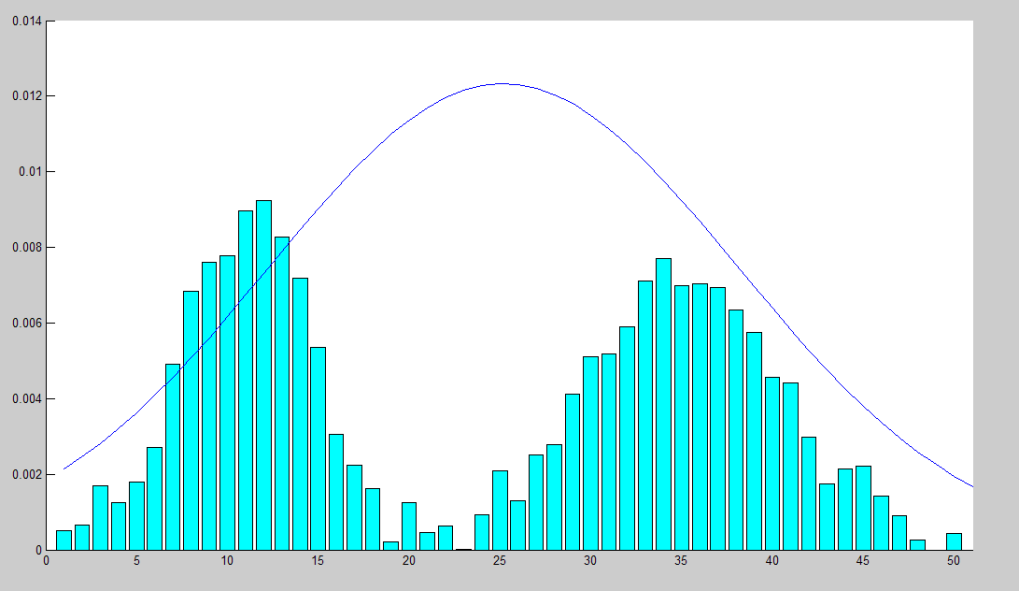


Figura 6.3: **Detección de objetos planares mediante aproximación de cámara afín.** *Ejemplo de clasificación*

- Ahora supongamos que el error supera el umbral inferior que permite determinar si una clase el válida, pero aun así, no es lo suficientemente elevado como para determinar que se necesita aumentar el número de clases. Por tanto, la clase se descarta pero el valor de K no se modifica. Estos casos se producen, por ejemplo, cuando se detecta varias clases distintas donde realmente sólo habría una, tal y como se muestra en la figura 6.4.

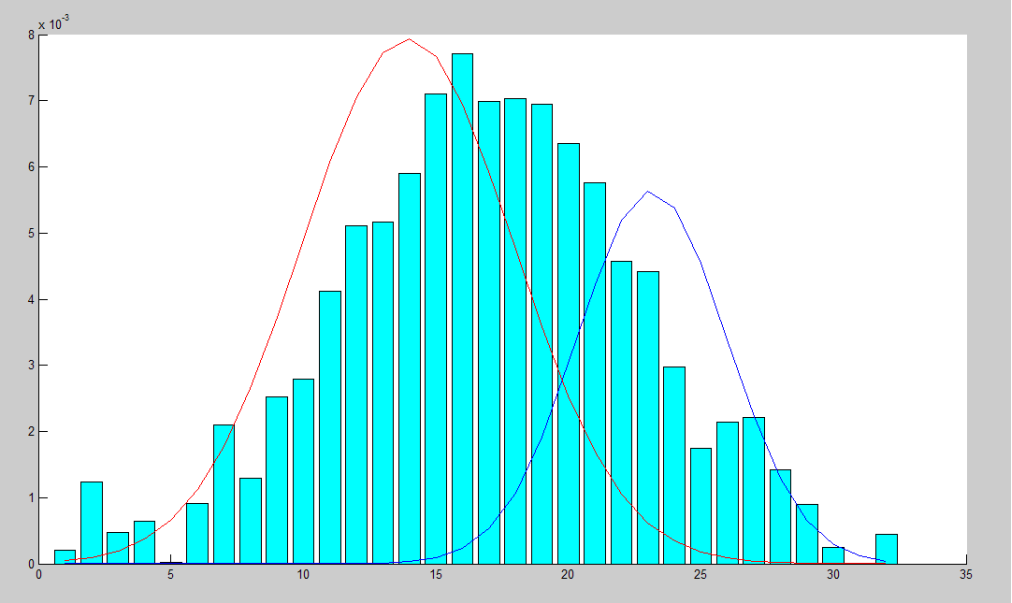


Figura 6.4: **Detección de objetos planares mediante aproximación de cámara afín.** *Ejemplo de clasificación*

- Si alguna clase se queda vacía, se elimina y se decrementa en una unidad el valor de K.

3. Finalmente, se anulan todos los puntos del acumulador pertenecientes a las clases que fueron detectadas correctamente y se vuelve a aplicar los pasos anteriores con el resto de puntos del acumulador (pertenecientes a clases erroneas). Teóricamente, este proceso se repetirá hasta que todos los puntos hayan sido agrupados. Sin embargo, en la práctica, el número de objetos presentes en la imagen es limitado (simplemente por el hecho de que el tamaño de la imagen es limitado) e incluso, puede ser nulo. Por tanto, para evitar bucles infinitos (en el caso de que no haya ninguna clase) o que el número K de clases estimadas crezca indefinidamente, se ha limitado el número total de iteraciones del algoritmo y a su vez, se ha fijado un valor máximo para K.

En los casos prácticos, se ha observado que los picos de las diferentes clases en vez de asemejarse a funciones Gaussianas, se asemejan más a funciones exponenciales M-dimensionales. En la figura 6.5 se muestra un corte del acumulador en función de dos parámetros, dejando el resto fijos a un valor. Se puede observar como la forma del pico (que se corresponde con un objeto) sigue una distribución exponencial.

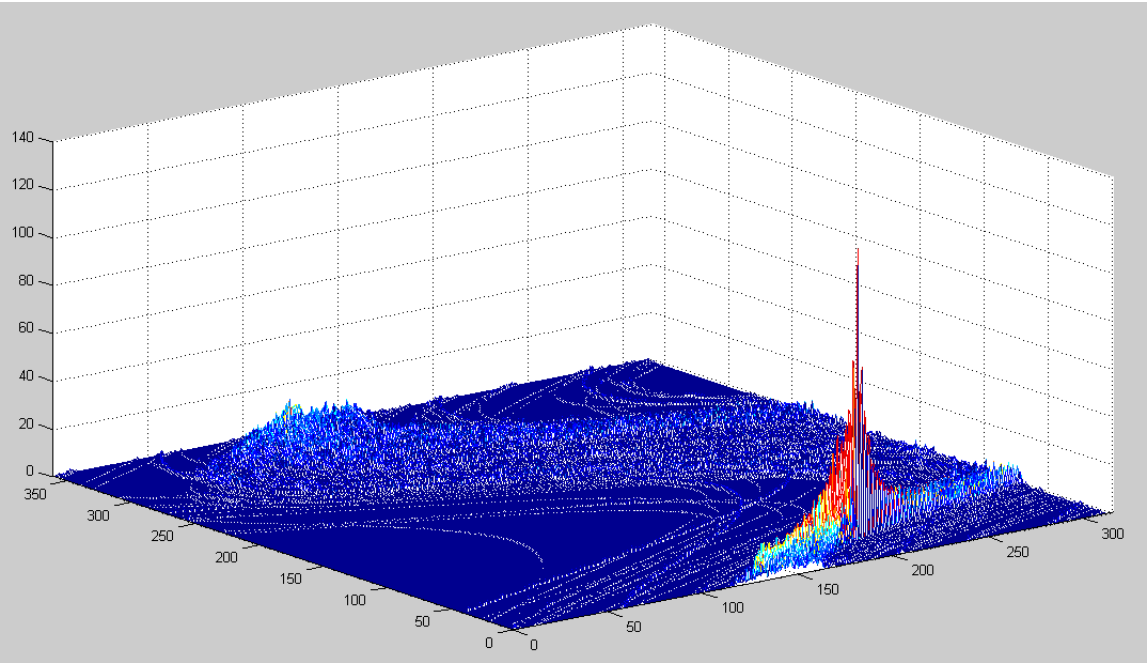


Figura 6.5: **Detección de objetos planares mediante aproximación de cámara afín.** *Ejemplo en el que se muestra un corte de un acumulador de Hough. Puesto que el acumulador es de 6 dimensiones, la representación se ha realizado en función de dos de sus parámetros fijando el resto a un determinado valor.*

La función de aproximación que se va a utilizar es la siguiente:

$$g(\mathbf{x}) = \frac{K}{(2\pi)^{N/2}(|\Sigma|)} \exp\Big[-(\mathbf{x}-\hat{\mathbf{x}})^T \Sigma^{-1} (\mathbf{x}-\hat{\mathbf{x}}) \Big] \tag{6.10}$$

Hay que asegurarse que la función anterior se corresponde con una función de densidad de probabilidad y calcular su media y varianza (que se corresponderán con los estadísticos calculados con K-medias en función de los puntos de cada clase) para determinar de qué forma se relacionan con $\hat{\mathbf{x}}$ y Σ. Para simplificar los cálculos, se va a suponer que la dimensión es 1 ($N = 1$):

$$\int_{-\infty}^{\infty} \frac{K}{\sqrt{2\pi}\sigma} \exp\left(-\frac{|x-\hat{x}|}{\sigma} \right) = 1 \tag{6.11}$$

$$\int_{-\infty}^{\infty} \frac{K}{\sqrt{2\pi}\sigma} \exp\left(- \frac{|x-\hat{x}|}{\sigma} \right) dx =$$

$$= \frac{K}{\sqrt{2\pi}\sigma} \left[\int_{-\infty}^{\hat{x}} \exp\left(\frac{x-\hat{x}}{\sigma} \right) dx + \int_{\hat{x}}^{\infty} \exp\left(- \frac{x-\hat{x}}{\sigma} \right) dx \right] =$$

$$= \frac{K}{\sqrt{2\pi}\sigma} \left[2\sigma \exp\left(\frac{x-\hat{x}}{\sigma} \right) \Big|_{-\infty}^{\hat{x}} - 2\sigma \exp\left(- \frac{\hat{x}-x}{\sigma} \right) \Big|_{\hat{x}}^{\infty} \right] =$$

$$= \frac{2K}{\sqrt{2\pi}} \qquad \longrightarrow \qquad K = \sqrt{2\pi}2$$

Por tanto:

$$g(\mathbf{x}) = \frac{1}{2^N(|\Sigma|)} \exp\Big[- (\mathbf{x}-\hat{\mathbf{x}})^T \Sigma^{-1} (\mathbf{x}-\hat{\mathbf{x}}) \Big] \tag{6.12}$$

Respecto a la media:

$$E[x] = \int_{-\infty}^{\infty} \frac{x}{2\sigma} \exp\left(- \frac{|x-\hat{x}|}{\sigma} \right) =$$

$$= \int_{-\infty}^{\hat{x}} \frac{x}{2\sigma} \exp\left(\frac{x-\hat{x}}{\sigma} \right) + \int_{\hat{x}}^{\infty} \frac{x}{2\sigma} \exp\left(- \frac{x-\hat{x}}{\sigma} \right)$$

Integrando por partes ambas integrales se obtiene:

$$E[x] = \frac{1}{2} \Bigg[x \exp\left(\frac{x-\hat{x}}{\sigma} \right) \Big|_{-\infty}^{\hat{x}} - \int_{-\infty}^{\hat{x}} \exp\left(\frac{x-\hat{x}}{\sigma} \right) -$$

$$- x \exp\left(- \frac{x-\hat{x}}{\sigma} \right) \Big|_{\hat{x}}^{\infty} + \int_{\hat{x}}^{\infty} \exp\left(- \frac{x-\hat{x}}{\sigma} \right) \Bigg] =$$

$$= \frac{1}{2}\left[\hat{x} - \sigma\ \exp\left(\frac{x-\hat{x}}{\sigma}\right)\Bigg|_{-\infty}^{\hat{x}} + \hat{x} + \sigma\ \exp\left(\frac{x-\hat{x}}{\sigma}\right)\Bigg|_{\hat{x}}^{\infty}\right]$$

Por tanto:

$$E[x] = \hat{x}$$

Por último se obtiene la relación existente entre la varianza y σ:

$$VAR[x] = E[x^2] - E[x]^2$$

$$E[x^2] = \int_{-\infty}^{\infty} \frac{x^2}{2\sigma}\ \exp\left(-\frac{|x-\hat{x}|}{\sigma}\right) =$$

$$= \int_{-\infty}^{\hat{x}} \frac{x^2}{2\sigma}\ \exp\left(\frac{x-\hat{x}}{\sigma}\right) + \int_{\hat{x}}^{\infty} \frac{x^2}{2\sigma}\ \exp\left(-\frac{x-\hat{x}}{\sigma}\right)$$

Integrando por partes:

$$E[x] = \frac{1}{2}\Bigg[x^2 \exp\left(\frac{x-\hat{x}}{\sigma}\right)\Bigg|_{-\infty}^{\hat{x}} - \int_{-\infty}^{\hat{x}} 2x\ \exp\left(\frac{x-\hat{x}}{\sigma}\right) -$$

$$-x^2 \exp\left(-\frac{x-\hat{x}}{\sigma}\right)\Bigg|_{\hat{x}}^{\infty} + \int_{\hat{x}}^{\infty} 2x\ \exp\left(-\frac{x-\hat{x}}{\sigma}\right)\Bigg] =$$

$$= \hat{x}^2 + 4\sigma \qquad \longrightarrow \qquad VAR[x] = 4\sigma$$

En el algoritmo 7 se muestra todas las modificaciones realizadas al algoritmo K-medias:

Algoritmo 7 Clasificación de puntos utilizando el algoritmo K-medias modificado con distancia de Mahalanobis

Punto de partida: Acumulador normalizado M-dimensional de N celdas:

$$\mathcal{X} = \{\mathbf{X}_j \in \mathbb{R}^M | j = 1, \ldots, N\}$$

Cada celda está ponderada por un valor normalizado: w_j

Inicialización:
Número de clases iniciales: $K_{ini} = 2 \quad \rightarrow \quad K = K_{ini}$.
Número máximo de clases: $K_{max} = 6$.
Umbrales: $\epsilon_{max},\ \epsilon_{min}$.
Número máximo de iteraciones: N^{it}_{max}.

while $((N^{it} < N^{it}_{max})\ \&\ (K < K_{max})\ \&$ (existan celdas no nulas)) **do**
  Aplicar el Algoritmo **??** del capítulo 5 para obtener las K clases.

$$\text{Clase } C_i \ \rightarrow \ \begin{cases} E[\mathcal{X}^i] = \mu_i, \quad VAR[\mathcal{X}^i] = 4\Sigma_i \\ \\ \mathcal{X}^i = \{(\mathbf{X}^i_j,\ w^i_j) | j = 1, \ldots, N_i\} \end{cases} \qquad \text{para:} \quad i = 1, \ldots, K$$

  for $(i = 1$ to $K)$ **do**
    if (C_i está vacía) **then**
      $K = K - 1$
    else
      Se calcula el error cuadrático medio.

$$g(\mathbf{X}) = \frac{1}{2^N (|\Sigma|)^{1/2}}\ \exp\Big[-(\mathbf{X} - \mu)^T \Sigma^{-1} (\mathbf{X} - \mu) \Big]$$

$$\text{Error} = \sqrt{\sum_{j=1}^{N} \frac{(w^i_j - g(\mathbf{X}^i_j))^2}{N}}$$

      if $(\text{Error} > \epsilon_{max})$ **then**
        Se descarta la clase C_i.
        $K = K + 1$
      else if $(\text{Error} < \epsilon_{min})$ **then**
        La clase C_i es válida y se almacena $(\mu_i,\ \Sigma_i,\ \mathcal{X}^i)$ en una estructura de datos.
        Se ponen a cero las celdas del acumulador pertenecientes a la clase.
        $K = K - 1$
      else
        Se descarta la clase C_i.
      end if
    end if
  end for
end while

Una vez que finaliza el algoritmo K-medias, los parámetros de salida son las medias, las matrices de covarianza y los puntos pertenecientes a cada una de las clases. El valor de cada centroide se corresponde con los 6 parámetros de la matriz de transformación entre el patrón y un posible objeto en la imagen. Por último, hay que evaluar si estos máximo relativos se corresponden realmente con un objeto en la imagen:

- El número de descriptores SIFT que se detectan en una sola imagen es grande, por lo que en general, el número de correspondencias correctas entre el patrón y cada uno de los objetos presentes será elevado (en condiciones normales de oclusión, escala, etc). De esta forma, si una clase detectada con K-medias se corresponde realmente con una transformación afín, el número de puntos de dicha clase deberá ser elevado. Por tanto, los máximos cuya amplitud no sobrepase un cierto umbral se descartan.

  Es más, debido a la forma de construir el acumulador, es muy probable que gran parte de las celdas reciban alguna "votación". Así que seguramente el resto de celdas que no pertenezcan a un pico tomarán valores no nulos influyendo de forma negativa a la hora de calcular los centroides y las matrices de covarianza. Esto se puede asemejar a una señal de ruido. Por tanto, antes de aplicar el algoritmo K-medias al acumulador, se va a realizar un filtrado previo anulando las celdas cuyo valor no sobrepase un umbral mínimo (fijado de forma adaptativa en función del valor máximo de todo el acumulador).

  El problema de este filtrado previo es que además de eliminar el ruido"del acumulador, también es posible eliminar picos de amplitud muy pequeña que se corresponden a objetos con alto grado de oclusión.

  En la figura 6.6 se muestra otro corte del mismo acumulador del ejemplo de la figura 6.5, pero en este caso se corresponde con una zona donde no hay ningún máximo. Se puede apreciar que a pesar de ser una región del acumulador donde no hay ningún máximo, las celdas no toman un valor nulo.

- Las correspondencias pertenecientes a un mismo objeto seguramente no "votarán" por el mismo conjunto de valores $(K_x,\ K_y,\ \theta,\ s,\ d_x,\ d_y)$ pero estas "votaciones" estarán concentradas en una área reducida en torno al conjunto de parámetros $(K_x,\ K_y,\ \theta,\ s,\ d_x,\ d_y)$. Por tanto, para considerar que una clase se corresponde realmente con un objeto, no sólo hay que evaluar las amplitudes de los máximos sino que también hay que comprobar que la desviación típica de la misma no supere un cierto valor.

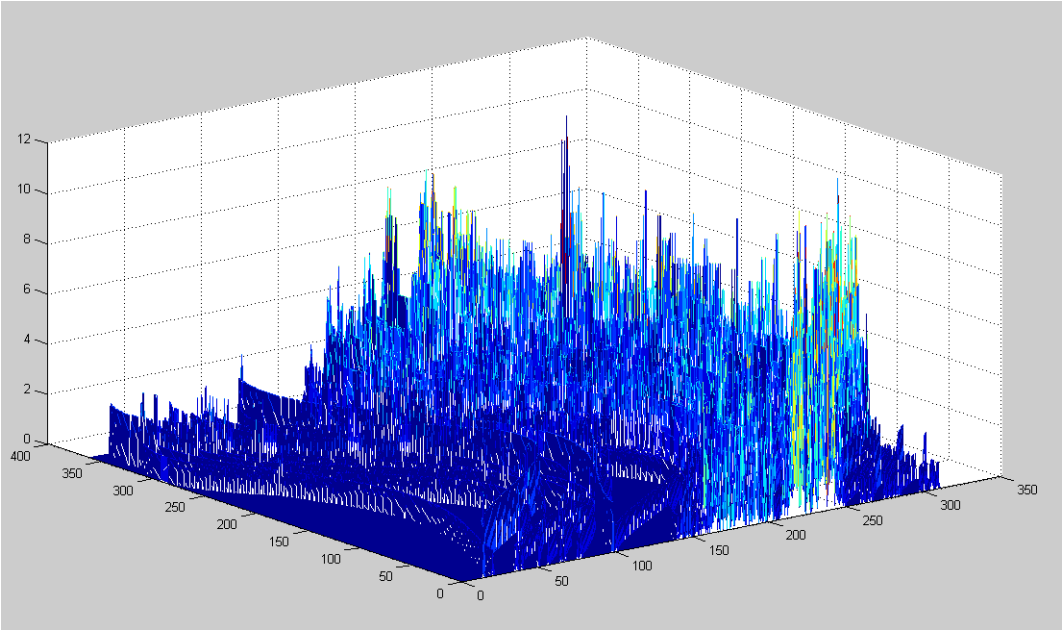


Figura 6.6: **Detección de objetos planares mediante aproximación de cámara afín.** *Ejemplo en el que se muestra un corte de un acumulador de Hough. Puesto que el acumulador es de 6 dimensiones, la representación se ha realizado en función de dos de sus parámetros fijando el resto a un determinado valor.*

6.2.2. Implementación final de la Transformada de Hough para la detección de objetos planares

El número de parámetros necesarios para definir una transformación afín son 6, por lo que el acumulador de Hough tendrá 6 dimensiones. Hay que definir el rango de valores que podrán tomar estos parámetros en función de las condiciones en las que pueden estar los objetos en las imágenes a analizar:

- Ángulo de rotación $\Longrightarrow$ Los objetos podrán estar en cualquier posición, por tanto se va a considerar que θ puede variar entre 1 y 360 grados, tomando variaciones de 1 grado.

$$\theta \in [1^{o}, 360^{o}] \qquad \Delta\theta = 1^{o}$$

- Parámetro de deformación s $\Longrightarrow$ Se va considerar que la deformación máxima del objeto estará en torno a los ± 30 grados. Por tanto, el parámetro s tomará valores entre $\pm 0{,}6$ con incrementos de $0{,}01$ (aproximadamente medio grado). En total, el parámetro s podrá tomar 121 valores distintos.

$$s \in [-0{,}6, 0{,}6] \qquad \Delta s = 0{,}01$$

- Escala $K \implies$ Se va a considerar que el factor de escala podrá tomar valores entre $1{,}3$ y $0{,}3$ con incrementos de $0{,}1$ (se va a utilizar 11 valores distintos de escala en cada eje).

$$K \in [0{,}3, 1{,}3] \qquad \Delta K = 0{,}1$$

- Desplazamiento $\implies$ Los objetos podrán aparecer centrados en cualquier pixel de la imagen. Como se utilizan imágenes de 680×480, se van a considerar desplazamientos máximos de 480 píxeles en el eje x y de 680 píxeles en el eje y, con variaciones de 1 píxel.

$$d_x \in [0, 480] \qquad \Delta d_x = 1\text{pixel}$$

$$d_y \in [0, 680] \qquad \Delta d_y = 1 \text{ pixel}$$

Por tanto, el acumulador de 6 dimensiones tendrá un tamaño de $360 \times 121 \times 11 \times 11 \times 480 \times 680$. Sin embargo, en la práctica no es posible construir con Matlab una matriz de estas dimensiones por limitaciones de memoria. Incluso disminuyendo el rango de valores de cada parámetro sigue dando problemas. Debido a esto, no se puede implementar el algoritmo de completo de la Transformada de Hough tal cual se ha descrito en los apartado anteriores. Por tanto, se ha realizado la siguiente modificación:

- Si analizamos la ecuación 6.9, se puede observar que los parámetros d_x y d_y son independientes entre ellos y además, el único parámetro común del que ambos dependen es el ángulo de rotación θ:

$$d_x = f(K_x, \theta)$$

$$d_y = f(K_y, s, \theta)$$

- Con la primera ecuación de 6.9, se puede construir un acumulador de 3 dimensiones, calculando los valores de d_x en función de los pares de correspondencias $\mathbf{X_i} \leftrightarrow \mathbf{m}_i$ y todas las combinaciones posibles de valores de K_x y θ.

$$d_x = u - K_x\big(x\cos(\theta) - y\sin(\theta)\big)$$

  Con los máximos de este acumulador se obtienen los valores de 3 de los 6 de los parámetros de una posible transformacion afín. El método para evaluar los máximos y determinar si son o no válidos es el mismo que el descrito en el apartado anterior.

- Para cada posible candidato del acumulador anterior se construye un nuevo acumulador, ahora de 4 dimensiones, calculando d_y en función de los distintos valores que puedan tomar K_y, s y θ. Para el ángulo de rotación, se toma un intervalo de valores reducido en torno al ángulo obtenido en el primer acumulador.

$$d_y = v - K_y\big(x\sin(\theta) + y\cos(\theta)\big) - s\big(x\cos(\theta) - y\sin(\theta)\big)$$

- De esta forma, se obtienen los 3 parámetros restantes de la transformación afín (en caso de que se haya detectado algún pico en el segundo acumulador).

6.2.3. Resultados

Los resultado obtenidos con la Transformada de Hough no son satisfactorios. A continuación se enumeran algunos de los problemas encontrados:

- Por un lado, tenemos la limitación de memoria de Matlab. Para poder solucionarlo, se tendría que implementar el algoritmo en algún lenguaje de programación como por ejemplo C/C++.
- La causa principal de los malos resultados de la Transformada de Hough se deben al algoritmo de K-medias. Se ha probado este algoritmo generando de forma sintéticas clases de tipo Gaussianas y los resultados eran bastante buenos (en el capítulo **??** se mostró un ejemplo de cómo el algoritmo convergía hacia una solución). Sin embargo, a la hora de aplicar K-medias a un acumulador real, el algoritmo diverge en muchas ocasiones.
- Otro problema aparece con los umbrales de error mínimo y máximo de K-medias. Se ha intentado hacer un ajuste de estos umbrales con poco éxito, pues los umbrales que son validos para una imagen, no lo son para otra debido a que la forma de los picos son muy variados. En ocasiones, por ejemplo, los picos serán muy abruptos formados por pocas celdas con valores muy altos. Con que pertenezca a esta clase algún punto un poco alejado del pico, la función exponencial se expande por lo que el error aumenta.
- También existe ambigüedad a la hora de determinar los umbrales para la matriz de covarianzas. Además, estos umbrales no pueden ser iguales, dependerán del parámetro que tengan asociado. Por ejemplo, la máxima desviación que se puede permitir al ángulo de rotación no será la misma que para el parámetro de deformación s.
- Para terminar, otra desventaja de este método es que el tiempo de ejecución es muy alto, debido a la gran cantidad de operaciones que debe realizar el algoritmo para construir los acumuladores, además del alto consumo de memoria. Esto hace que su aplicación, en el caso de haber obtenido buenos resultados, fuese limitada. Por ejemplo, en aplicaciones de tiempo real o en sistemas con memoria limitada (como un robot) sería imposible utilizar este método.

6.2.4. RANSAC

El punto de partida es un conjunto de $\hat{N}$ pares de puntos obtenidos en el "matching" inicial entre un patrón y una imagen. Hay que determinar cuáles de estos pares se corresponden con "inliers" (en el caso que haya al menos un objeto en la imagen) y a su vez clasificarlos si existe más de un mismo objeto en la imagen. Para ello, se va a proponer un segundo método, RANSAC.

En el apartado 5.2 del capítulo 5 se realizó una breve descripción general del método de RANSAC. En concreto, se va a utilizar el Algoritmo 6 que está adaptado para multiples objetos. Antes de implementar dicho algoritmo tenemos que definir una serie de parámetros necesarios:

- Hay conocer la función paramétrica que relaciona los puntos del patrón con los "inliers" de la imagen y el número mínimo de pares de puntos necesarios para calcular el valor de los parámetros de la función.
- Una vez que se ha obtenido una solución, hay que asignarle una votación en función de los puntos que se adapten a ella sin sobrepasar un umbral t.

6.2.4.1. Definición del modelo paramétrico

La relación entre los puntos del patrón de un objeto y los "inliers" en la imagen viene dada por la matriz de afinidad. Es más, supongamos que tenemos M repeticiones en la imagen del mismo objeto, cada una de ellas tendrá asociada una matriz de afinidad distinta:

$$\mathbf{m}_k^i = H_i \mathbf{X}_k^i \qquad i = 1, \ldots, M \tag{6.13}$$

donde:

$$H_i = \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ 0 & 0 & 1 \end{pmatrix} \tag{6.14}$$

Para aplicar el método de RANSAC, es necesario conocer la función paramétrica $f(\mathbf{\Phi})$ que relacione las correspondencias correctas entre ellas. Desarrollando la expresión 6.19 se obtienen las ecuaciones paramétricas que relacionan los puntos del patrón con las proyecciones del objeto en la imagen:

$$\mathbf{m}_k^i = H_i \mathbf{X}_k^i \qquad \longrightarrow \qquad \begin{cases} u_k^i = h_{11}\, x_k^i + h_{12}\, y_k^i + h_{13} \\ \\ v_k^i = h_{21}\, x_k^i + h_{22}\, y_k^i + h_{23} \end{cases} \tag{6.15}$$

donde el vector de parámetros $\mathbf{\Phi}$ esta compuesto por los 6 elementos de las dos primeras filas de la matriz H:

$$\mathbf{\Phi} = (h_{11}, h_{12}, h_{13}, h_{21}, h_{22}, h_{23})^T$$

Utilizando la notación matricial, la ecuación 6.15 para n pares de puntos queda de la siguiente forma:

$$\begin{pmatrix} u_1 \\ v_1 \\ \vdots \\ u_n \\ v_n \end{pmatrix} = \begin{pmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_1 & y_1 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_n & y_n & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_n & y_n & 1 \end{pmatrix} \cdot \mathbf{\Phi} \tag{6.16}$$

El sistema anterior es de la forma $A\mathbf{h} = \mathbf{b}$ donde A es una matriz de dimensiones $2n \times 6$.

En la expresión 6.15 se puede observar que de cada par de puntos se obtienen 2 ecuaciones linealmente independientes. Puesto que la matriz de afinidad tiene 6 grados de libertad, se necesitan como mínimo 3 puntos para obtener una solución de H_i. Como se explicó en el apartado 5.2 del capítulo 5, el método RANSAC, al contrario de los algoritmos de estimación tradicionales, en vez de usar la mayor cantidad de datos posibles para obtener una solución inicial y luego eliminar los datos erróneos, utiliza el menor conjunto de datos que permita calcular una solución inicial, aumentando dicho conjunto con datos consistentes que se adapten al modelo obtenido. Por tanto, en este caso, RANSAC tomará de forma aleatoria 3 pares de puntos del conjunto de correspondencias iniciales cada vez que calcule una solución para el modelo. Este número mínimo de puntos viene reflejado en el parámetro s del algoritmo de RANSAC, por tanto $s = 3$.

Sin embargo, no todas las combinaciones de puntos serán válidos para obtener una solución de la matriz H. Para poder obtener una solución no degenerativa es necesario que los 3 puntos no sean colineales. Una homografía (y por tanto una transformación afín) define la relación entre los puntos de dos planos. Basta con tomar tres puntos de un plano para definirlo siempre que estos no pertenezcan a una misma recta. En este caso, no definen un único plano, sino un conjunto infinito de planos que se intersecan que intersecan en la recta definida por los tres puntos. Por tanto, al no tener un único plano definido, no se puede obtener la relación entre planos.

Cada vez que RANSAC calcula una solución para la matriz H, va tomando de forma aletoria 3 pares de correspondencias hasta obtener un conjunto no colineal con el que calcular una solución. Para evitar bucles infinitos, se fija un número máximo de iteraciones para encontrar este conjunto de datos que originan una solución no degenerativa.

6.2.4.2. Validación y votación de cada modelo inicial

Para estimar una solución válida para la matriz de afinidad, el algoritmo de RANSAC toma 3 pares de puntos de forma aleatoria y calcula la matriz H_j. Una vez que obtiene esta matriz, se comprueba qué pares de puntos de todo el conjunto de correspondencias iniciales se adapta a la solución obtenida.

Para determinar si la matriz H_j se adapta a una correspondencia $\mathbf{X_k} \leftrightarrow \mathbf{m}_k$, como medida de error se va a utilizar el error de reproyección:

$$\left.\begin{array}{l} d_1 = (\mathbf{m}_k - H_j\mathbf{X}_k)^2 \\ \\ d_2 = (\mathbf{X}_k - H_j^{-1}\mathbf{m}_k)^2 \end{array}\right\} \quad \longrightarrow \quad d_k^j = \sqrt{d_1 + d_2} \tag{6.17}$$

Todas las correspondencias cuya distancia (calculada según la ecuación anterior) sea inferior a un umbral t se consideran como "inliers" del modelo afín obtenido. Cada matriz H_j obtenida con RANSAC recibirá una puntuación en función del numero de "inliers". De entre todos los candidatos obtenidos, finalmente se toma como solución la matriz H_j con mayor puntuación y se recalcula H_j teniendo en cuenta ya todos los "inliers" y así obtener una mejor estimación de la transformación real entre el patrón y el objeto detectado.

El valor de t se ha fijado de forma experimental en $0{,}085$. Dicho umbral tampoco puede ser muy restrictivo pues se está suponiendo que los objetos únicamente sufren deformaciones afines.

Dentro de un rango pequeño de deformaciones proyectivas, las correspondencias que realmente pertenecen a un objeto se adaptan a los modelos afines calculados con RANSAC. Sin embargo, al aumentar la deformación, la distancia d crece y no todos los puntos se adaptarán a un único modelo sino que en general, se obtendrán varias matrices afines, cada una de ellas con su conjunto de "inliers" correspondiente cuando realmente dichos puntos se corresponderían con un sólo objeto. A partir de cierto ángulo de deformación, el ajuste entre los puntos y los modelos se hace casi imposible, unido también al hecho de que el número de descriptores detectados disminuye considerablemente. Todo esto se analizará con más detalle en el capítulo 7.

Aunque el valor del umbral no puede ser restrictivo, tampoco se puede poner un umbral mayor. En ese caso, teóricamente las correspondencias correctas se podrían ajustar mejor a los modelos, sin embargo, aumenta la probabilidad de que las falsas correspondencias se ajusten también y se consideren "inliers".

6.2.4.3. Algoritmo de RANSAC para el cálculo de la matriz afín de múltiples objetos

Para implementar el algoritmo de RANSAC que permita calcular las matrices afines de cada objeto detectado en la imagen, se va a utilizar como base varias funciones de Matlab implementadas por Peter Kovesi, profesor de la Escuela de Ciencias de la Computación e Ingeniería de Software de la Universidad de Western Australia. Dichas funciones se pueden descargar en la siguiente página web:

`http://www.csse.uwa.edu.au/~pk/research/matlabfns/`

Además del modelo paramétrico y del umbral t, en el apartado 5.2 del capítulo 5 se definio una serie de parámetros propios del algoritmo general de RANSAC. En concreto, a cada parámetros se le va a asignar los siguientes valores:

- **Número mínimo de muestras** $s \rightarrow s = 3$.
- **Umbral de distancia** $t \rightarrow t = 0{,}085$.
- **Probabilidad** $p \rightarrow p = 0{,}99$.
- **Número de iteraciones** $N^{it} \rightarrow$ Este parámetro se va a calcular en cada iteración, tal y como se comentó en el Algoritmo 4 del capítulo 5.
- **Proporción de "outliers"** $\epsilon \rightarrow$ En cada iteración se irá recalculando su valor en función del número de "inliers" encontrados. Con este valor se utiliza para el cálculo de N^{it}.
- **Número de iteraciones máxima** $N^{it}_{max} \rightarrow N^{it}_{max} = 1000$.
  Este umbral se fija para evitar bucles infinitos. Independientemente del valor que tome N^{it} y de si se ha encontrado o no un modelo, el algoritmo finaliza si llega a la iteración 1000.
- **Número de iteraciones** $N^{deg} \rightarrow N^{deg} = 100$.
  Este umbral define el máximo de iteraciones para encontrar un conjunto de puntos no degenerativos.

A continuación, se muestra todo el algoritmo completo de RANSAC aplicado a la detección de múltiples objetos y al cálculo de las matrices de transformación afín:

Algoritmo 8 Algoritmo de RANSAC para el cálculo de transformaciones afines

Punto de partida: Tenemos un conjunto Y de M correspondencias $\mathbf{X}_i \leftrightarrow \mathbf{m}_i$ entre la imagen y el patrón:

$$Y = \{(\mathbf{X}_i, \mathbf{m}_i) \mid i = 1, \ldots, M\}$$

Inicialización: $N^{it}_{max} = 1000,\ N^{deg} = 100,\ s = 3,\ p = 0{,}99,\ t = 0{,}085,\ ,\text{cont} = 1$

while (cont = 1) **do**
  $H_{best} = NaN$, $N^{inlier} = 0$, num_deg $= 0$, num_it $= 0$
  while (num_it $< min(N^{it}, N^{it}_{max})$) **do**
    Se selecciona de forma aleatoria un subconjunto Y^s de s elementos de Y.
    if ($(Y^s$ es degenerativo) & (num_deg $< N^{deg}$)) **then**
      Se toma un nuevo subconjunto Y^s de forma aleatoria.
      num_deg = num_deg + 1
    else
      Se calcula $H = f(Y^s)$.
      Se obtiene el conjunto Y^{inlier} formado por N^{inlier} puntos de Y que se corresponden con los "inliers" de la matriz H:

$$Y^{inlier} = \left\{(\mathbf{X}_i, \mathbf{m}_i) \mid \left((\mathbf{m}_k - H_j\mathbf{X}_k)^2 + (\mathbf{X}_k - H_j^{-1}\mathbf{m}_k)^2\right)^{1/2} < t\right\}$$

      if $(N^{inlier} > N^{inlier}_{best})$ **then**
        $H_{best} = H$ $\quad Y^{inlier}_{best} = Y^{inlier}$
        $\epsilon = 1 - (N^{inlier}/M)$ $\quad N^{it} = \frac{log(1-p)}{log(1-(1-\epsilon)^s)}$
      end if
      num_it = num_it + 1
    end if
    if $((H_{best} = NaN) \mid (\text{num\_it} > N^{it}_{max}))$ **then**
      cont $= 0$
    else
      Se almacena H_{best} e Y^{inlier}_{best} en una estructura de datos.
      $Y = \{Y - Y^{inlier}_{best}\}$
      if (length(Y) $< s$) **then**
        cont = 0
      end if
    end if
  end while
end while

En la mayoría de los casos prácticos, el número de "outliers" presentes en la imagen es elevado por lo que la probabilidad de encontrar conjuntos reducidos de "outliers" que generen una solución válida con RANSAC es alta (en la práctica, se ha observado que el número de correspondencias asociadas a un modelo erróneo suele estar entre los 10 ó 7 puntos o incluso menos). El número de puntos asociado a estas falsas soluciones es muy reducido en comparación con el número de "inliers" de las soluciones válidas. Por tanto, si el número de correspondencias asociadas a una solución no supera un cierto umbral se descarta.

El valor del umbral de decisión no puede ser el mismo para todos los casos, sino que se calcula en función del número de descriptores totales encontrados en la imagen patrón. De forma experimental, se ha fijado que el número de "inliers" necesarios para considerar que una solución es válida debe superar el $3\,\%$ del número total de descriptores del patrón.

6.2.4.4. Modificaciones del algoritmo de RANSAC

El tiempo de ejecución del Algoritmo 9 crece de forma exponencial a medida que aumenta el número de objetos a detectar en la imagen. El porcentaje de "outliers" para cada objeto aumenta considerablemente (hay que tener en cuenta que estos "outliers" no son solamente las falsas correspondencias con el fondo, sino también los "inliers" correspondientes al resto de objetos aún no detectados) y por tanto, el número de iteraciones N se dispara.

En el Algoritmo 9, en cada iteración se man tiene únicamente un solo modelo. A medida que encuentra nuevos modelo con un número de "inliers" mayor, descarta los anteriores. Es muy probable que algunos de estos modelos descartados se correspondan con otros objetos de la imagen con menor número de "inliers" y que en iteraciones posteriores serán detectados nuevamente.

Si en vez de descartar un modelo cuando se encuentra otro mejor, se alamcenan

En esta apartado se propone una mejora para el algoritmo de RANSAC. En cada iteración, se mantiene un conjuto $\bar{\mathbf{Phi}}$ con los m mejores modelos encontrados (en vez de descartar un modelo cuando se encuentra otro con más votación), de forma que al finalizar la iteración se obtendrá un conjunto de soluciones $\bar{\mathbf{Phi}}$. El valor de m dependerá de la probabilidad p de diseño. Además, cada solución adicional en $\bar{\mathbf{Phi}}$ será correcta con una probabilidad menor que p, sin embargo, el tiempo de cómputo mejora considerablemente. En el Algoritmo **??** se muestra la modificación propuesta.

Algoritmo 9 Modificación del Algoritmo de RANSAC para múltiples objetos

Inicialización: $N^1_{best} = 0, \ldots, N^m_{best} = 0$, $\mathbf{\Phi}^1_{best} = NaN, \ldots, \mathbf{\Phi}^m_{best} = NaN$, $p = 0{,}99$, num_it $= 0$

while (num_it $< min(N^{it}, N^{it}_{max})$) **do**

  Se selecciona de forma aleatoria un subconjunto Y^s de s elementos de Y no degenerativos.

  Se calcula $\mathbf{\Phi} = g(Y^s)$.

  Se obtiene el conjunto Y^{inlier} formado por N^{inlier} puntos de Y que se corresponden con los "inliers" de la matriz $\mathbf{\Phi}$:

$$Y^{inlier} = \left\{(y_i, \mathbf{m}_i) \mid |f(y_i,\ \mathbf{\Phi})| < t\right\}$$

  for $i = 1$ to m **do**

    if $\left((N^{inlier} > N^i_{best}) \quad \& \quad (\mathbf{\Phi} \neq \{\mathbf{\Phi}^1_{best}, \ldots, \mathbf{\Phi}^m_{best}\})\right)$ **then**

      $\mathbf{\Phi}^i_{best} = \mathbf{\Phi}$

      $Y^i_{best} = Y^{inlier}$

      $N^i_{best} = N^{inlier}$

      break

    end if

  end for

  $\epsilon = 1 - (N^1_{best}/M) \qquad N^{it} = \frac{log(1-p)}{log(1-(1-\epsilon)^s)}$

  num_it $=$ num_it $+ 1$

end while

En la figura **??** se puede observar la mejora de tiempos de computo al introducir la mejora del algoritmo. El experimentoestima un conjunto de N matices afines con 200 puntos por cada matriz, de los cuales 30 son "outliers". A medida que se incrementa el número N de matrices a estimar, el número totales de puntos a estimar aumenta y por tanto la proporcion de "outliers" ϵ. El número de modelos simulataneos que se pueden estimar en cada iteración se fija a 5. Dicho número se ha obtenido de forma experimental y para valores de m mayores no se obtiene ninguna mejora significativa.

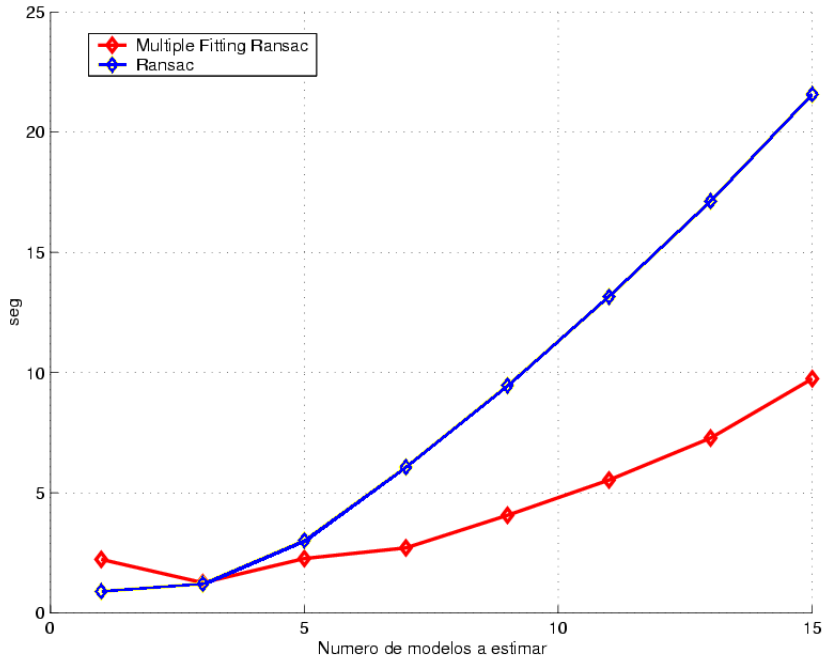


Figura 6.7: **RANSAC con aproximación afín** *Mejora en el tiempo de computo al introducir la modificación en el algoritmo de RANSAC.*

6.2.5. Resultados

A continuación se van a mostrar algunos ejemplos en los que se han detectado varios objetos utilizando el algoritmo de RANSAC junto con la aproximación de cámara afín. Los resultados que se van a visualizar son los siguientes:

- **"Matching" inicial** - Se va a mostrar la imagen y el patrón en la misma figura. Además, se va a trazar una línea de color verde entre cada par de correspondencias encontradas con el método SIFT.
- **"Inliers"** - En esta gráfica se muestran los "inliers" detectados por RANSAC. En el caso de haber más de un objeto, se ha utilizado diferentes colores para representar las líneas que unen los puntos del patrón con los de los de la imagen (cada objeto tendrá un color diferente).
- **"Outliers"** - Se muestra el conjunto de "outliers" resultantes tras aplicar RANSAC.
- **"Reproyección del perfil del objeto** - Para cada objeto detectado, RANSAC no sólo devuelve los "inliers" asociados a él, sino que también se obtiene la matriz de transformación sufrida por el objeto patrón. Esta imagen permite ver claramente si la estimación de la matriz H es buena, o si por el contrario, el error que se ha cometido es alto.

- **Primer ejemplo**.

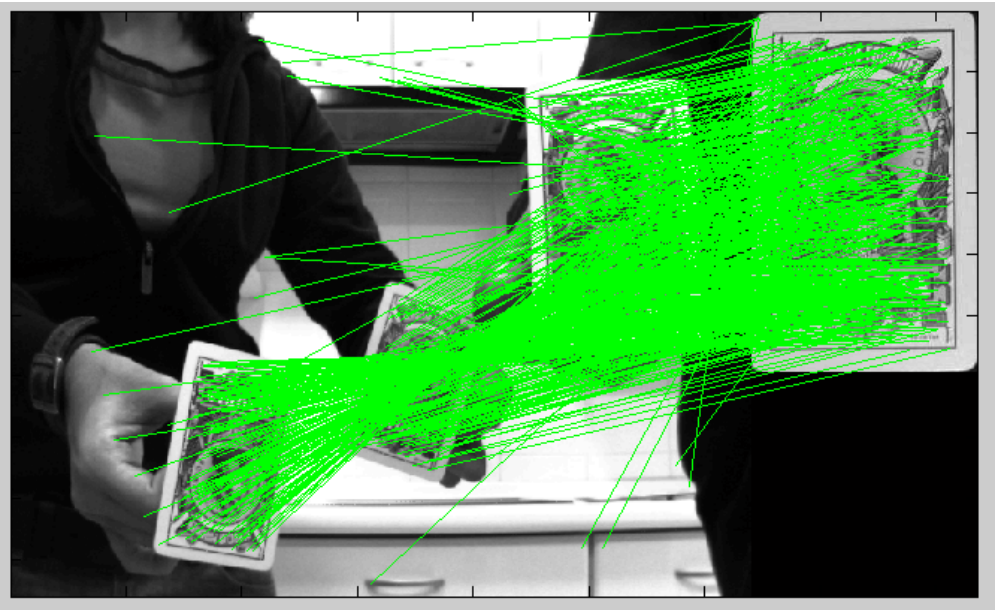


Figura 6.8: **RANSAC utilizando el modelo de cámara afín.** *En esta imagen se muestra el "matching" inicial obtenido por SIFT. Cada par de correspondencias entre un punto del patrón y la imagen se representa con una recta de color verde .*

Figura 6.9: **RANSAC utilizando el modelo de cámara afín.** *En esta imagen se muestran los "inliers" tras aplicar RANSAC junto con la aproximación afín.*

Figura 6.10: **RANSAC utilizando el modelo de cámara afín.** *En esta imagen se muestran los "outliers" tras aplicar RANSAC junto con la aproximación afín.*

Figura 6.11: **RANSAC utilizando el modelo de cámara afín.** *En esta imagen se muestra el perfil de reproyección del objeto detectado tras aplicar RANSAC junto con la aproximación afín.*

- **Segundo ejemplo**.

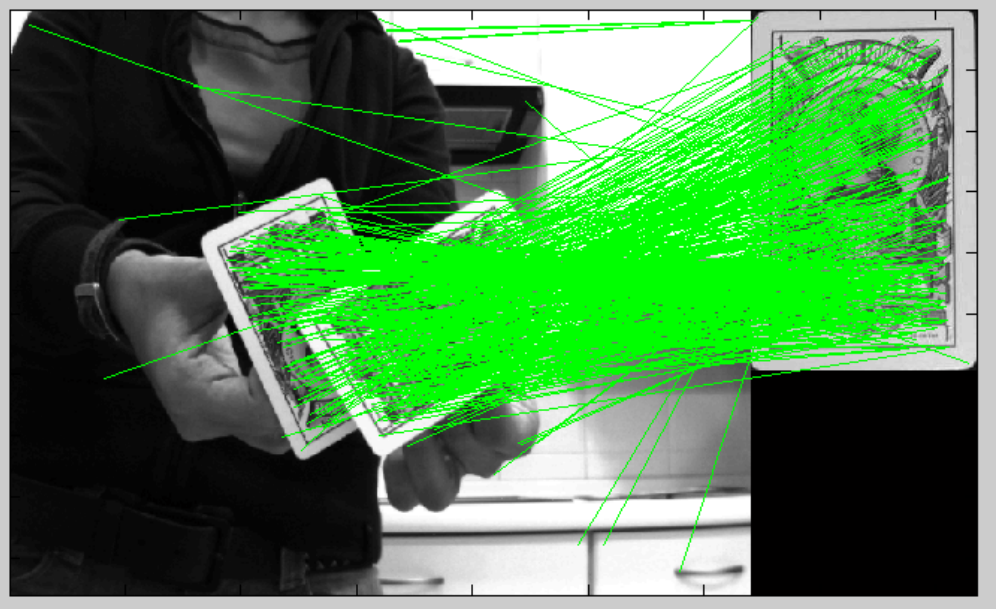


Figura 6.12: **RANSAC utilizando el modelo de cámara afín.** *En esta imagen se muestra el "matching" inicial obtenido por SIFT. Cada par de correspondencias entre un punto del patrón y la imagen se representa con una recta de color verde .*

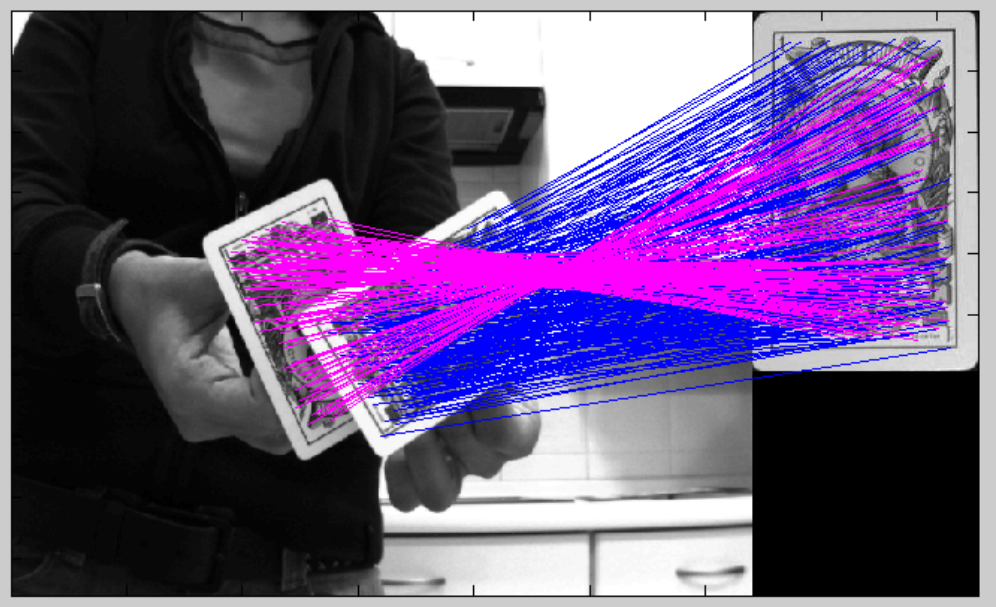


Figura 6.13: **RANSAC utilizando el modelo de cámara afín.** *En esta imagen se muestran los "inliers" tras aplicar RANSAC junto con la aproximación afín.*

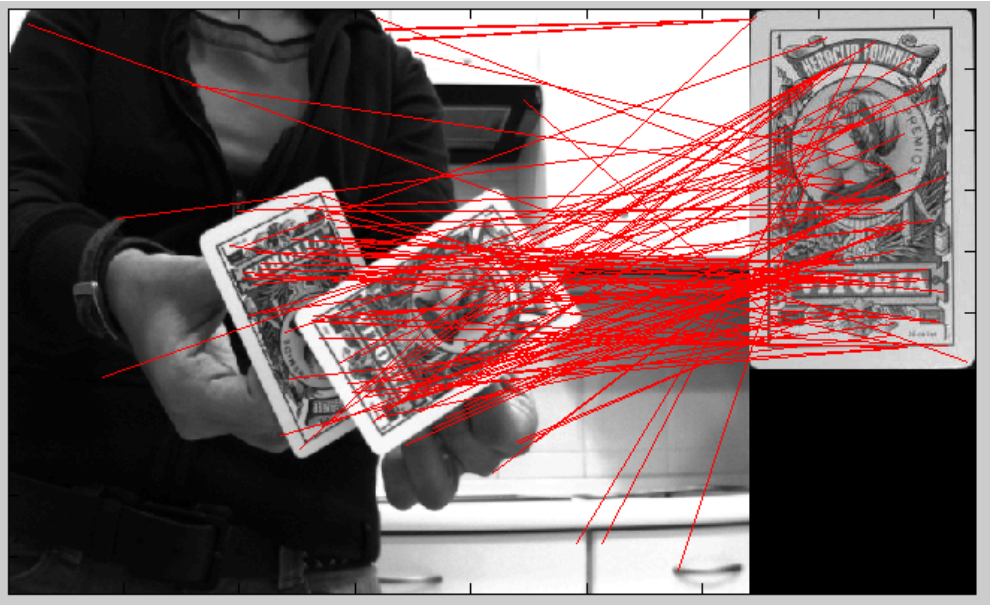


Figura 6.14: **RANSAC utilizando el modelo de cámara afín.** *En esta imagen se muestran los "outliers" tras aplicar RANSAC junto con la aproximación afín.*

Figura 6.15: **RANSAC utilizando el modelo de cámara afín.** *En esta imagen se muestra el perfil de reproyección del objeto detectado tras aplicar RANSAC junto con la aproximación afín.*

- **Tercer ejemplo**.

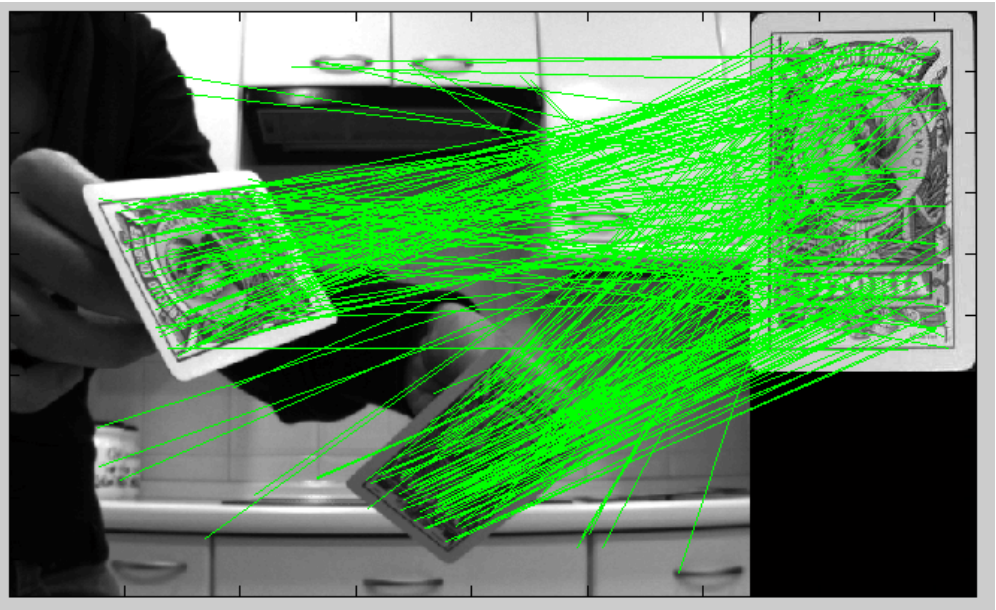


Figura 6.16: **RANSAC utilizando el modelo de cámara afín.** *En esta imagen se muestra el "matching" inicial obtenido por SIFT. Cada par de correspondencias entre un punto del patrón y la imagen se representa con una recta de color verde* .

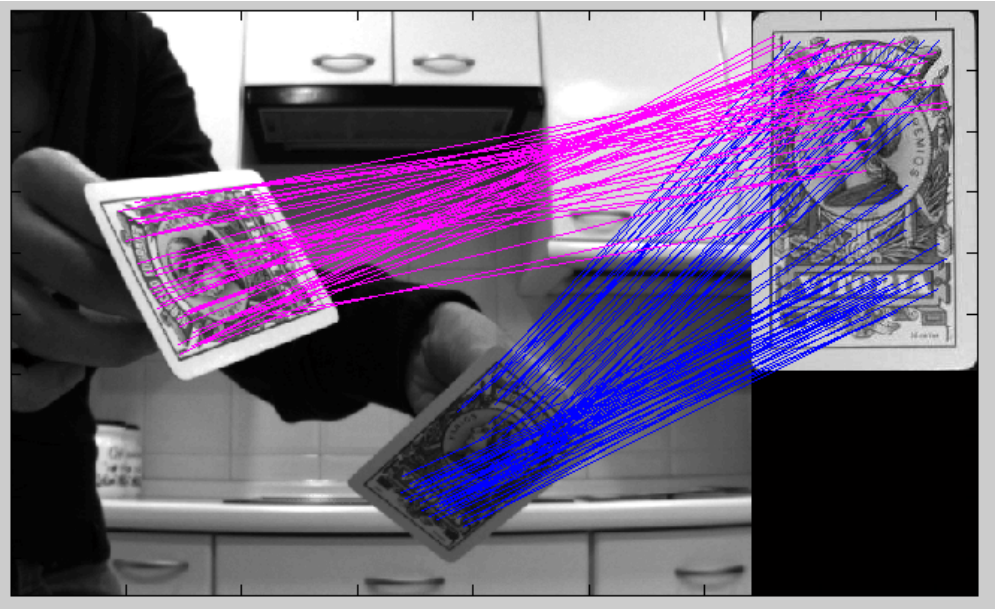


Figura 6.17: **RANSAC utilizando el modelo de cámara afín.** *En esta imagen se muestran los "inliers" tras aplicar RANSAC junto con la aproximación afín.*

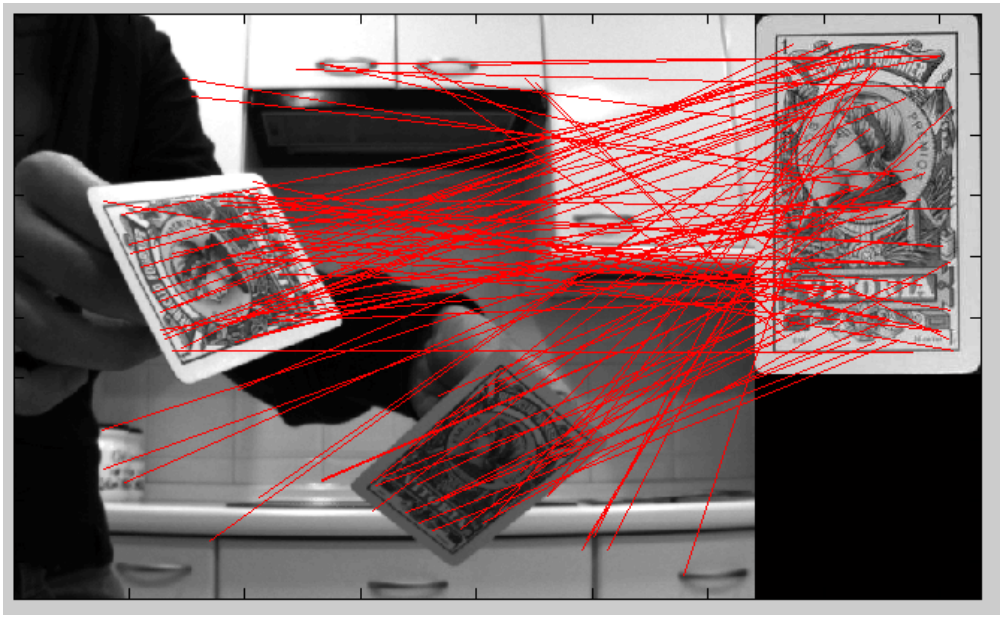


Figura 6.18: **RANSAC utilizando el modelo de cámara afín.** *En esta imagen se muestran los "outliers" tras aplicar RANSAC junto con la aproximación afín.*

Figura 6.19: **RANSAC utilizando el modelo de cámara afín.** *En esta imagen se muestra el perfil de reproyección del objeto detectado tras aplicar RANSAC junto con la aproximación afín.*

Con los dos primeros ejemplos se puede comprobar que el sistema de detección utilizando RANSAC junto con la aproximación de cámara proyectiva ofrece muy buenos resultados a la hora de detectar objetos. El sistema es capaz de detectar varios objetos en una misma imagen. También se puede apreciar que incluso si un objeto esta percialmente ocluido, el sistema es capaz de detectar el objeto. Esto se debe a la gran cantidad de descriptores SIFT que se generan en una sola imagen. De esta forma, aunque el grado de oclusión del objeto sea muy grande, el número de descriptores encontrados en la imagen que pertenecen al objeto será mayor en comparación con el número mínimo de descriptores necesarios para poder obtener una solución válida.

En el ejemplo 3, se puede apreciar como la aproximación de cámara afín no es válida cuando las deformaciones proyectivas que sufren los objetos son muy significativas. Se puede observar que el perfil estimado no se adapta a los objetos detectados, por lo que se comete un error bastante significativo en la estimación de la matriz de transformación (en la figura 6.19 se puede apreciar una de las carácterísticas significativas de las transformaciones afines: las lineas paralelas se proyectan como tales en la imagen).

En el capítulo 7 se analizará con más detalle el modelo afín y se analizará las condiciones que deben existir para garantizar que el error de aproximación es despreciable. A su vez, se hará un estudio de la influencia de ciertos factores como el ruido y las oclusiones en estos sistemas de detección.

6.3. Detección de objetos planares mediante cámara proyectiva

En el apartado anterior se ha considerado que los aobjetos al proyectarse en el plano imagen sólo sufren deformaciones afines. Esta suposición sólo es válida si la distancia entre la cámara y los objetos es grande comparada con el tamaño de los mismos.

Para el resto de los casos, si se utiliza la aproximación de cámara afín, el error que se comete en la detección es grande o incluso, el sistema es incapaz de detectar los objetos. Por tanto, hay que considerar el caso general en el cual el modelo de cámara es proyectiva y la relación entre un plano en $\mathbb{P}^3$ y su proyección en el plano imagen viene dada por una **homografía**. Ahora, la relación existente entre los puntos del patrón X^j y sus correspondientes proyecciones en la imagen será una homografía:

$$m_k^j = \lambda H \begin{pmatrix} x_k^j \\ y_k^j \\ 1 \end{pmatrix} \qquad X_k^j = \begin{pmatrix} x_k^j \\ y_k^j \\ 0 \end{pmatrix}$$

donde:

$$H = \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & 1 \end{pmatrix} \tag{6.18}$$

El problema de la obtención de la pose del objeto se complica pues la matriz H tiene 8 grados de libertad (2 mñas en comparación con el caso de aproximación afín). Al tener que calcular de el valor 2 parámetros más, los tiempos de ejecución se disparan, sobre todo para los casos en los que el número de objetos a detectar es alto (en el capílo 7 se compararán los tiempos de ejecución de cada caso).

En este apartado sólo se va a describir el proceso seguido para el cálculo de la pose utilizando RANSAC (no tiene sentido usar Hough porque el número de parámetros a calcular empieza a ser muy elevado y por las limitaciones encontradas a la hora de implementar el algoritmo en Matlab).

6.3.1. RANSAC

Al igual que pasaba en el caso afín, el punto de partida es un conjunto de $\hat{N}$ pares de puntos obtenidos en el "matching" inicial entre un patrón y una imagen. Hay que determinar cuáles de estos pares se correspon-den con "inliers" y agruparlos en el caso de que haya más de un objeto. Para ello, se va a utilizar el método RANSAC suponiendo ahora que el modelo de la cámara es proyectiva.

Aunque ahora la relación entre los puntos del objeto y sus proyecciones vendrá dada pon una homografía, el algoritmo de RANSAC que se tiene que aplicar es el mismo del apartado anterior. En concreto, se va a utilizar el Algortimo **??** que se corresponde con la modificación del algoritmo de RANSAC adaptado para múltiples objetos. Lo única diferencia estará en la definición de la función paramétrica y el valor del umbral t:

- Ahora la relación entre los puntos del objeto y los puntos poyectados viene dada por la matriz de homografía, por tanto, hay que obtener la función paramétrica que define dicha homografía.
- Ahora se calcula la relación real existente entre los puntos y sus proyecciones (sin considerar ningún tipo de aproximación), por tanto, el umbral t que permite determinar si una solución se adapta a un par de correspondencias debe ser más rectricitivo.

6.3.1.1. Definición del modelo paramétrico

La relación entre los puntos del patrón de un objeto y los "inliers" en la imagen viene dada por la matriz de homografía. Es más, supongamos que tenemos M repeticiones en la imagen del mismo objeto, cada una de ellas tendrá asociada una matriz de homografía distinta:

$$\mathbf{m}_k^i = H_i \mathbf{X}_k^i \qquad i = 1, \ldots, M \tag{6.19}$$

donde:

$$H_i = \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & 1 \end{pmatrix} \tag{6.20}$$

Para aplicar el método de RANSAC, es necesario conocer la función paramétrica $f(\mathbf{\Phi})$ que relacione las correspondencias correctas entre ellas. En este caso, el vector de parámetros $\mathbf{\Phi}$ esta compuesto por los 8 primeros elementos de la matriz H:

$$\mathbf{\Phi} = (h_{11}, h_{12}, h_{13}, h_{21}, h_{22}, h_{23}, h_{31}, h_{32})^T$$

Todo el proceso de estimación de la matriz de homografía se explicó con detalle en el apartado 3.2.3 del capítulo 3 y se obtuvo la expresión general del sistema de ecuaciones final para un conjunto de n correspondencias:

$$\begin{bmatrix} \mathbf{0}^T & -z'_1\,\mathbf{x_1}^T & y'_1\,\mathbf{x_1}^T \\ z'_1\,\mathbf{x_1}^T & \mathbf{0}^T & -x'_1\,\mathbf{x_1}^T \\ \vdots & \vdots & \vdots \\ \mathbf{0}^T & -z'_n\,\mathbf{x_n}^T & y'_n\,\mathbf{x_n}^T \\ z'_n\,\mathbf{x_n}^T & \mathbf{0}^T & -x'_n\,\mathbf{x_n}^T \end{bmatrix} \cdot \mathbf{\Phi} = \mathbf{0}$$

También se demostró en este apartado se demostró que cada par de puntos genera 2 ecuaciones linealmente independientes. Como la homografía tiene 8 grados de libertad, como mínimo se necesitan 4 puntos para obtener una solución única de la matriz H. Por tanto, para el caso general de homografía, el parámetro s del algoritmo de RANSAC debe valer 4 (ahora el algoritmo RANSAC tomará de forma aleatoria 4 pares de puntos del conjunto de correspondencias iniciales cada vez que calcule una solución para el modelo).

Al igual que pasaba en el caso de la aproximación afín, no todas las combinaciones de puntos serán válidos para obtener una solución de la matriz H. Para poder obtener una solución no degenerativa es necesario que los 4 puntos no sean colineales. Cada vez que RANSAC vaya a calcular una solución para la matriz H, va tomando de forma aletoria 4 pares de correspondencias hasta encontrar un conjunto no colineal con el que calcular una solución.

6.3.1.2. Validación y votación de cada modelo inicial

El proceso de validación es el mismo que el utilizado para el caso de afinidad. Para estimar una solución válida para la matriz H, el algoritmo de RANSAC toma ahora 4 pares de puntos de forma aleatoria y calcula la matriz H_j. Una vez que obtiene esta matriz, se comprueba qué pares de puntos de todo el conjunto de correspondencias iniciales se adapta a la solución obtenida. Para ello se va a calcular el error de reproyección para cada par de puntos:

$$\left.\begin{array}{l} d_1 = (\mathbf{m}_k - H_j\mathbf{X}_k)^2 \\ \\ d_2 = (\mathbf{X}_k - H_j^{-1}\mathbf{m}_k)^2 \end{array}\right\} \quad \longrightarrow \quad d_k^j = \sqrt{d_1 + d_2} \tag{6.21}$$

Todas las correspondencias cuya distancia sea inferior a un umbral t se consideran como "inliers" del modelo obtenido. Al igual que pasaba en el caso de aproximación afín, cada matriz H_j obtenida con RANSAC recibirá una puntuación en función del numero de "inliers". De entre todos los candidatos, se toma como solución aquella matriz H_j con

mayor puntuación y se vuelve a recalcular su valor teniendo en cuenta todos los “inliers” para obtener una solución más precisa.

La única diferencia respecto al método de detección utilizando la aproximación de cámara afín es que el valor del umbral t es más pequeño. Ahora, al suponer que el modelo de la cámara es proyectiva, cada solución que se calcula con RANSAC debe ser la transformación real que sufre el objeto al ser proyectado en el plano imagen, sin considerar ningún tipo de aproximación. Por tanto, independientemente del grado de deformación (ya sea proyectiva de mayor o menor grado o coincida con una afinidad) si la solución es correcta, ésta deberá adaptarse muy bien a los puntos pertenecientes al objeto. Por tanto, el umbral que se va a utilizar va a ser más restrictivo.

6.3.1.3. Resultados

A continuación se van a mostrar algunos ejemplos en los que se han detectado varios objetos utilizando el algoritmo de RANSAC junto con la suposición de cámara proyectiva. Los resultados que se van a visualizar son los mismos que para el caso de afinidad:

- **Primer ejemplo**.

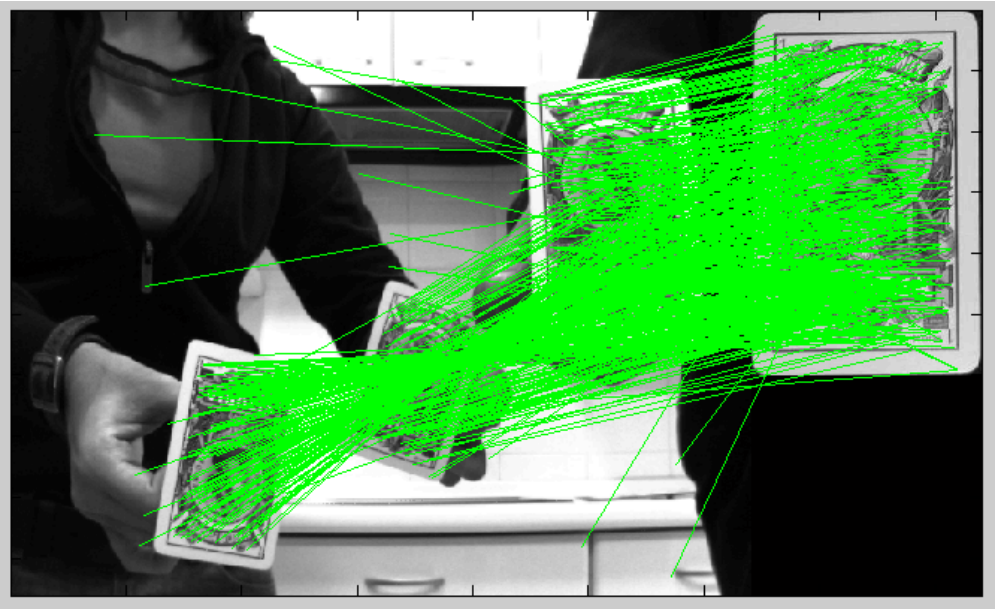


Figura 6.20: **RANSAC utilizando el modelo de cámara proyectiva.** *En esta imagen se muestra el “matching” inicial obtenido por SIFT. Cada par de correspondencias entre un punto del patrón y la imagen se representa con una recta de color verde* .

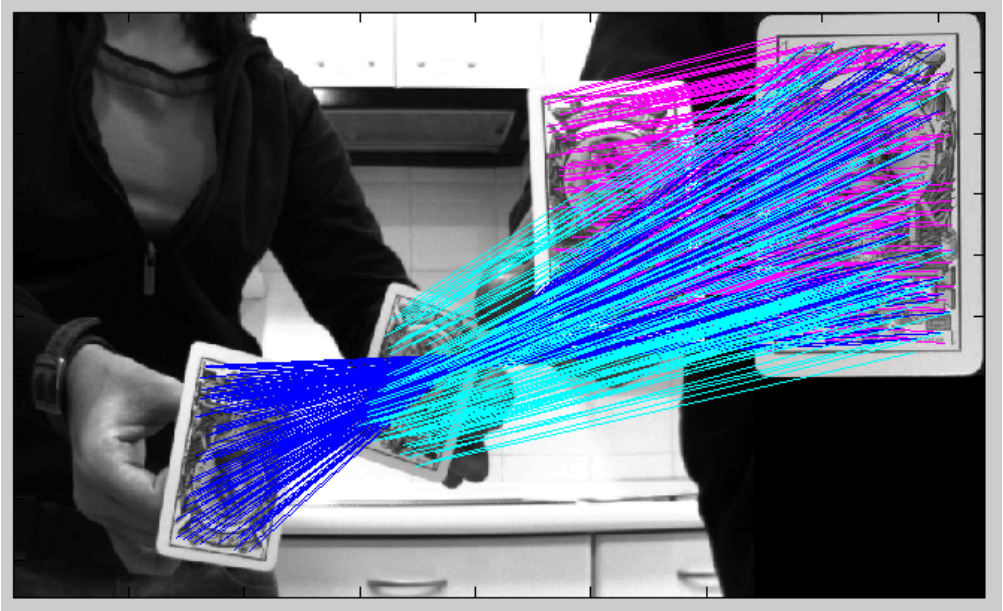


Figura 6.21: **RANSAC utilizando el modelo de cámara proyectiva.** *En esta imagen se muestran los "inliers" tras aplicar RANSAC junto con el modelo de cámara proyectiva.*

Figura 6.22: **RANSAC utilizando el modelo de cámara proyectiva.** *En esta imagen se muestran los "outliers" tras aplicar RANSAC junto con el modelo de cámara proyectiva.*

Figura 6.23: **RANSAC utilizando el modelo de cámara proyectiva.** *En esta imagen se muestra el perfil de reproyección del objeto detectado tras aplicar RANSAC junto con el modelo de cámara proyectiva.*

- **Segundo ejemplo**.

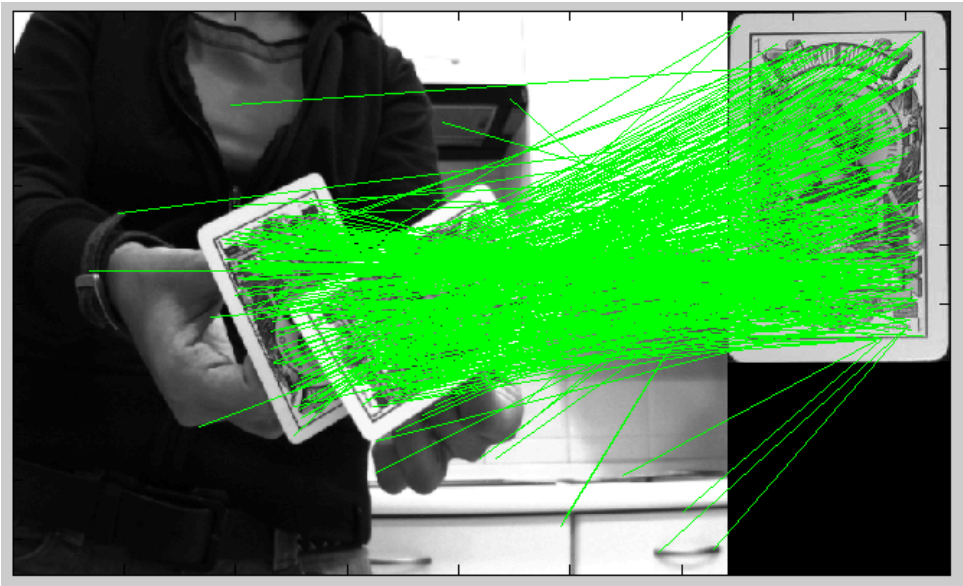


Figura 6.24: **RANSAC utilizando el modelo de cámara proyectiva.** *En esta imagen se muestra el "matching" inicial obtenido por SIFT. Cada par de correspondencias entre un punto del patrón y la imagen se representa con una recta de color verde .*

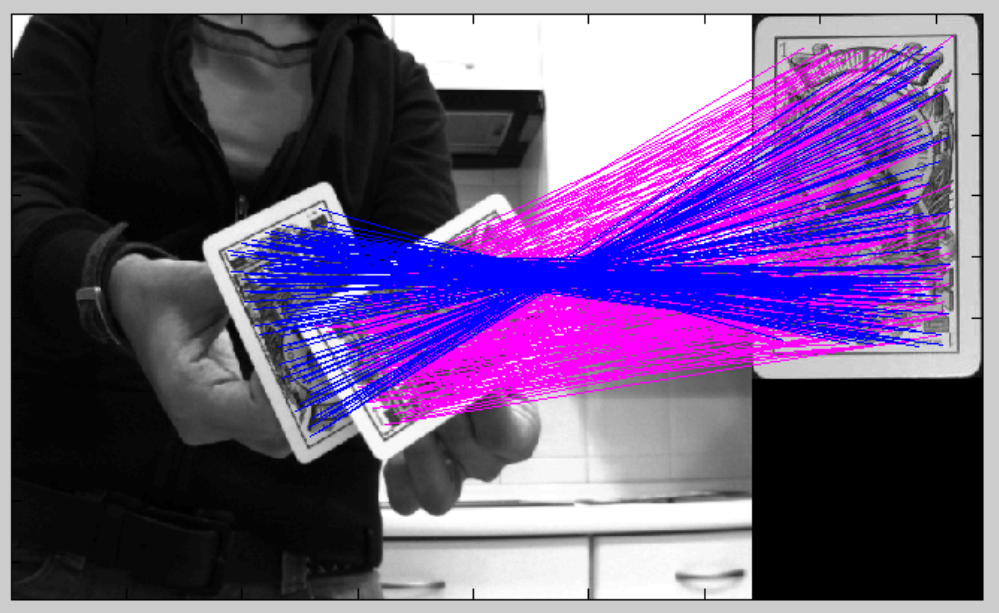


Figura 6.25: **RANSAC utilizando el modelo de cámara proyectiva.** *En esta imagen se muestran los "inliers" tras aplicar RANSAC junto con el modelo de cámara proyectiva.*

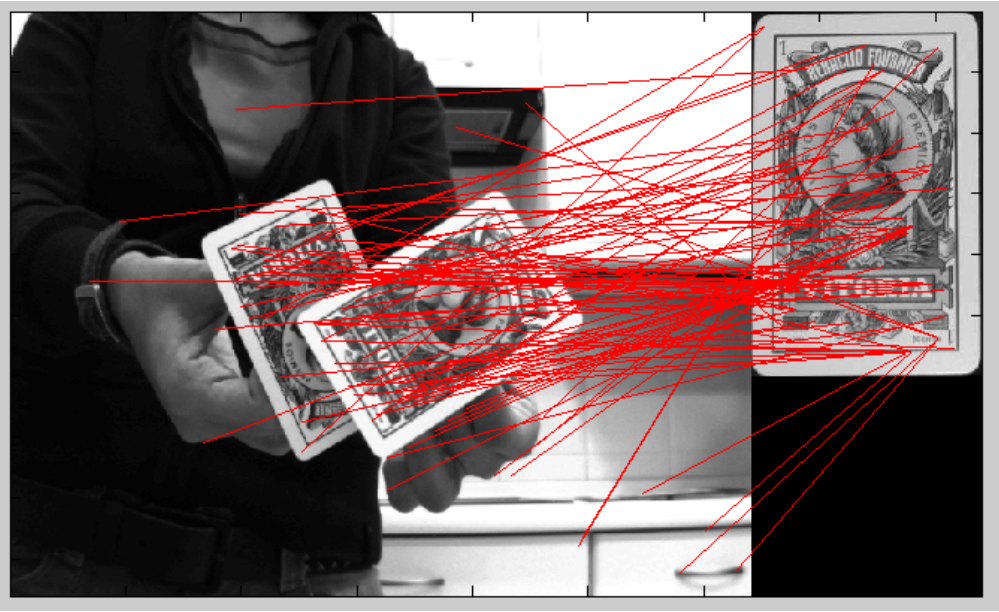


Figura 6.26: **RANSAC utilizando el modelo de cámara proyectiva.** *En esta imagen se muestran los "outliers" tras aplicar RANSAC junto con el modelo de cámara proyectiva.*

Figura 6.27: **RANSAC utilizando el modelo de cámara proyectiva.** *En esta imagen se muestra el perfil de reproyección del objeto detectado tras aplicar RANSAC junto con el modelo de cámara proyectiva.*

- **Tercer ejemplo**.

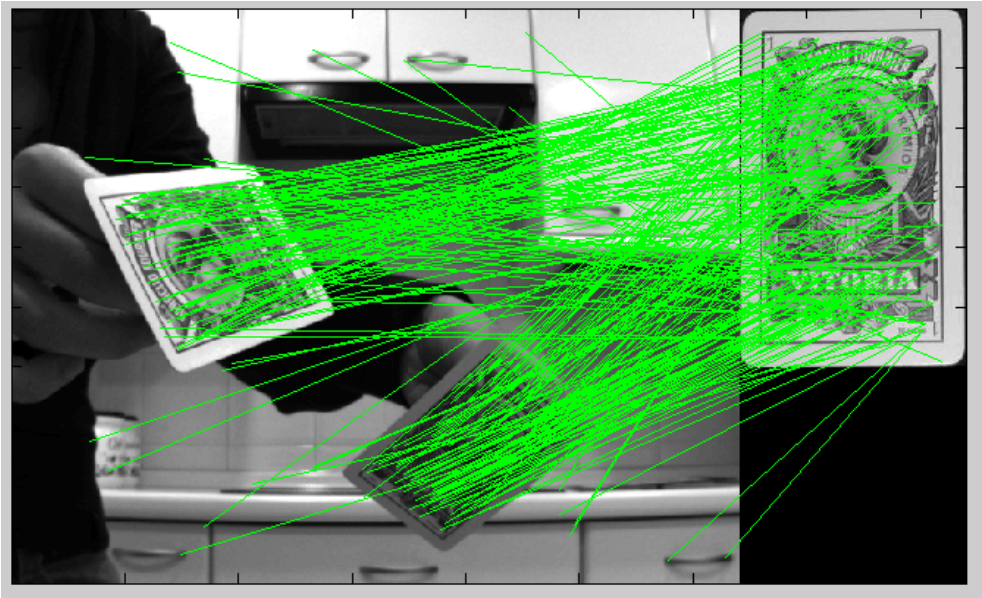


Figura 6.28: **RANSAC utilizando el modelo de cámara proyectiva.** *En esta imagen se muestra el "matching" inicial obtenido por SIFT. Cada par de correspondencias entre un punto del patrón y la imagen se representa con una recta de color verde .*

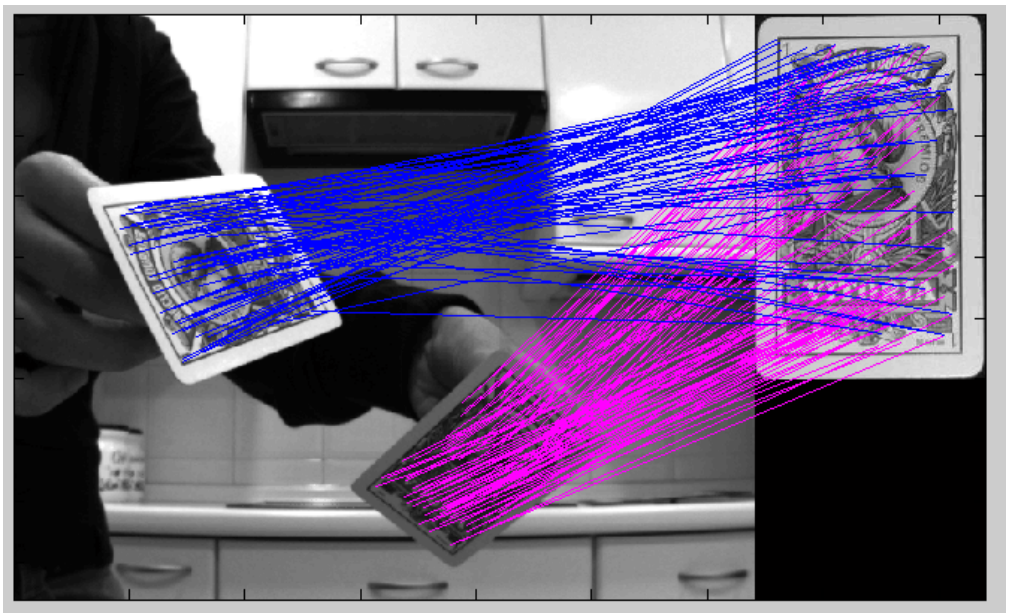


Figura 6.29: **RANSAC utilizando el modelo de cámara proyectiva.** *En esta imagen se muestran los "inliers" tras aplicar RANSAC junto con el modelo de cámara proyectiva.*

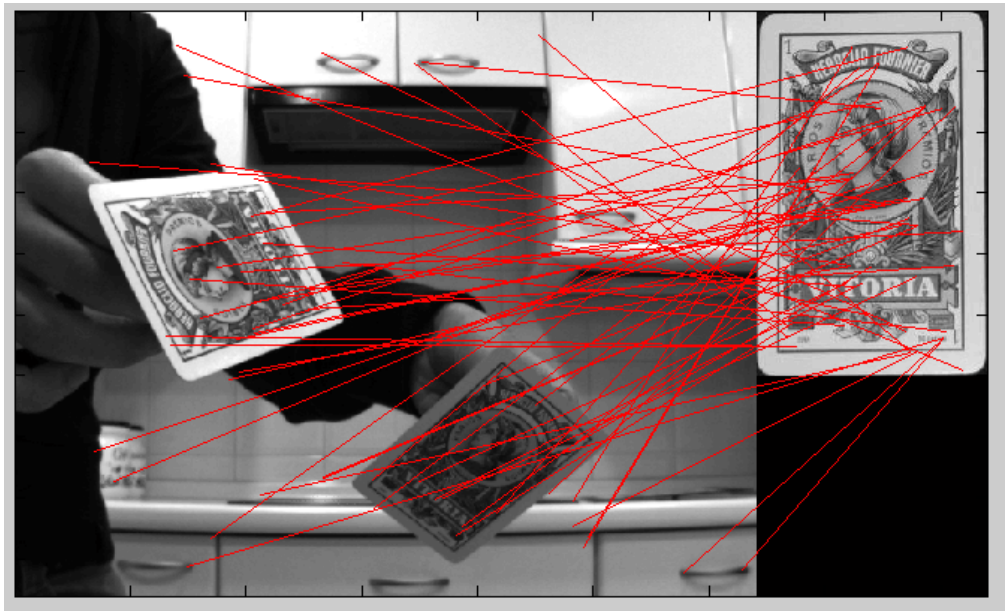


Figura 6.30: **RANSAC utilizando el modelo de cámara proyectiva.** *En esta imagen se muestran los "outliers" tras aplicar RANSAC junto con el modelo de cámara proyectiva.*

Figura 6.31: **RANSAC utilizando el modelo de cámara proyectiva.** *En esta imagen se muestra el perfil de reproyección del objeto detectado tras aplicar RANSAC junto con el modelo de cámara proyectiva.*

Con estos ejemplos se puede comprobar que el sistema de detección utilizando RANSAC junto con la suposición de cámara proyectiva ofrece muy buenos resultados a la hora de detectar objetos. El sistema es capaz de detectar varios objetos en una misma imagen incluso si estos objetos están parcielmente ocultos.

Una diferencia clara que se puede apreciar en el último ejemplo es que ahora el perfil se adapta perfectamente al objeto (no como pasaba en el modelo afín). Esto se debe a que ahora no se está teniendo en cuenta ningún tipo de aproximación a la hora de calcular la matriz de transformación y por tanto, los resultados que se obtienen son una estimación de dicha transformación real.

En el capítulo 7 se analizará con más detalle las diferencias entre el modelo afín y el proyectivo. A su vez, se hará un estudio de la influencia de ciertos factores como el ruido y las oclusiones en estos sistemas de detección.

Capítulo 7

Resultados y Simulaciones

En el capítulo 6 se describieron los tres métodos propuestos para detectar los “inliers” presentes en una imagen y así resolver el problema de la detección de objetos planares:

1. **Detección mediante la Transformada de Hough suponiendo que el modelo de la cámara es afín.**
2. **Detección mediante el algoritmo RANSAC y la aproximación de cámara afín.**
3. **Detección mediante el algoritmo RANSAC para el caso general de cámara proyectiva.**

En este capítulo se va a hacer una descripción de todas las simulaciones que se han realizado para validar estos métodos y analizar el comportamiento de los mismos frente a variaciones de ciertos factores como el ruido y la escala.

Respecto al modelo de detección utilizando la Transformada de Hough, no se han obtenido buenos resultados, tal y como se comentó en el capítulo 6. Por tanto, se ha utilizado únicamente RANSAC en todos las simulaciones que aparecen en este capítulo.

MODELO 3D.

7.1. Detección de objetos planares

En los capítulos anteriores se han propuesto 3 métodos para resolver el problema de la detección de objetos planares. De estos 3 métodos, sólo se han obtenido buenos resultados de dos de ellos, en concreto de los métodos basados en RANSAC.

En este apartado, se va a describir las simulaciones que se han realizado para comprobar el funcionamiento de los dos métodos de detección basados en RANSAC. Además de validar ambos métodos, hay que analizar su comportamiento y el error de estimación frente a variaciones de ciertos factores como el ruido, las oclusiones, cambios de escala, etc. Las simulaciones realizadas son las siguientes:

1. **Ejemplos generales**

2. **Estudio del error en función de la distancia** - En capítulos anteriores se explicó que el método de detección utilizando RANSAC junto con la aproximación de cámara afín sólo es válida si la distancia entre los objetos y la cámara es grande en comparación con el tamaño de los objetos. En esta simulación se va a evaluar y comparar el error de estimación de ambos modelos en función de la distancia. El objetivo de esta simulación es determinar bajo que condiciones se puede utilizar la aproximación afín por ser el error despreciable y además, bajo estas condiciones, evaluar si el uso de dicha aproximación es más eficiente en comparación con el caso general.

3. **Estudio del error en función del grado de oclusión** - Se va a evaluar el efecto que produce en la detección el hecho de que el objeto aparezca en la imagen más o menos oculto y cómo influye dicho grado de oclusión en el error de estimación de la matriz H.

4. **Estudio del error en función de la escala y la deformación proyectiva** - Otros dos factores que influyen en el proceso de detección son la escala y la deformación que sufre el objeto al ser proyectado en la imagen. Por tanto, se va a estudiar los efectos en la detección y los errores de estimación en función de estos dos parámetros.

5. **Estudio del error en función del ruido** - Otro factor importante a tener en cuenta es ruido. Se va a analizar los efectos del ruido en el proceso de detección y los errores de estimación en función del mismo.

6. **Estudio de detección para múltiples objetos** - El l algoritmo de RANSAC proporciona muy buenos resultados en la detección de un único objeto, incluso cuando el porcentaje de “outliers” es muy elevado. Sin embargo, el objetivo final de este proyecto es conseguir un sistema que sea capaz de detectar todos los objetos repetidos en la imagen. Por tanto, hay que comprobar cómo se comporta el algoritmo de RANSAC adaptado para múltiples objetos que se propuso en el capítulo **??**.

7.1.1. Ejemplos generales

En este primer apartado se van a mostrar algunos ejemplos de detección utilizando imágenes reales para dar una idea general del funcionamiento completo del sistema. En cada ejemplo, se va a aplicar RANSAC utilizando la aproximación de cámara afín y el modelo de cámara proyectiva.

Los pasos que siempre realiza el sistema (tanto para el modelo afín como para el proyectivo) son los siguientes:

1. Extracción y localización de los descriptores, tanto de la imagen a analizar como del patrón.
2. Proceso de "matching" inicial entre los descriptores de ambas imágenes.
3. Aplicando RANSAC al conjunto de correspondencias iniciales se detectan los objetos de la imagen y se obtienen sus matrices de transformación junto con los "inliers" asociados a cada objeto.

En todos estos ejemplos, se han utilizado imágenes reales. En concreto, el objeto a detectar es un naipe de la baraja española. Los resultados que se van a visualizar son las siguientes:

- **"Matching" inicial** - Se va a mostrar la imagen y el patrón en la misma figura. Además, se va a trazar una línea de color verde entre cada par de correspondencias encontradas con el método SIFT.
- **"Inliers"** - En esta gráfica se muestran los "inliers" detectados por RANSAC. En el caso de haber más de un objeto, se ha utilizado diferentes colores para representar las líneas que unen los puntos del patrón con los de los de la imagen (cada objeto tendrá un color diferente).
- **"Outliers"** - Se muestra el conjunto de "outliers" resultantes tras aplicar RANSAC.
- **"Reproyección del perfil del objeto** - Para cada objeto detectado, RANSAC no sólo devuelve los "inliers" asociados a él, sino que también se obtiene la matriz de transformación sufrida por el objeto patrón. Esta imagen permite ver claramente si la estimación de la matriz H es buena, o si por el contrario, el error que se ha cometido es alto.

- **RANSAC y modelo de cámara afín**

- **Primer ejemplo**.

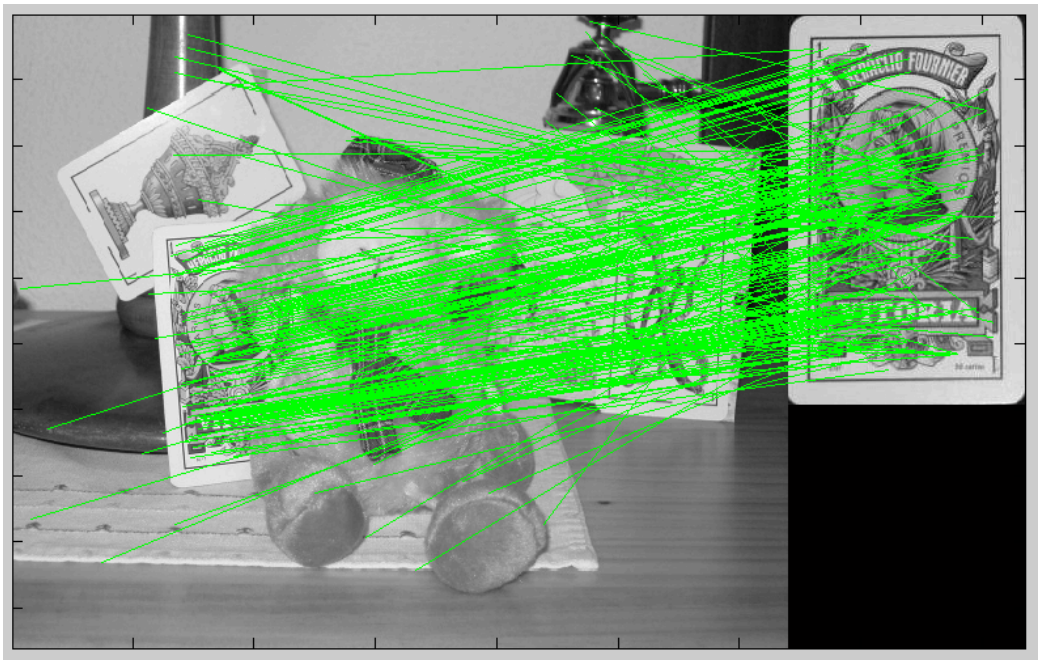


Figura 7.1: **Resultados y simulaciones.** *En esta imagen se muestra el "matching" inicial obtenido por SIFT. Cada par de correspondencias entre un punto del patrón y la imagen se representa con una recta de color verde .*

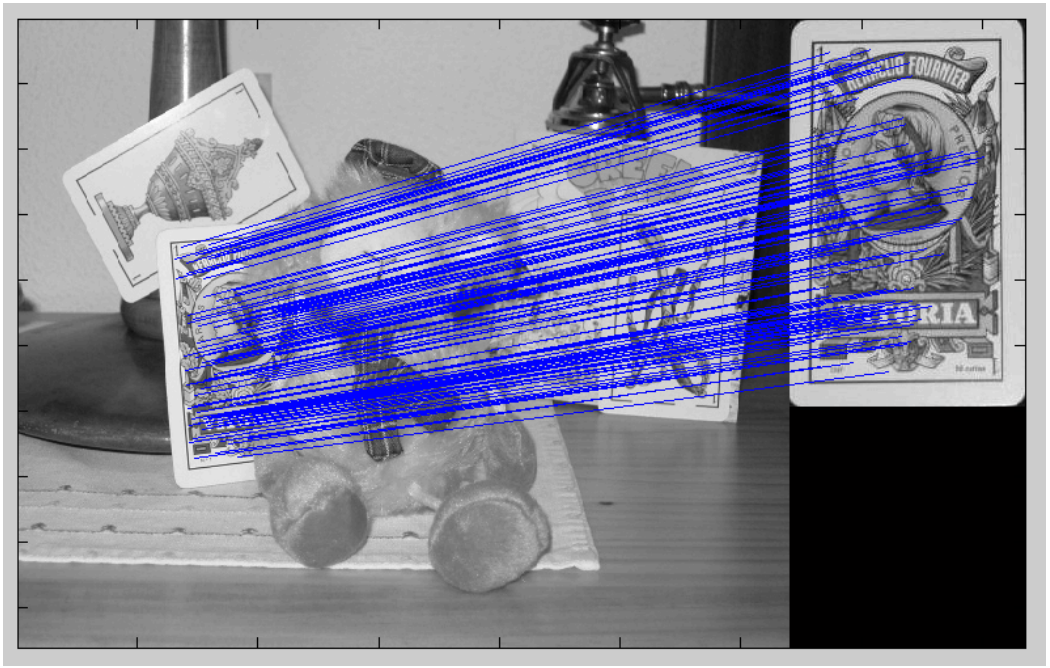


Figura 7.2: **Resultados y simulaciones.** *En esta imagen se muestran los "inliers" tras aplicar RANSAC junto con la aproximación afín.*

Figura 7.3: **Resultados y simulaciones.** *En esta imagen se muestran los "outliers" tras aplicar RANSAC junto con la aproximación afín.*

Figura 7.4: **Resultados y simulaciones.** *En esta imagen se muestra el perfil de reproyección del objeto detectado tras aplicar RANSAC junto con la aproximación afín.*

- **Segundo ejemplo**.

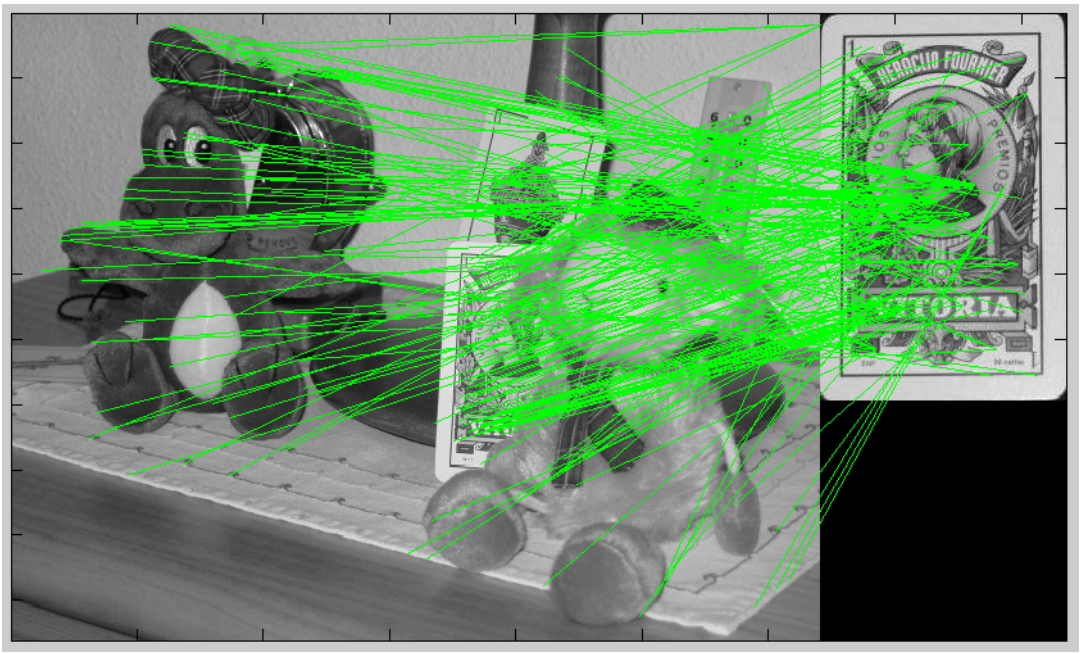


Figura 7.5: **Resultados y simulaciones.** *En esta imagen se muestra el "matching" inicial obtenido por SIFT. Cada par de correspondencias entre un punto del patrón y la imagen se representa con una recta de color verde* .

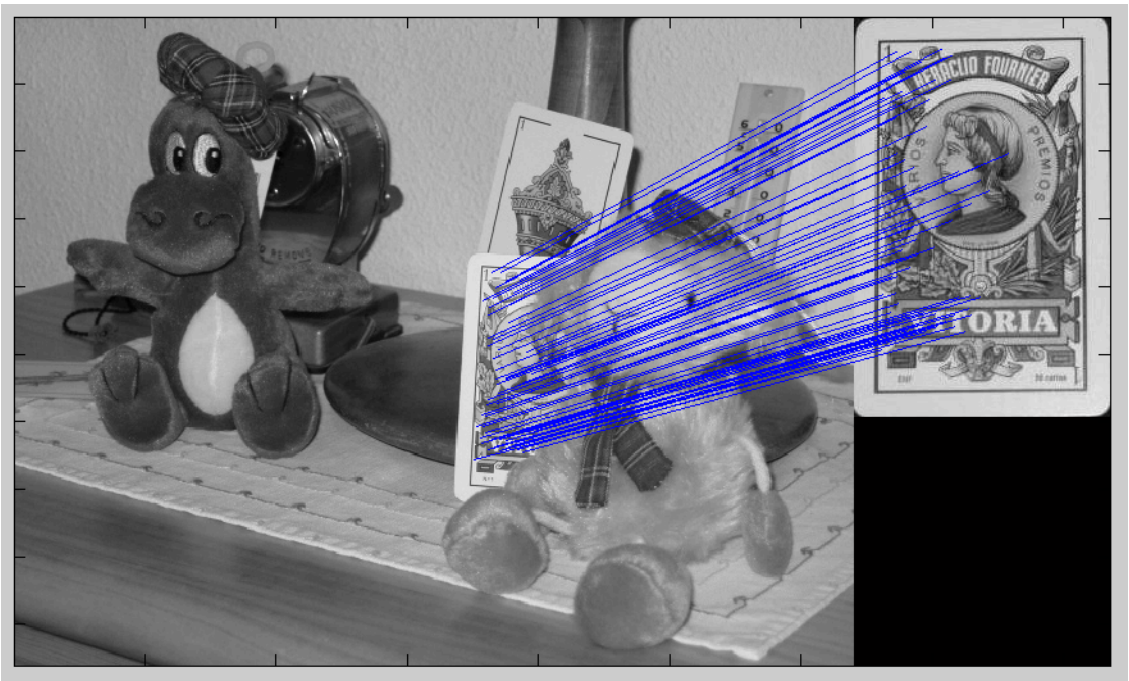


Figura 7.6: **Resultados y simulaciones.** *En esta imagen se muestran los "inliers" tras aplicar RANSAC junto con la aproximación afín.*

Figura 7.7: **Resultados y simulaciones.** *En esta imagen se muestran los "outliers" tras aplicar RANSAC junto con la aproximación afín.*

Figura 7.8: **Resultados y simulaciones.** *En esta imagen se muestra el perfil de reproyección del objeto detectado tras aplicar RANSAC junto con la aproximación afín.*

- **Tercer ejemplo.**

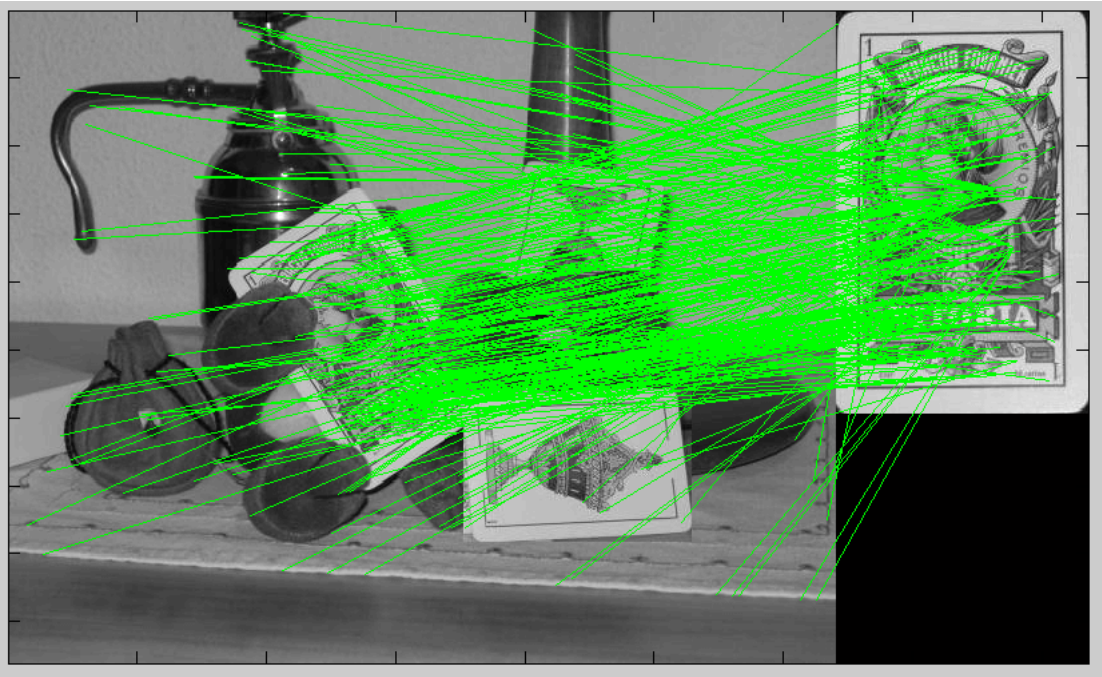


Figura 7.9: **Resultados y simulaciones.** *En esta imagen se muestra el "matching" inicial obtenido por SIFT. Cada par de correspondencias entre un punto del patrón y la imagen se representa con una recta de color verde .*

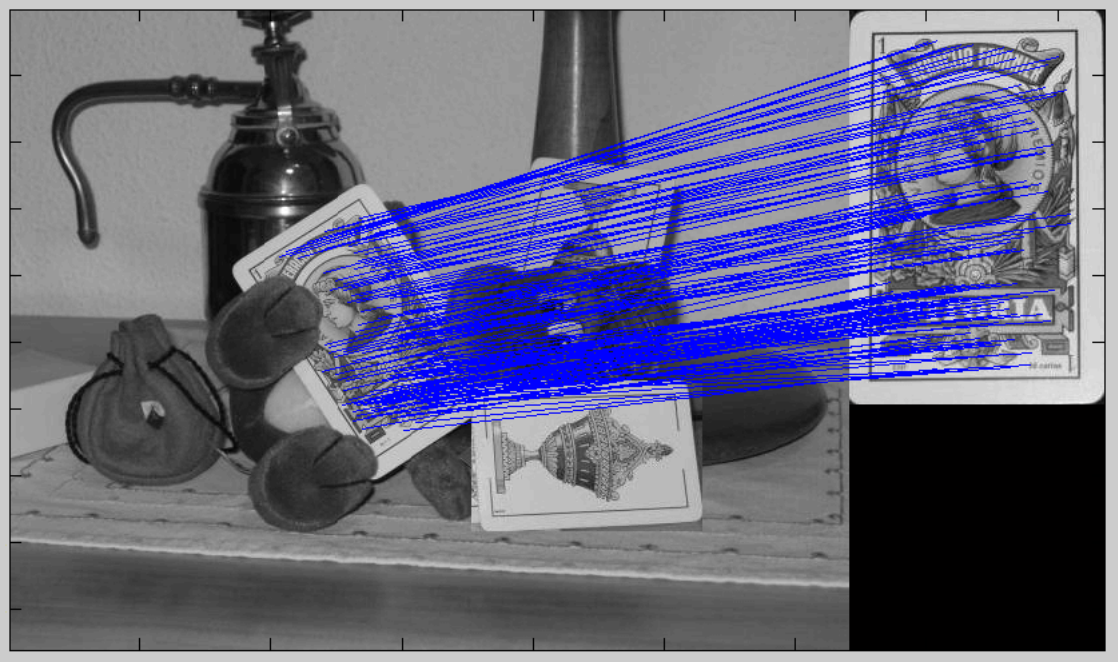


Figura 7.10: **Resultados y simulaciones.** *En esta imagen se muestran los "inliers" tras aplicar RANSAC junto con la aproximación afín.*

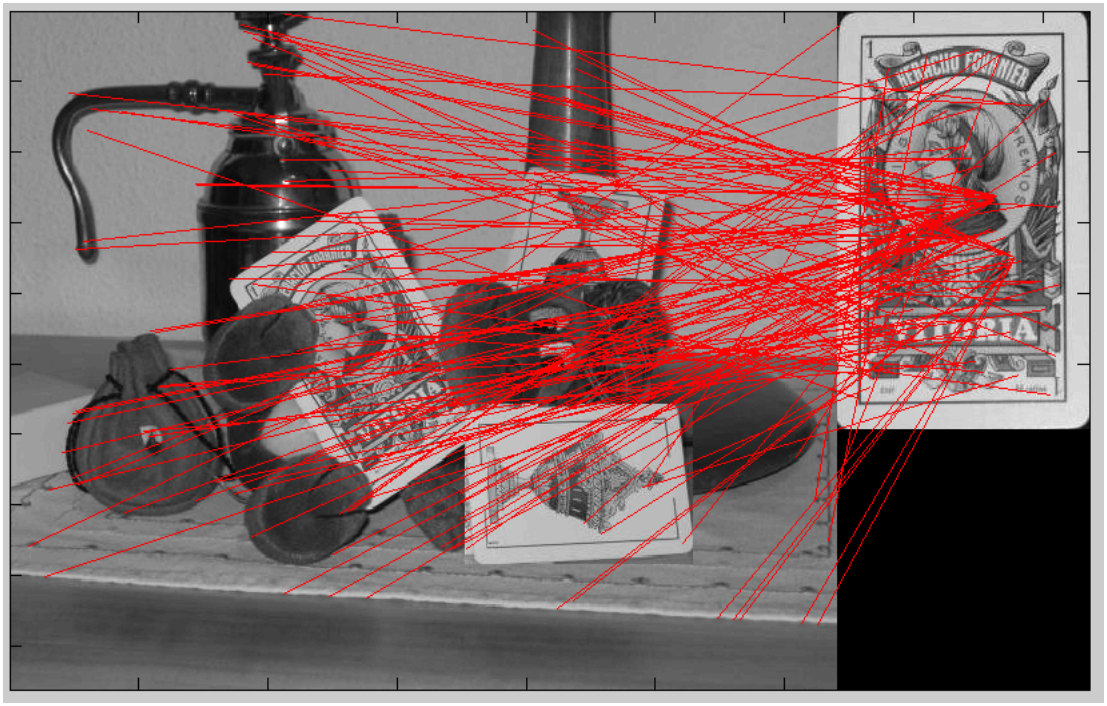


Figura 7.11: **Resultados y simulaciones.** *En esta imagen se muestran los "outliers" tras aplicar RANSAC junto con la aproximación afín.*

Figura 7.12: **Resultados y simulaciones.** *En esta imagen se muestra el perfil de reproyección del objeto detectado tras aplicar RANSAC junto con la aproximación afín.*

- **Cuarto ejemplo.**

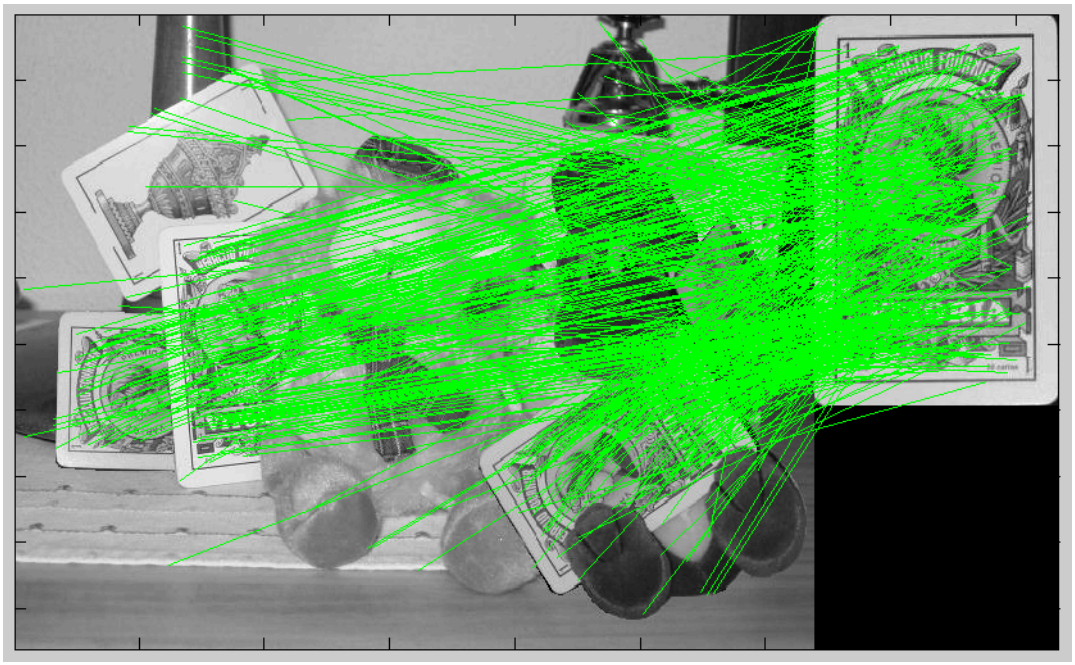


Figura 7.13: **Resultados y simulaciones.** *En esta imagen se muestra el "matching" inicial obtenido por SIFT. Cada par de correspondencias entre un punto del patrón y la imagen se representa con una recta de color verde .*

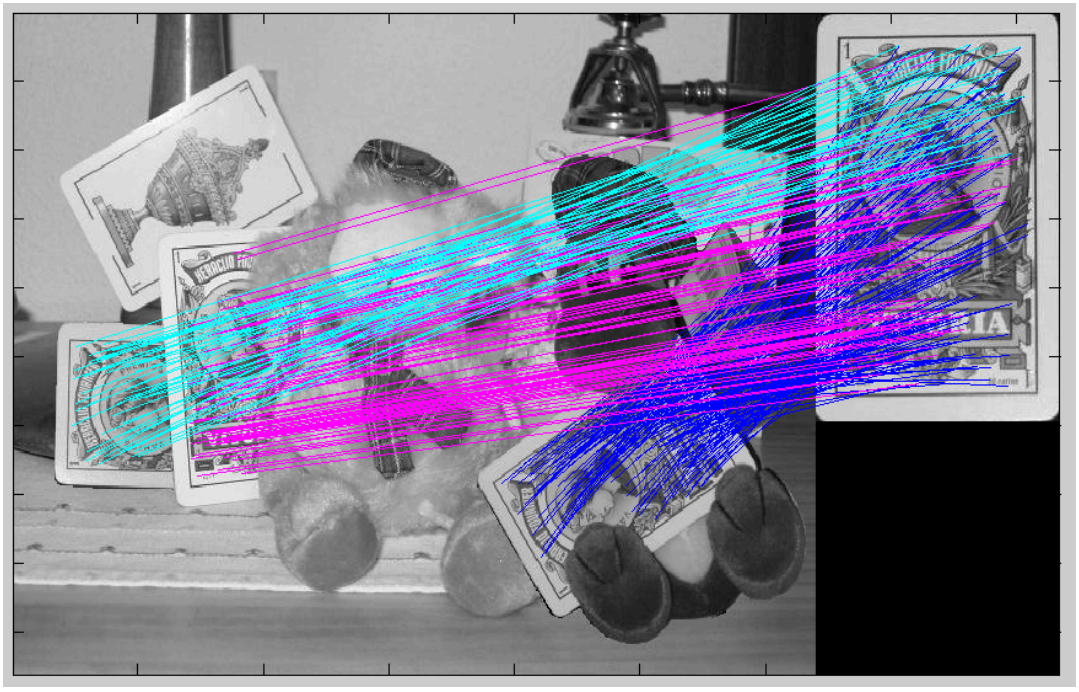


Figura 7.14: **Resultados y simulaciones.** *En esta imagen se muestran los "inliers" tras aplicar RANSAC junto con la aproximación afín.*

Figura 7.15: **Resultados y simulaciones.** *En esta imagen se muestran los "outliers" tras aplicar RANSAC junto con la aproximación afín.*

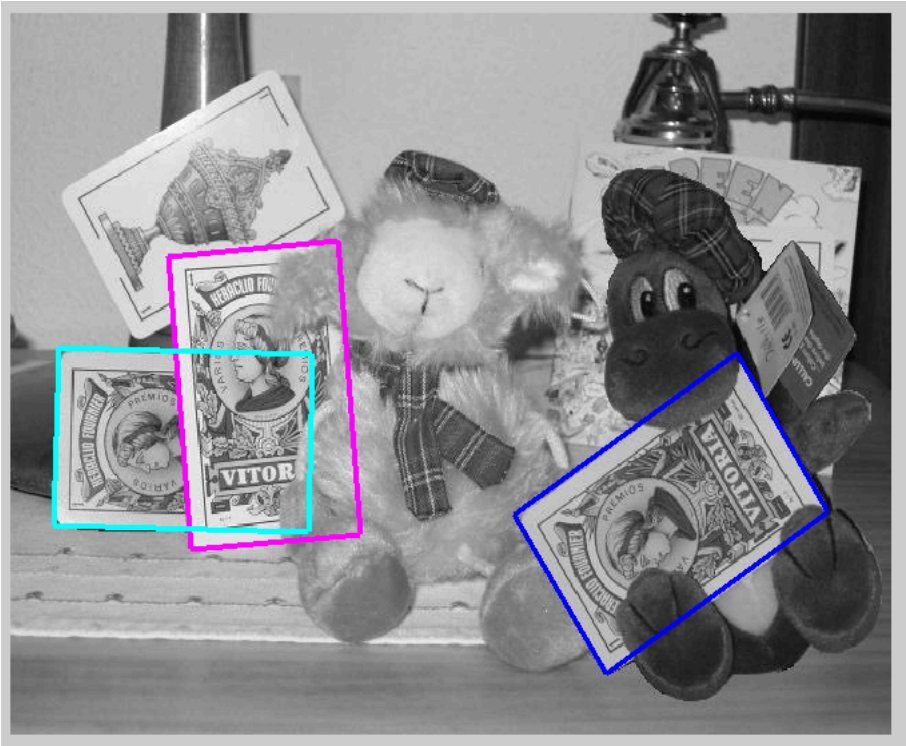


Figura 7.16: **Resultados y simulaciones.** *En esta imagen se muestra el perfil de reproyección del objeto detectado tras aplicar RANSAC junto con la aproximación afín.*

En la siguiente tabla, se muestra los resultados numéricos obtenidos en estos ejemplos:

	Ejemplo 1	Ejemplo 2	Ejemplo 3	Ejemplo4
Número de objetos	1	1	1	3
Descriptores en la imagen	1474	1297	1344	1968
Descriptores del patrón	842	842	842	842
"Matching" inicial	229	207	300	489
Número de "Inliers"	127	71	157	166, 96 y 43
Número de "Outliers"	102	136	143	184

Una de las ventajas que presentan el método SIFT es el alto número de descriptores que se pueden encontrar en una imagen. En estos ejemplos, se puede apreciar que el número total de descriptores, tanto en el patrón como en la imagen es muy alto. Una consecuencia directa de esto es que el sistema es capaz de detectar objetos incluso cuando estos se encuentran parcialmente ocultos.

Para calcular una solución de la matriz de transformación únicamente hace falta conocer 3 correspondencias entre el patrón y el objeto en la imagen para el caso de la aproximación afín. Debido a la gran concentración de descriptores que puede tener un objeto (por ejemplo, en una área tan reducida como puede ser el naipe se encuentran hasta 842 descriptores) es muy probable encontrar en la imagen al menos 3 correspondencias incluso cuando el objeto está prácticamente oculto (suponiendo, claro está, que la distribución de descriptores por el objeto es más o menos uniforme) En el cuarto ejemplo, se puede observar que sólo la mitad de uno de los naipes está visible y aun así el número de "inliers" detectados es muy alto en comparación con el mínimo requerido para obtener una solución de la matriz H, por lo que sistema es capaz de detectar el objeto. Más adelante se hará un estudio más detallado de los efectos de la oclusión en la detección.

Si nos fijamos en las gráficas donde se ha reproyectado el perfil del naipe utilizando las matrices estimadas por RANSAC se puede apreciar que los resultados son muy buenos pues el perfil se adapta completamente a los objetos de la imagen. Por las condiciones en las que fueron tomadas las imágenes, no se puede apreciar los errores de estimación de la matriz H cuando la deformación proyectiva no es despreciable. Posteriormente se hará un estudio de las condiciones en las que se puede aplicar la aproximación afín con ejemplos aclaratorios.

- **RANSAC y modelo de cámara proyectiva**

- **Primer ejemplo**.

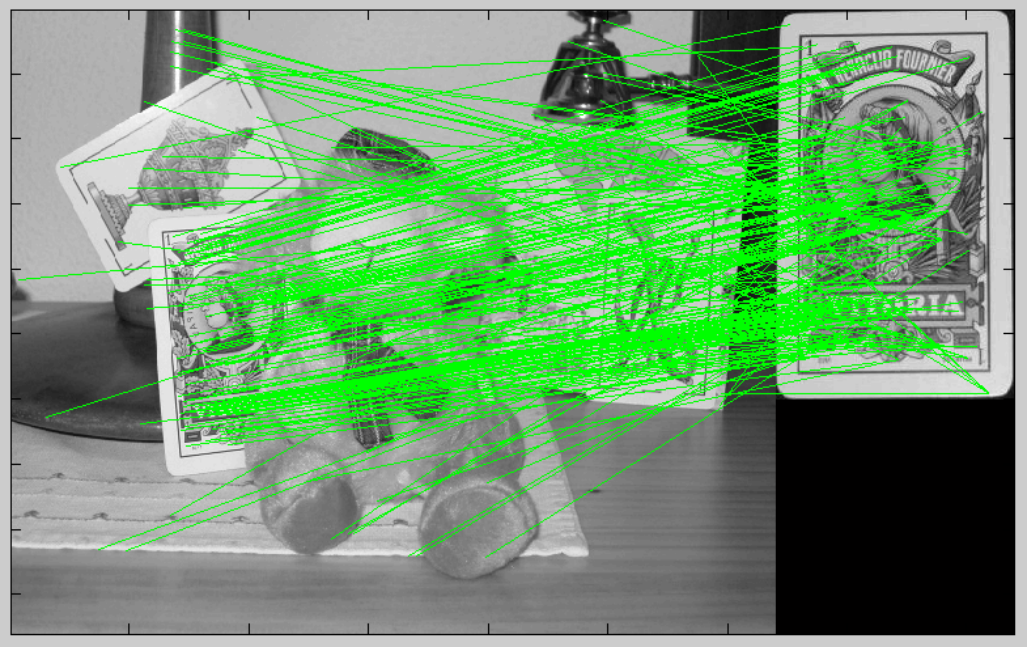


Figura 7.17: **Resultados y simulaciones.** *En esta imagen se muestra el "matching" inicial obtenido por SIFT. Cada par de correspondencias entre un punto del patrón y la imagen se representa con una recta de color verde* .

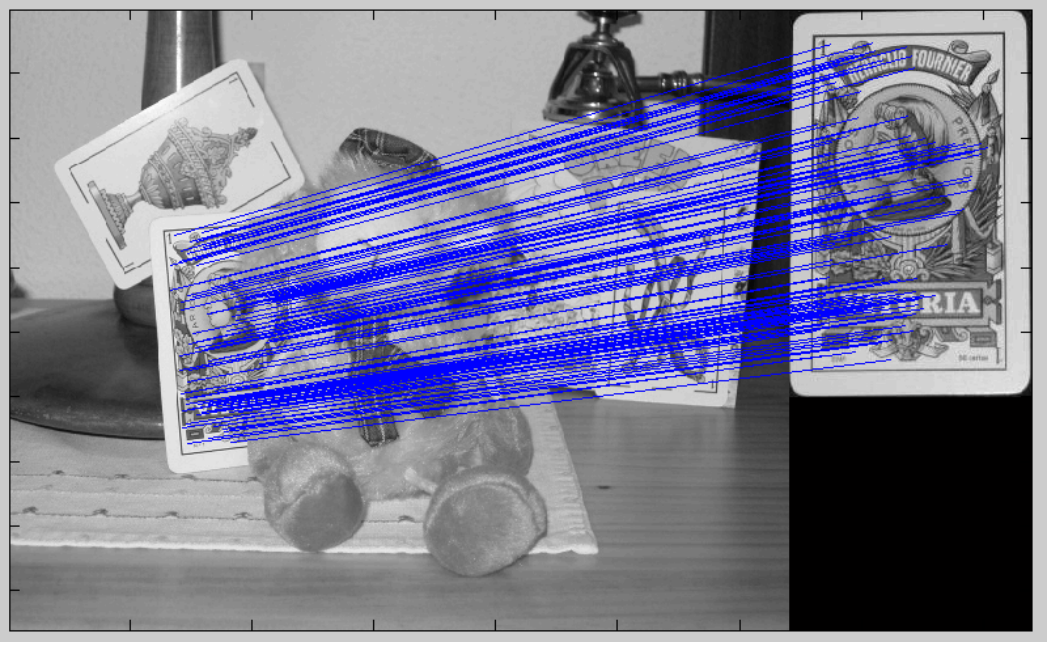


Figura 7.18: **Resultados y simulaciones.** *En esta imagen se muestran los "inliers" tras aplicar RANSAC junto con la aproximación afín.*

Figura 7.19: **Resultados y simulaciones.** *En esta imagen se muestran los "outliers" tras aplicar RANSAC junto con la aproximación afín.*

Figura 7.20: **Resultados y simulaciones.** *En esta imagen se muestra el perfil de reproyección del objeto detectado tras aplicar RANSAC junto con la aproximación afín.*

- **Segundo ejemplo**.

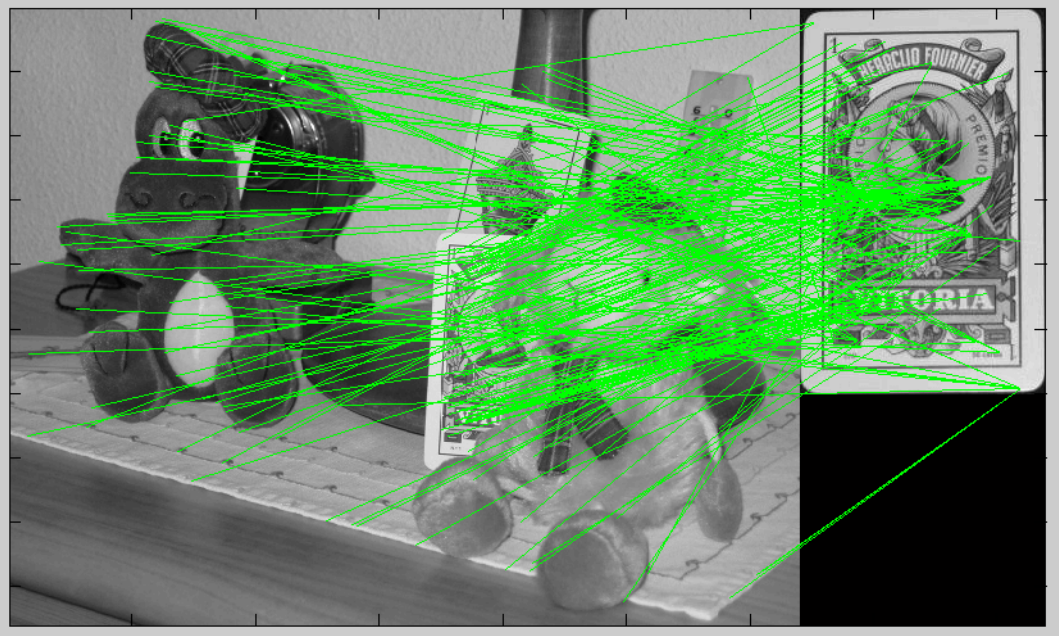


Figura 7.21: **Resultados y simulaciones.** *En esta imagen se muestra el "matching" inicial obtenido por SIFT. Cada par de correspondencias entre un punto del patrón y la imagen se representa con una recta de color verde* .

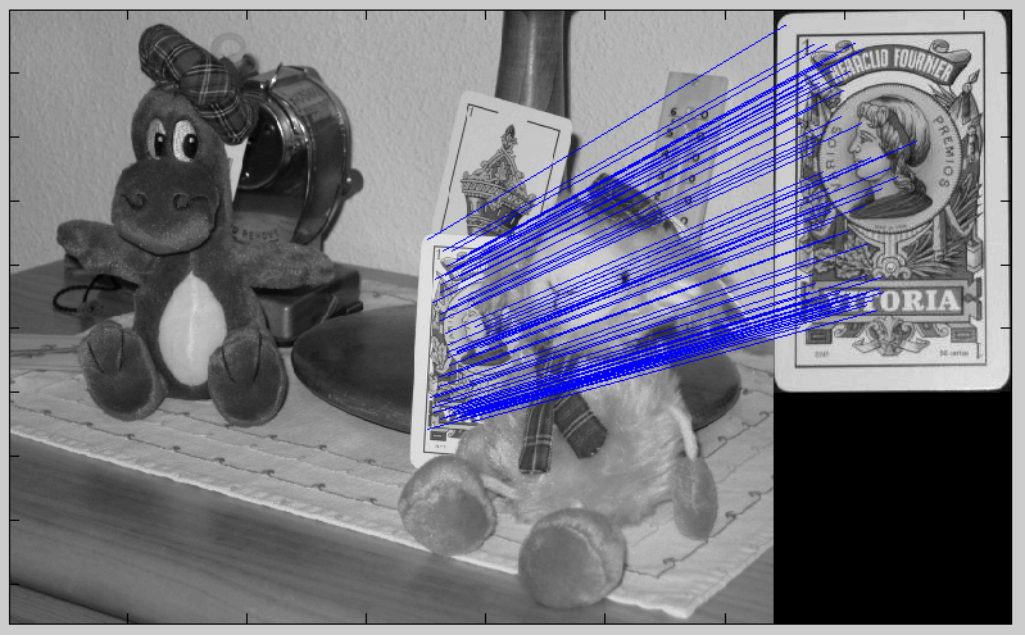


Figura 7.22: **Resultados y simulaciones.** *En esta imagen se muestran los "inliers" tras aplicar RANSAC junto con la aproximación afín.*

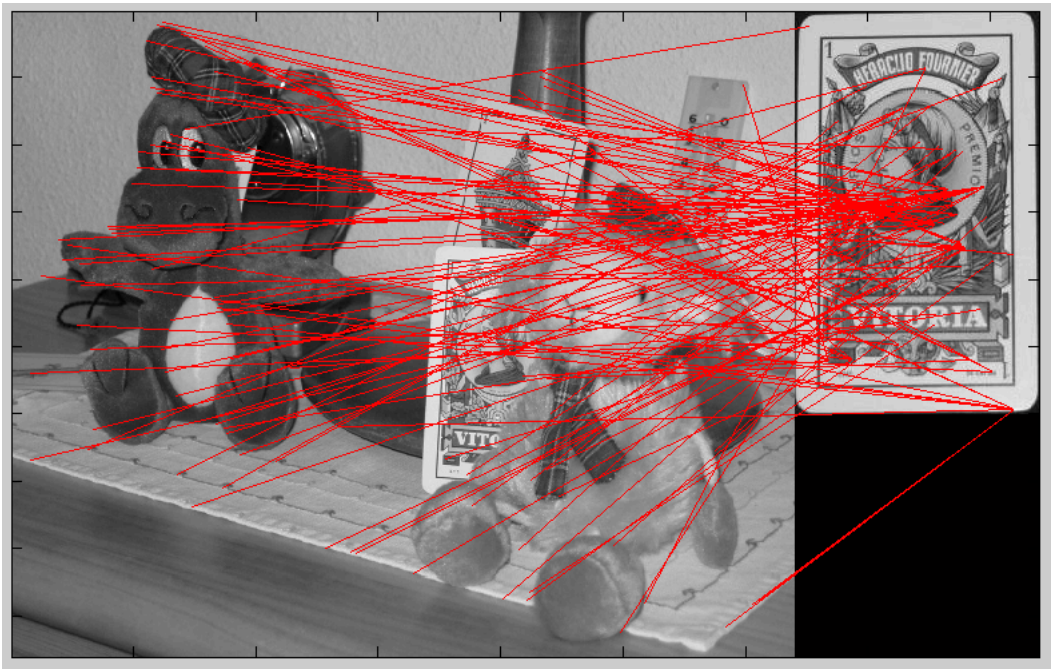


Figura 7.23: **Resultados y simulaciones.** *En esta imagen se muestran los "outliers" tras aplicar RANSAC junto con la aproximación afín.*

Figura 7.24: **Resultados y simulaciones.** *En esta imagen se muestra el perfil de reproyección del objeto detectado tras aplicar RANSAC junto con la aproximación afín.*

- **Tercer ejemplo.**

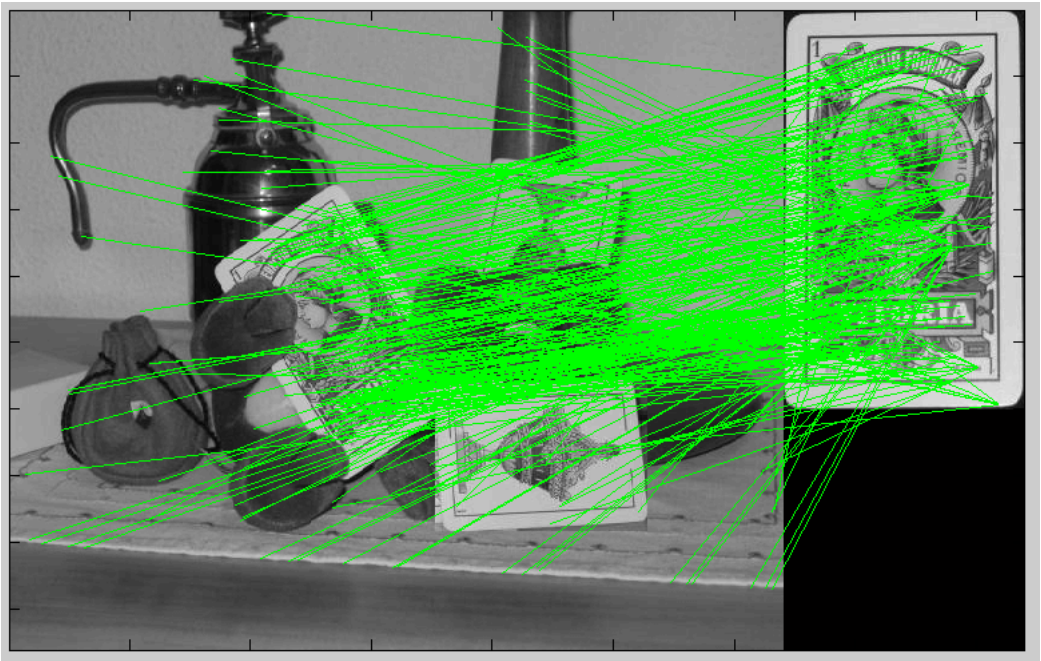


Figura 7.25: **Resultados y simulaciones.** *En esta imagen se muestra el "matching" inicial obtenido por SIFT. Cada par de correspondencias entre un punto del patrón y la imagen se representa con una recta de color verde .*

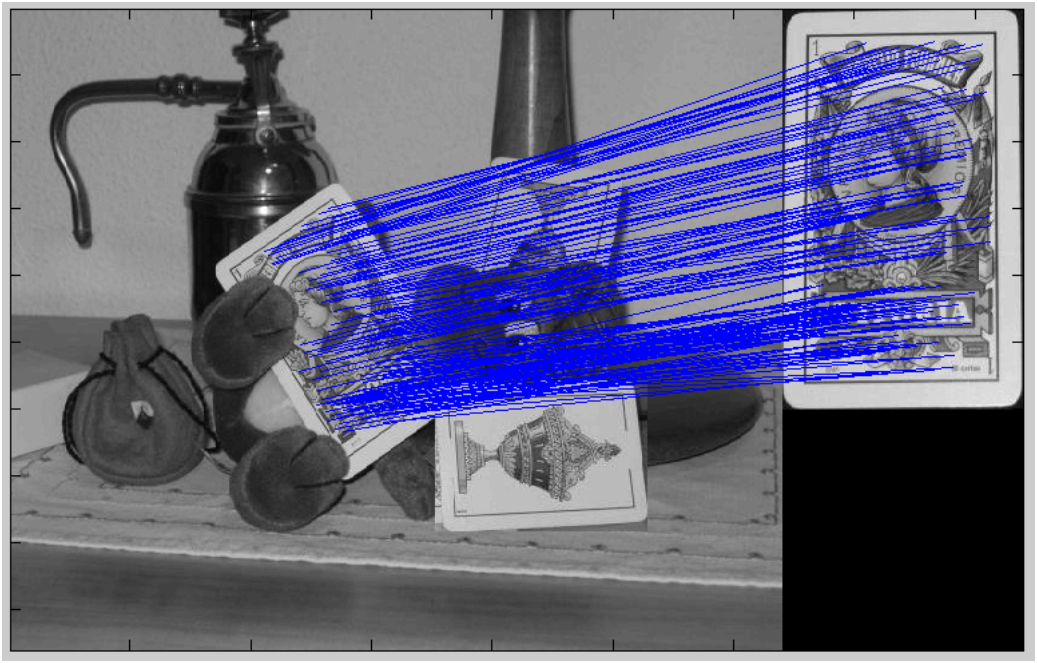


Figura 7.26: **Resultados y simulaciones.** *En esta imagen se muestran los "inliers" tras aplicar RANSAC junto con la aproximación afín.*

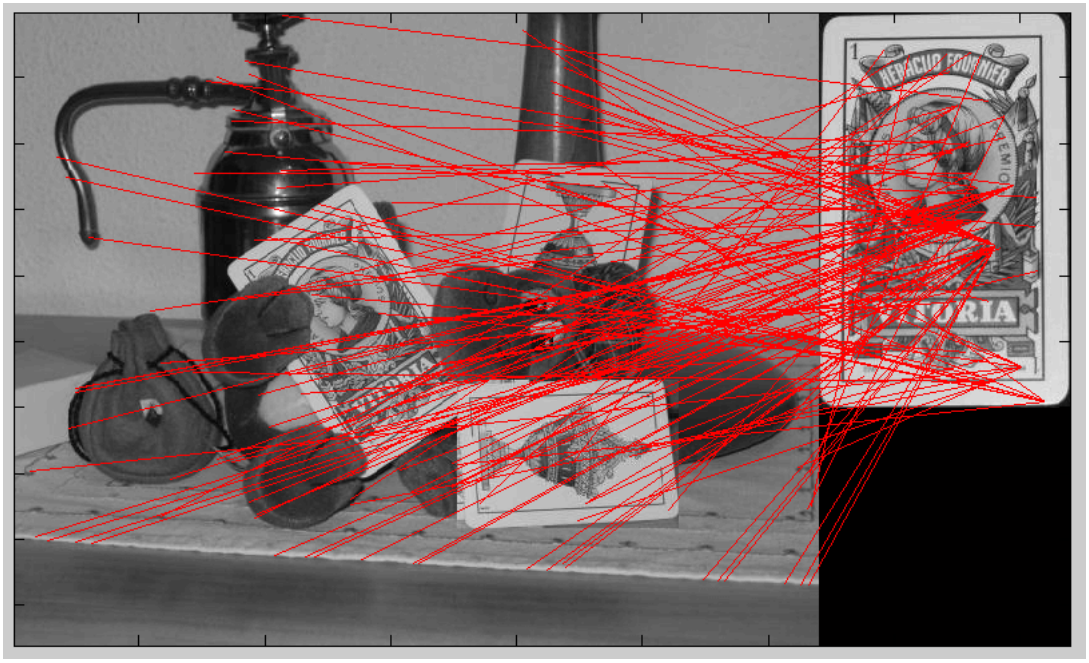


Figura 7.27: **Resultados y simulaciones.** *En esta imagen se muestran los "outliers" tras aplicar RANSAC junto con la aproximación afín.*

Figura 7.28: **Resultados y simulaciones.** *En esta imagen se muestra el perfil de reproyección del objeto detectado tras aplicar RANSAC junto con la aproximación afín.*

- **Cuarto ejemplo.**

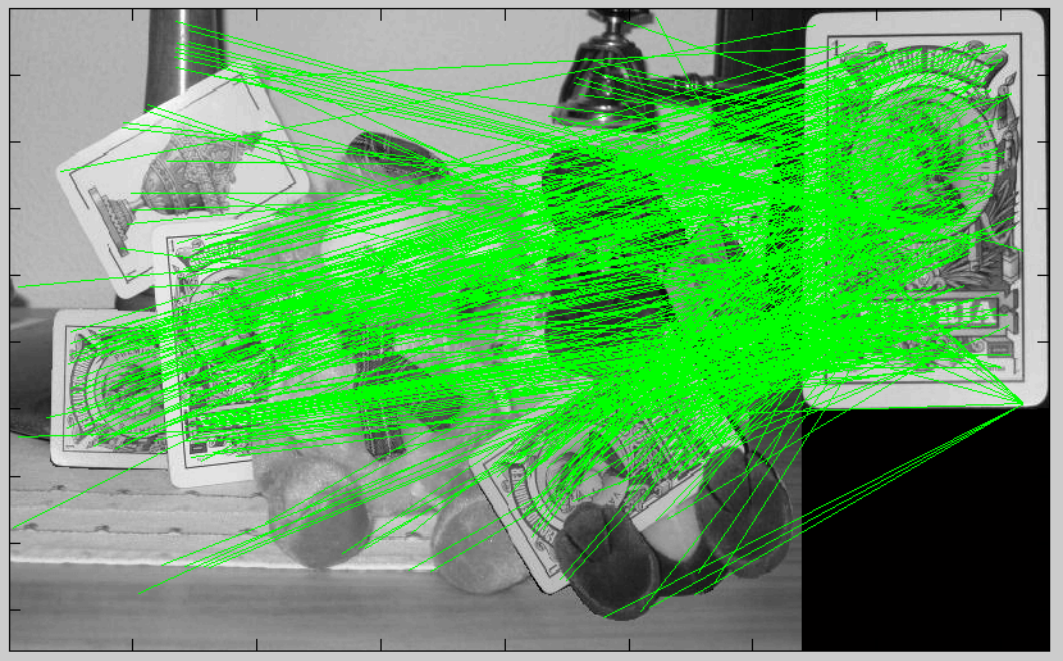


Figura 7.29: **Resultados y simulaciones.** *En esta imagen se muestra el "matching" inicial obtenido por SIFT. Cada par de correspondencias entre un punto del patrón y la imagen se representa con una recta de color verde* .

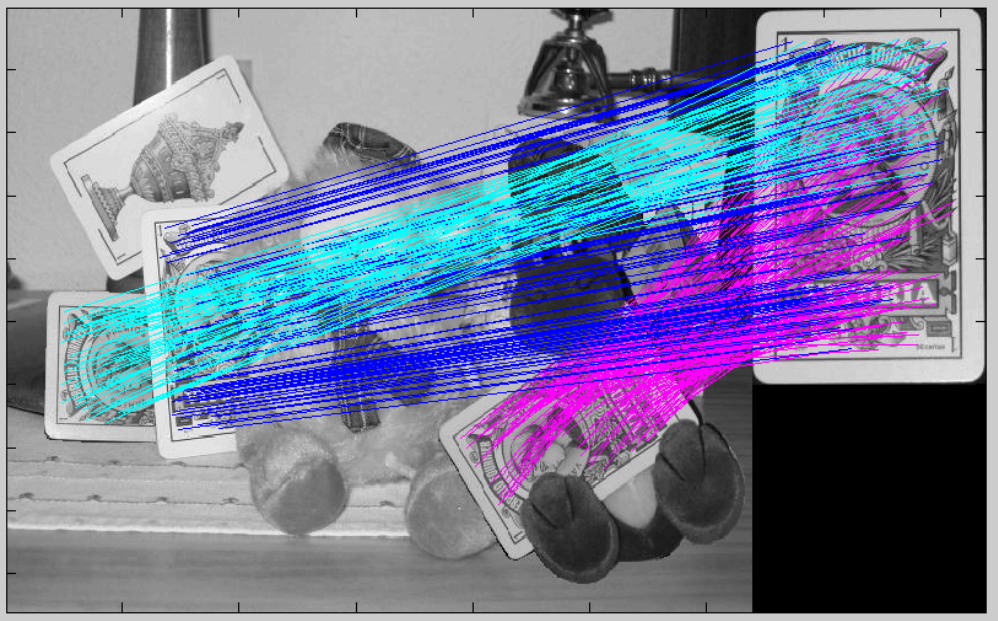


Figura 7.30: **Resultados y simulaciones.** *En esta imagen se muestran los "inliers" tras aplicar RANSAC junto con la aproximación afín.*

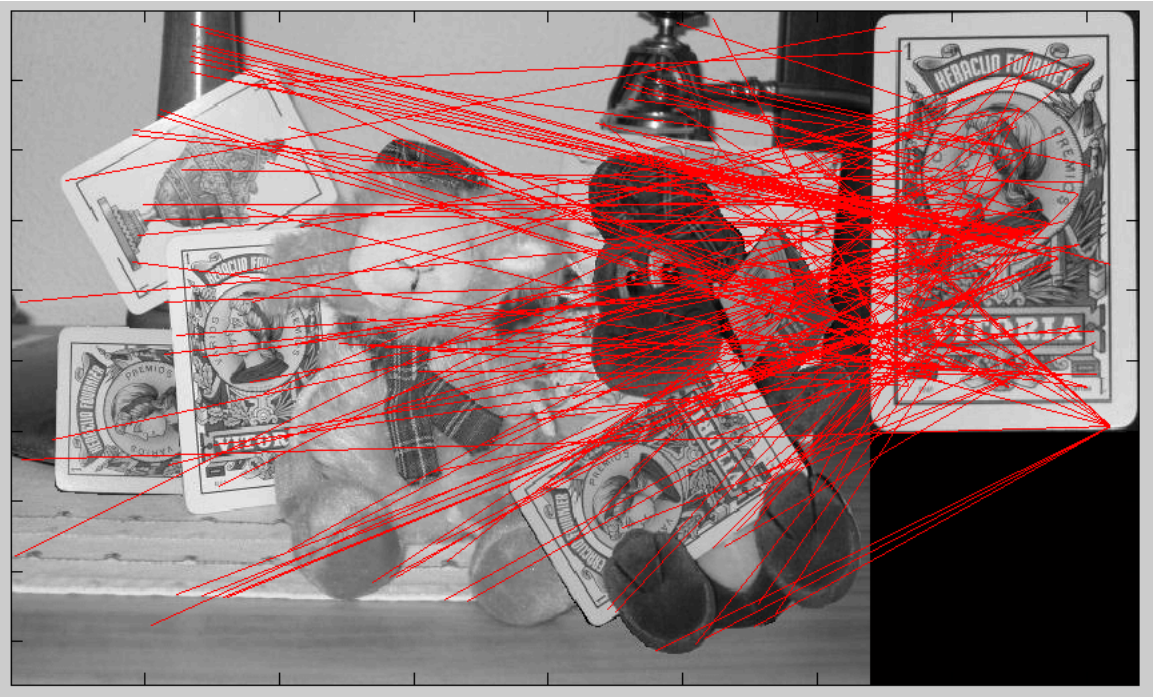


Figura 7.31: **Resultados y simulaciones.** *En esta imagen se muestran los "outliers" tras aplicar RANSAC junto con la aproximación afín.*

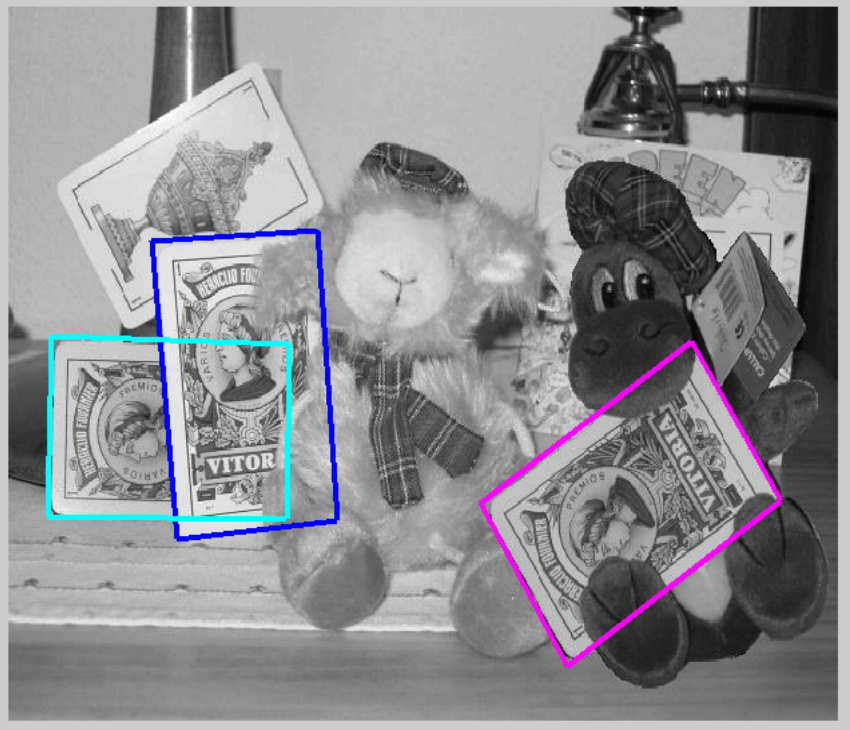


Figura 7.32: **Resultados y simulaciones.** *En esta imagen se muestra el perfil de reproyección del objeto detectado tras aplicar RANSAC junto con la aproximación afín.*

En la siguiente tabla, se muestra los resultados numéricos obtenidos en estos ejemplos:

	Ejemplo 1	Ejemplo 2	Ejemplo 3	Ejemplo4
Número de objetos	1	1	1	3
Descriptores en la imagen	1474	1297	1344	1968
Descriptores del patrón	842	842	842	842
"Matching" inicial	242	200	306	510
Número de "Inliers"	129	74	168	171, 104 y 59
Número de "Outliers"	113	126	138	176

Los resultados obtenidos utilizando RANSAC junto con el modelo de cámara proyectiva son similares a los del método anterior.

En concreto, estos ejemplos no valen para comparar las diferencias y los errores de estimación entre las soluciones obtenidas con la aproximación afín o considerando el caso general de cámara proyectiva pues las imágenes están tomadas a una distancia dentro del rango en el que la aproximación afín es válida. Aun así, se puede apreciar como ahora el número de "inliers" detectados es ligeramente mayor en comparación con el método afín. Esto se debe a que ahora se estiman los 8 parámetros de cada matriz H y por tanto, estas soluciones se adaptarán mejor al conjunto de correspondencias reales entre el patrón y cada objeto. Aun así, esta diferencia es insignificante y no hay prácticamente diferencia entre las reproyecciones de un modelo y el otro. Esto significa que la diferencia entre usar la aproximación afín o el caso general es despreciable.

7.1.2. Estudio del error en función de la distancia

En el capítulo 6 se planteo un sistema de detección suponiendo que el modelo de la cámara se podía aproximar a un modelo afín. Debido a las características que presenta una cámara afín, la relación entre un plano en $\mathbb{P}^3$ y su proyección en el plano imagen viene dada por una transformación afín en vez de por una homografía. Al considerar esta aproximación, el cálculo de la matriz H se simplifica pues únicamente tendrá 6 grados de libertad.

Sin embargo, la suposición de cámara afín no se puede aplicar en cualquier situación. Dicha aproximación será válida si la distancia entre la cámara y los objetos es grande comparada con el tamaño de los mismos. A medida que nos alejamos del objeto, las deformaciones son menos apreciables (en la imagen, las líneas paralelas se asemejarán más a líneas paralelas) y por tanto, la matriz H puede considerarse afín garantizando que el error cometido en cálculo de H se mantiene dentro de unos valores permisibles.

Cuando no tenemos esta situación, si se desea obtener una solución fiel a la realidad hay que utilizar el modelo de cámara proyectiva para el cual la relación entre el plano y su proyección viene dada por una homografía (8 grados de libertad).

En este apartado se realiza un estudio comparativo entre el error que se comete en la estimación de la matriz de transformación al utilizar un modelo de cámara proyectiva y la aproximación afín en función de la distancia a la que se encuentre el objeto. Con este análisis podemos determinar bajo qué condiciones de trabajo se puede aplicar la aproximación de cámara afín.

Para hacer el estudio del error se va a realizar el siguiente experimento:

- Tenemos una cámara calibrada situada en el origen de coordenadas de nuestro espacio de trabajo. También tenemos un objeto planar de dimensiones conocidas (ancho y largo). El origen del sistema de referencia del objeto se encuentra en el centro del mismo coincidiendo el eje z con el vector normal al plano.

- Para evaluar el error se van a tomar una serie de imágenes del objeto situado a diferentes distancias de la cámara y se le aplica rotaciones aleatorias en cada eje del sistema de referencia del objeto. En la figura 7.33 se muestra un esquema aclaratorio del experimeto que se pretende realizar.

- Debido a que el número de imágenes que se deben tomar es alto para obtener unos resultados estadísticos fiables, se van a generar las imágenes de forma sintética. Se parte de una imagen de un objeto del que se conocen sus dimensiones teóricas. Por tanto, se puede establecer una relación directa entre el objeto teórico real y la imagen:

$$M = \begin{pmatrix} \frac{l}{l_{imagen}} & 0 & u_o \\ 0 & \frac{w}{w_{imagen}} & v_o \\ 0 & 0 & 1 \end{pmatrix}$$

siendo l y w el largo y el ancho del objeto respectivamente expresados en milímetros. Si el origen de coordenadas del objeto no coincide con el origen tomado en la imagen, hay que aplicar un offset. Dicho offset se corresponde con (u_o, v_o) que son las coordenadas del centro del objeto expresados en el sistema de referencia de la imagen. Para este experimento, se va a considerar que el lado mayor del objeto mide 1000 mm.

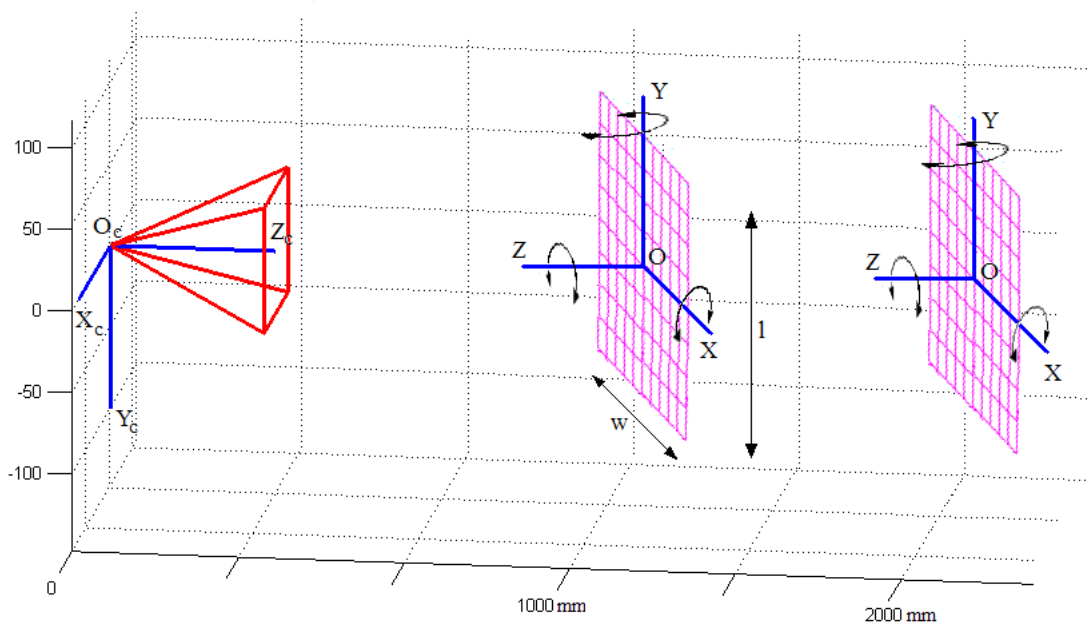


Figura 7.33: **Resultados y simulaciones.** *Esquema aclaratorio del experimento que se va a realizar. Se toman varias imágenes del objeto colocado a distintas distancias de la cámara aplicándole y se le aplica una serie de rotaciones aleatorias en cada eje.*

Al objeto se le aplica una rotación en cada uno de los tres ejes. El valor de cada ángulo de rotación de Euler va a ser aleatorio, dentro de un rango de valores:

$$R = R_{\gamma \text{ en } z} \cdot R_{\beta \text{ en } y} \cdot R_{\alpha \text{ en } x}$$

$$= \begin{pmatrix} \cos\gamma & \sin\gamma & 0 \\ -\sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos\beta & 0 & -\sin\beta \\ 0 & 1 & 0 \\ \sin\beta & 0 & \cos\beta \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & \sin\alpha \\ 0 & -\sin\alpha & \cos\alpha \end{pmatrix}$$

$$\alpha = [-20, 20]$$

$$\beta = [-20, 20]$$

$$\gamma = [-20, 20]$$

Por otro lado, el objeto se va a encontrar a una distancia d de la cámara, por tanto, la matriz de traslación que hay que aplicarle es la siguiente:

$$T = \begin{pmatrix} 0 \\ 0 \\ d \end{pmatrix} \qquad\qquad d \in [1000, 8000]\, mm$$

A partir de la matriz de rotación, la de traslación y la matriz de calibración de la cámara se obtienen la proyección del objeto en la imagen:

$$\mathbf{m} = \lambda K(R\mathbf{X} + T) \qquad \mathbf{X} = \begin{pmatrix} x \\ y \\ 0 \end{pmatrix}$$

donde $\mathbf{X}$ es un punto del objeto expresado en las coordenadas del mismo. Desarrollando la expresión anterior:

$$\mathbf{m} = KC_1x + KC_2y + KT$$

donde C_i se corresponde con el vector columna i de la matriz de rotación. Puesto que la imagen se va a crear de forma sintética a partir de una imagen, hay que obtener la matriz de homografía que hay que aplicarle a la imagen para simular la transformación real que sufriría el objeto.

$$\mathbf{m} = \lambda H \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

$$= \lambda \begin{pmatrix} KC_1 & KC_2 & KT \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

$$H = f(K,\, R,\, T,\, M) \quad \longrightarrow \quad H = \begin{pmatrix} KC_1 & KC_2 & KT \end{pmatrix} \tag{7.1}$$

Finalmente, la imagen dol objeto transformada se superpone sobre otra imagen de fondo (la finalidad de la foto de fondo es la de generar “outliers” en el “matching” inicial).

- Una vez que se otiene el conjunto de imágenes del experimento, se les aplica a todas ellas los dos métodos de detección propuestos (algoritmo RANSAC junto con la aproximación de cámara afín y el mismo algoritmo considerando que la cámara es proyectiva). Si RANSAC es capaz de detectar el objeto, se evalúa el error que se comete en la estimación de la matriz H. Puesto que la posición del objeto en cada una de las imágenes y la transformación que se ha aplicado al mismo son conocidas, el error, expresado en píxeles, se calcula a partir de las distancias Euclídeas entre

las proyecciones reales de 4 puntos del patrón del objeto y sus correspondientes proyecciones utilizando la matriz H obtenida con RANSAC (se utiliza este método de reproyección debido a la dificultad para comparar la matriz de transformación real y la matriz H obtenida):

$$\text{Error} = \frac{1}{4}(d_1 + d_2 + d_3 + d_4)$$

donde:

$$d_i = (\mathbf{m}_i - H\mathbf{X_i})^{1/2}$$

En la figura 7.34 se muestra un esquema aclaratorio:

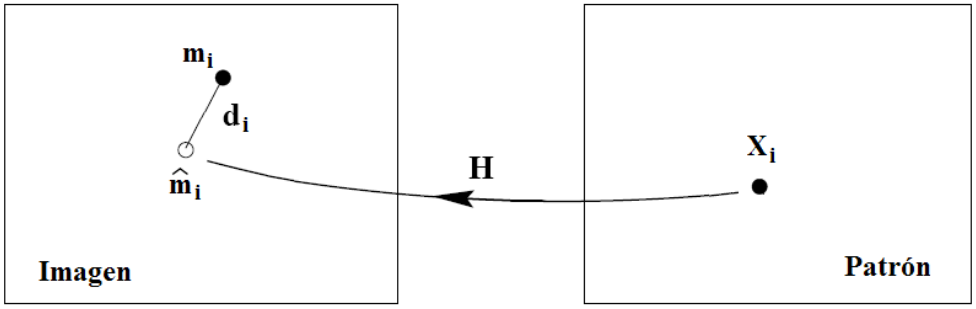


Figura 7.34: **Resultados y simulaciones.** *Esquema aclaratorio del método que se va a utilizar para determinar el error que se comete en la estimación de la matriz H.*

- Se va a representar varias gráficas que van a permitir analizar para qué distancias es válida la aproximación afín. Dichas gráficas son las siguientes:

  - Gráfica de la media y varianza del error que se comete utilizando la aproximación afín y el modelo de cámara proyectiva en función de la distancia a la que se encuentre el objeto. El error se calcula únicamente con las imágenes donde se ha detectado el objeto.
  - Gráfica de la probabilidad de detección en función de la distancia entre el objeto y la cámara.
  - Número medio de “inliers” y “outliers” en las imágenes (si el sistema detecta más de un objeto, se considera que la detección no es correcta y por tanto el número de “inliers” se iguala a 0).

- **Estudio del error utilizando la aproximación de cámara afín** - Primero se ha aplicado RANSAC con la aproximación afín para detectar el objeto en cada una de las imágenes generadas. Los resultados obtenidos son los siguientes:

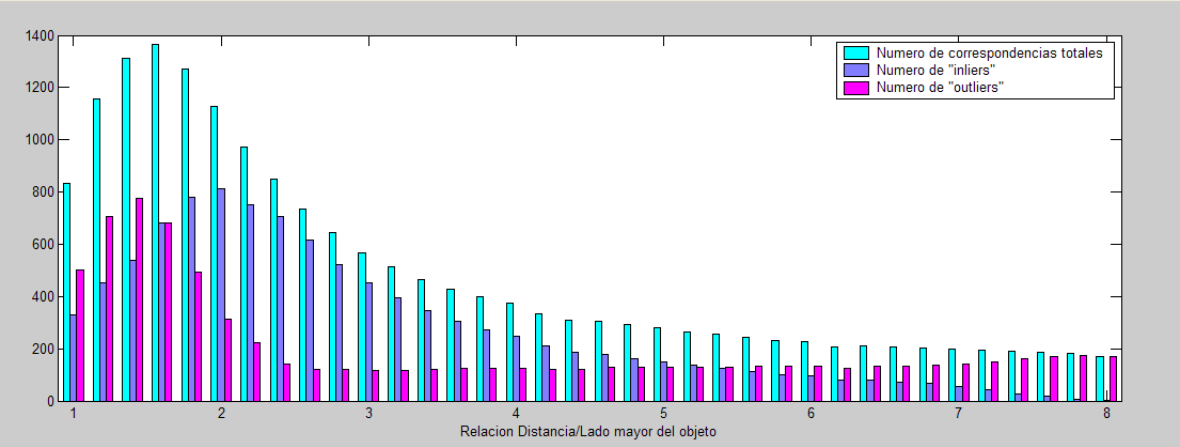


Figura 7.35: **Resultados y simulaciones.** *En está gráfica se representa el número medio de descriptores encontrados en la imagen junto con el número de "inliers" y de "outliers" totales en función de la relación distancia/tamaño del objeto.*

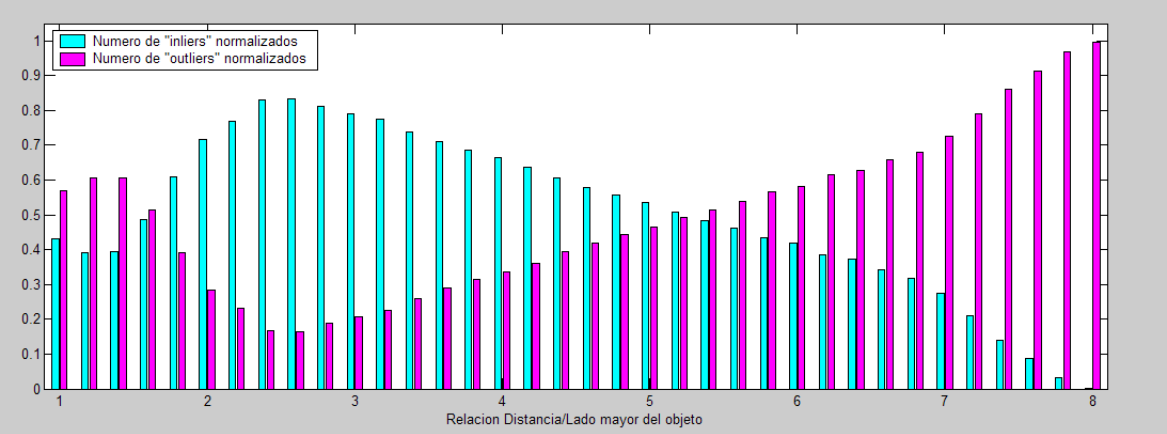


Figura 7.36: **Resultados y simulaciones.** *En está gráfica se representa el número medio de "inliers" y de "outliers" normalizados respecto del número total de descriptores encontrados en las imágenes en función de la relación distancia/tamaño del objeto.*

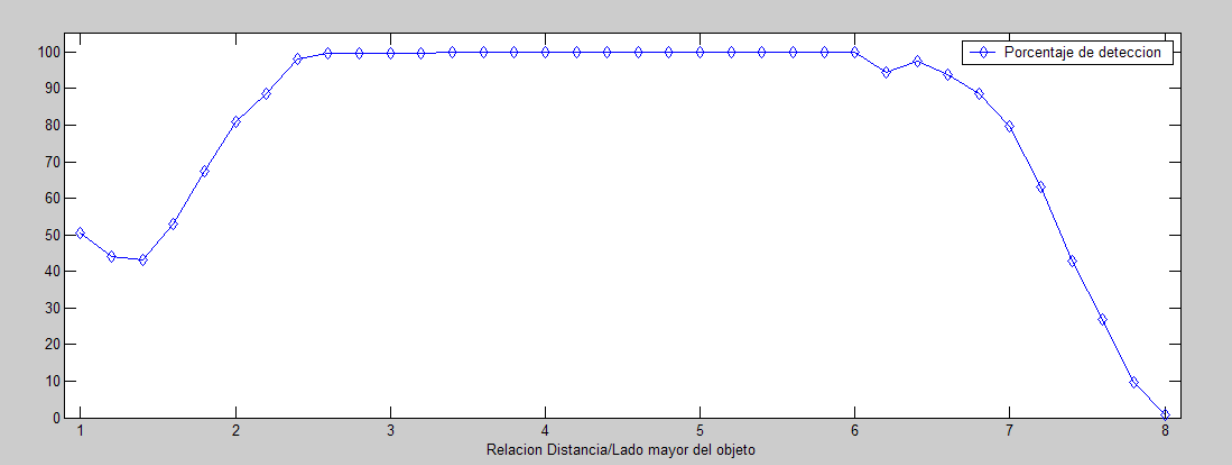


Figura 7.37: **Resultados y simulaciones.** *En esta gráfica se representa el porcentaje de detección en función de la relación distancia/tamaño del objeto.*

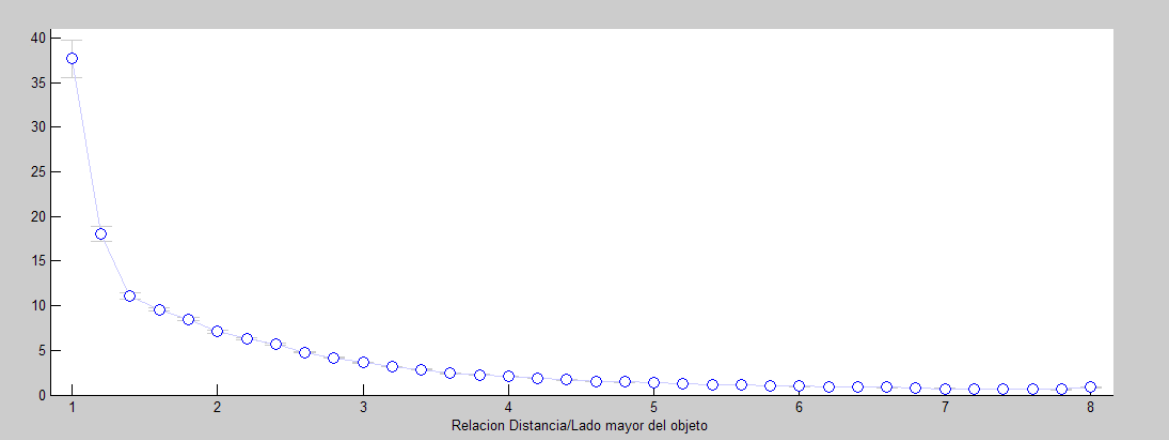


Figura 7.38: **Resultados y simulaciones.** *En esta gráfica se representa la media y la varianza de error que se comete en la estimación de la matriz H en función de la relación distancia/tamaño del objeto.*

Si se obsevan las figuras 7.35 y 7.36 se puede apreciar que el número medio de "inliers" normalizados es menor en comparación con el de "outliers" para distancias pequeñas. Este valor va creciendo a medida que aumenta la distancia, llegando a alcanzar su máximo para una relación entre la distancia y el tamaño del objeto en torno a $2{,}6$ (que se corresponde con una distancia de 2600 mm en nuestro caso). A partir de este punto, el número de "inliers" va decreciendo hasta anularse casi por completo para una relación de 8.

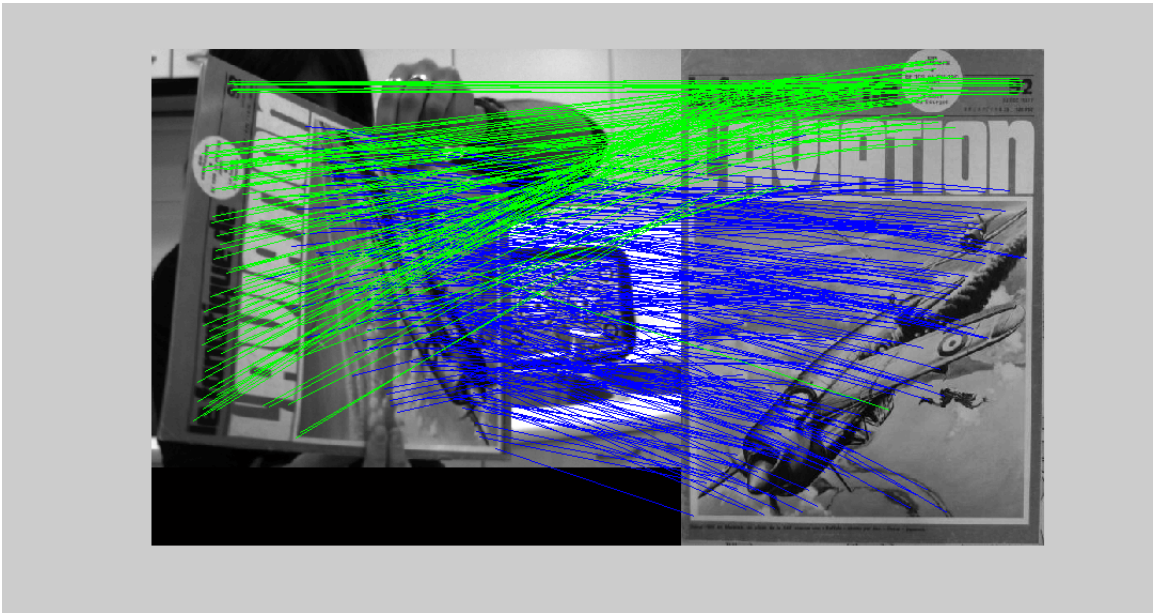
La justificación de este comportamiento es la siguiente:

- Cuando la distancia es muy pequeña, las deformaciones que sufre el objeto cuando es proyectado en la imagen son considerables. En esta situación, los puntos del "matching" inicial que se corresponderían con los "inliers" reales, no se adaptan a un único modelo en la mayoría de los casos, sino que generan subgrupos de puntos, cada uno de ellos con una solución distinta (por tanto, el resultado final es un error en la detección pues encuentra más de un objeto donde realmente sólo habría uno).

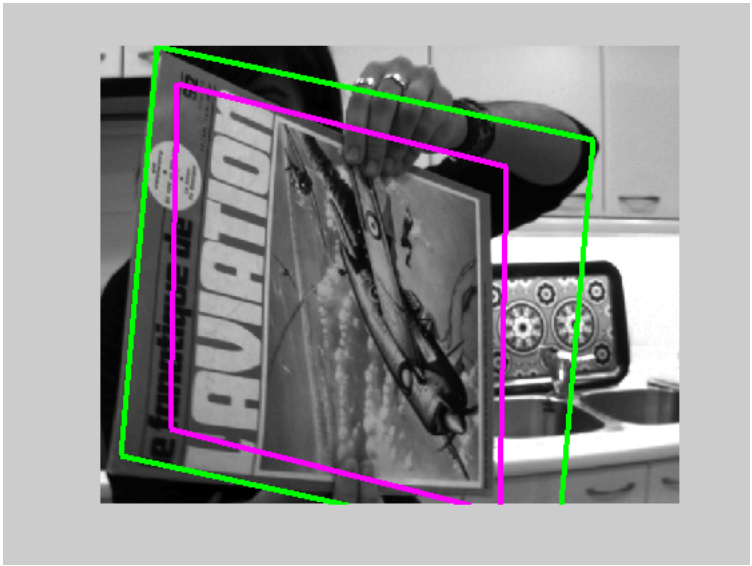
  La figura 7.39 se muestra un ejemplo que puede servir para clarificar todo esto. El objeto esta situado a muy poca distancia de la cámara por lo que la deformación proyectiva que sufre el objeto está muy acentuada (se puede apreciar en la imagen que las líneas paralelas no se proyectan como tales). Cuando esto ocurre, el algoritmo de RANSAC no es capaz de ajustar todas las correspondencias correctas a una única matriz afín.

- A medida que la distancia aumenta, la proyección del objeto se adapta mejor a una transformación afín, por lo que RANSAC es capaz de ajustar todas las correspondencias a un único modelo. Sin embargo, a partir de un determinada distancia, el número de "inliers" vuelve a disminuir de forma progresiva. Esta disminución no se debe a la aproximación realizada sino a la escala. Como se verá más adelante, el número de descriptores que se detectan en un objeto tiene una fuerte dependencia con la escala (a medida que la escala se reduce, el número de descriptores disminuye). Por tanto, a partir de una cierta distancia, el objeto deja de ser detectado porque el número de correspondencias correctas es nulo o no supera el mínimo necesario para considerar que el modelo detectado por RANSAC es correcto.

En la figura 7.37 se muestra el porcentaje de detección. Para una relación entre distancia y tamaño del objeto en torno a 1 (en nuestro caso 1000 mm), el porcentaje de obtener un detección correcta es muy bajo. A partir de $1{,}5$, la detección mejora a medida que aumentamos la distancia, llegando a un 100 % de detección a partir de una relación igual a $2{,}5$. El efecto negativo de la disminución de escala empieza a apreciarse a partir de los 6000 mm, decayendo rápidamente hasta llegar a anularse a los 8000 mm.



(a)

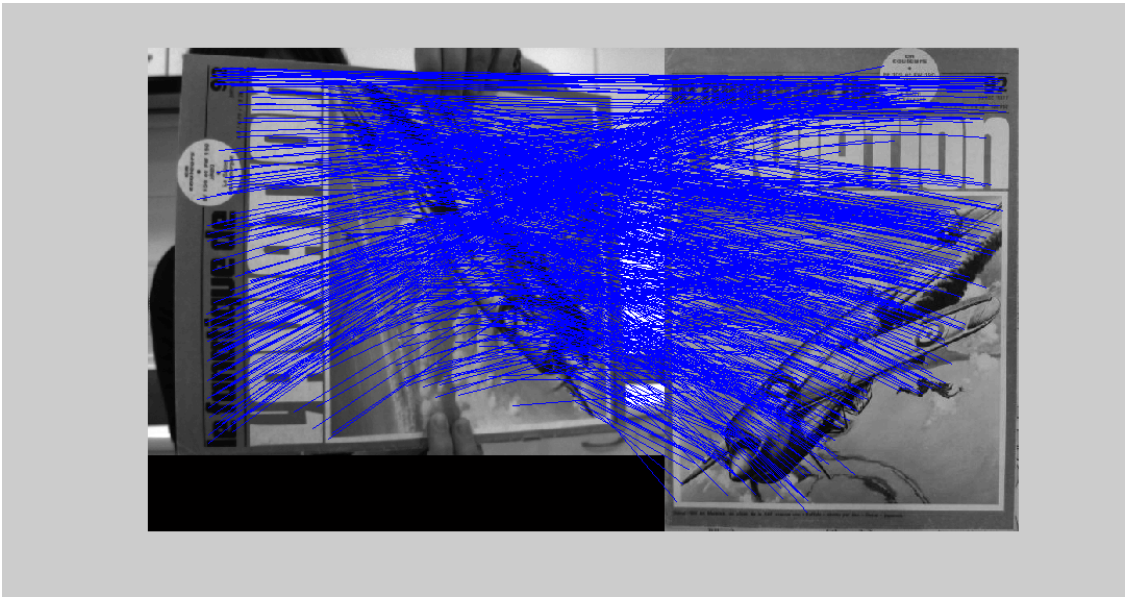


(b)

Figura 7.39: **Resultados y simulaciones.** *En estas gráficas se muestra un ejemplo de detección utilizando la aproximación de cámara afín cuando el objeto se encuentra a poca distancia de la cámara. (a) Conjunto de "inliers" detectados por RANSAC. Cada color se corresponde con un conjunto de "inliers" distintos. (b) Perfiles estimados de los objetos detectados.*

Para determinar el rango de distancias entre las cuales podemos aplicar la aproximación de cámara afín, no basta con analizar simplemente el porcentaje de detección. También hay que evaluar el error que se comete en la estimación de la matriz H. Cuando la distancia es pequeña en relación con el tamaño del objeto, no sólo la probabilidad de obtener una solución correcta es baja sino que también el error medio es grande cuando el sistema es capaz de obtener una solución. Debido a que

la matriz de transformación estimada es afín, el paralelismo teóricamente es invariante por lo que la matriz H no se adapta bien a la deformación real que sufre el objeto. En la figura 7.40 se muestra un ejemplo en el que se puede apreciar como, al proyectar la silueta del objeto detectado utilizando la matriz H estimada, las líneas paralelas permanecen paralelas por lo que el perfil sólo se adapta parcialmente al objeto.

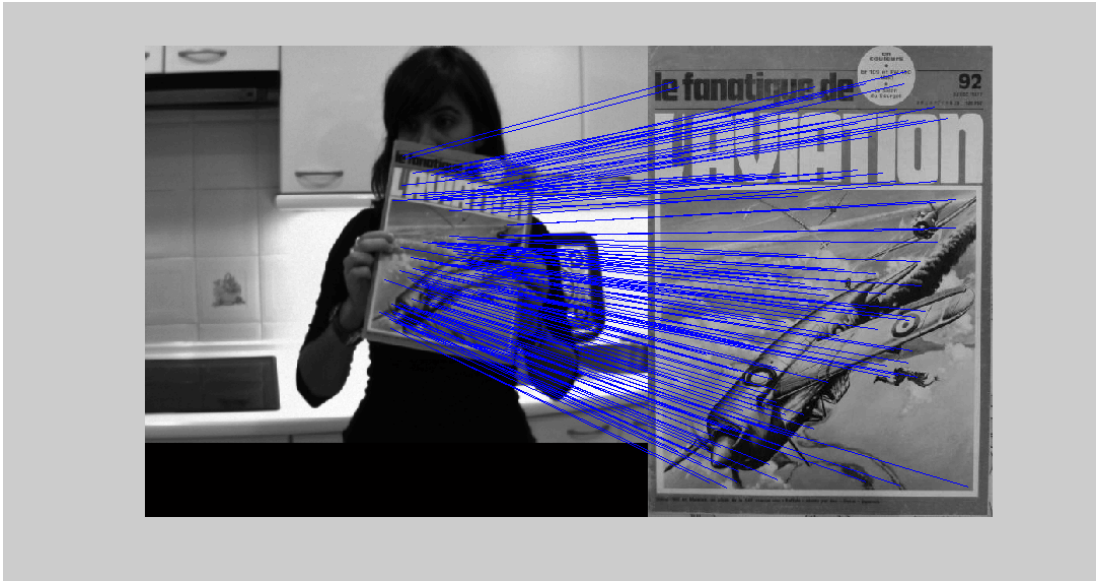


(a)

(b)

Figura 7.40: **Resultados y simulaciones.** *En estas gráficas se muestra un ejemplo de detección utilizando la aproximación de cámara afín cuando el objeto se encuentra a poca distancia de la cámara. En la imagen se puede apreciar como el perfil estimado del objeto no se adapta completamente al objeto. (a) Conjunto de "inliers" detectados por RANSAC. (b) Perfil estimado del objeto detectado.*

A medida que la distancia aumenta, el error disminuye pues las deformaciones proyectivas se atenúan y se asemejan más a transformaciones afines. La figura 7.41 se corresponde con una imagen del mismo objeto de la figura anterior tomada a más distancia. En este caso, se puede apreciar como el perfil estimado utilizando la aproximación de cámara afín se adapta perfectamente al objeto. También se puede ver el efecto de que produce la disminución de la escala en el número de "inliers" detectados.



(a)

(b)

Figura 7.41: **Resultados y simulaciones.** *En estas gráficas se muestra un ejemplo de detección utilizando la aproximación de cámara afín cuando el objeto se encuentra dentro del rango de distancias en las que se puede aplicar dicha aproximación. Se puede apreciar como el perfil estimado del objeto se adapta completamente a él. (a) Conjunto de "inliers" detectados por RANSAC. (b) Perfil estimado del objeto detectado.*

Analizando simultaneamente las figuras 7.37 y 7.38 se puede determinar que la aproximación de cámara afín se puede usar siempre y cuando la relación entre distancia y tamaño del objeto se encuentre comprendido entre 3 y $6{,}5$ aproximadamente. En este intervalo de trabajo se puede garantizar que el porcentaje de detección es prácticamente del 100 % y el error medio es pequeño (inferior a 5 píxeles de error).

- **Estudio del error utilizando el modelo de cámara proyectiva** - Se acaba de analizar bajo qué condiciones se obtienen buenos resultados de detección utilizando la aproximación afín. Ahora se va a realizar el mismo estudio utilizando el modelo cámara proyectiva para poder comparar ambos métodos y ver si la mejora que se obtiene en el porcentaje de detección y el error medio utilizando la matriz de homografía es despreciable respecto al caso afín y así determinar si realmente compensa su uso. Las gráficas obtenidas son las siguientes:

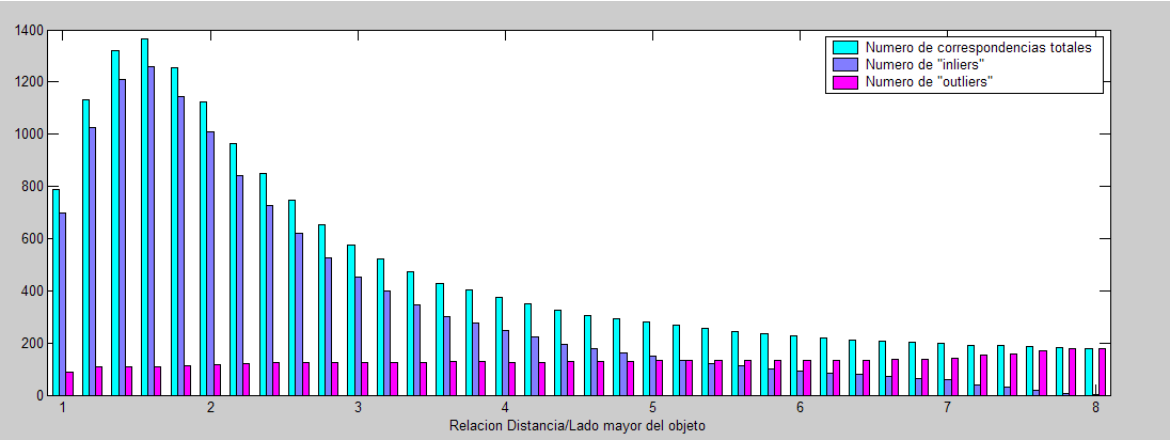


Figura 7.42: **Resultados y simulaciones.** *En está gráfica se representa el número medio de descriptores encontrados en la imagen junto con el número de "inliers" y de "outliers" totales en función de la relación distancia/tamaño del objeto.*

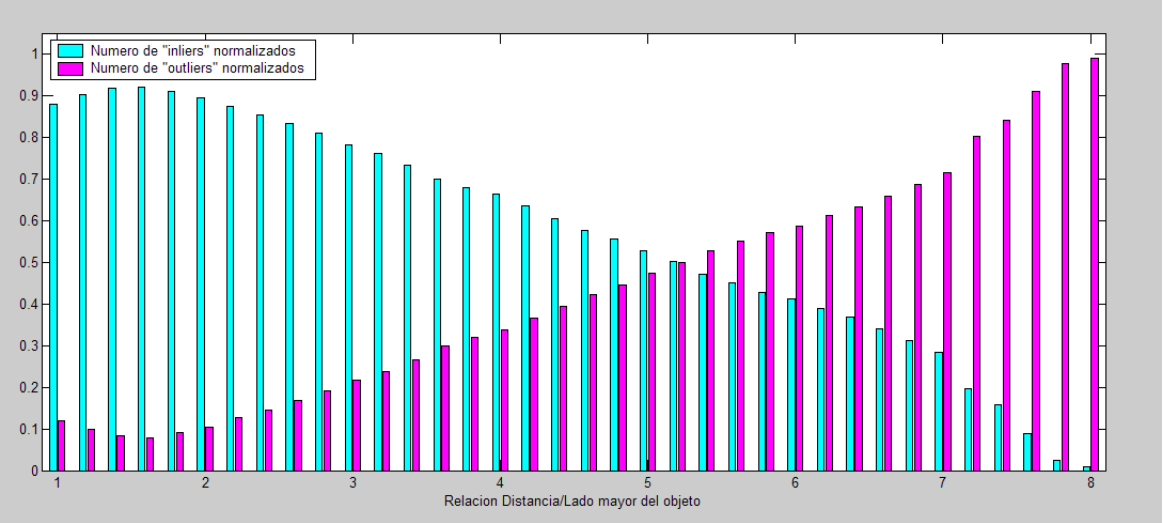


Figura 7.43: **Resultados y simulaciones.** *En está gráfica se representa el número medio de "inliers" y de "outliers" normalizados respecto del número total de descriptores encontrados en las imágenes en función de la relación distancia/tamaño del objeto.*

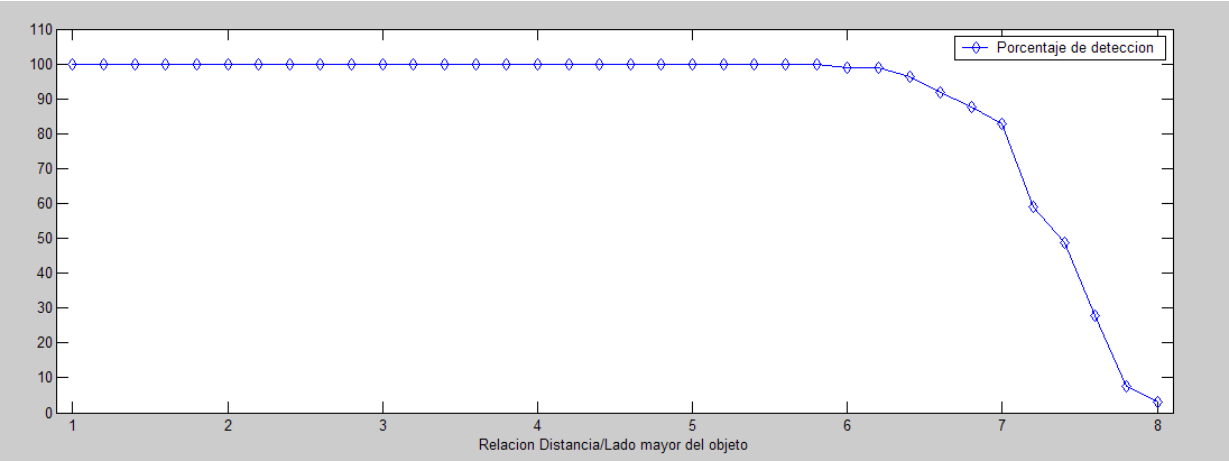


Figura 7.44: **Resultados y simulaciones.** *En esta gráfica se representa el porcentaje de detección en función de la relación distancia/tamaño del objeto.*

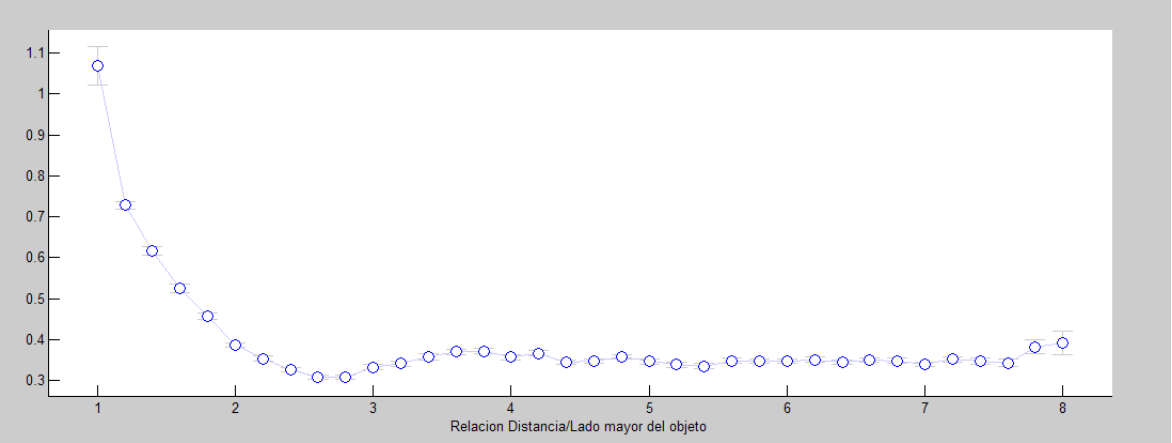


Figura 7.45: **Resultados y simulaciones.** *En esta gráfica se representa la media y la varianza de error que se comete en la estimación de la matriz H en función de la relación distancia/tamaño del objeto.*

Ahora se va a considerar el caso más general en el que las transformaciones que sufren los objetos son proyectivas. Por tanto, al no imponer ningún tipo de restricción en la matriz H, el algoritmo RANSAC debería ser capaz de calcular, en condiciones normales (de escala, ruido, etc), una solución muy próxima a la real.

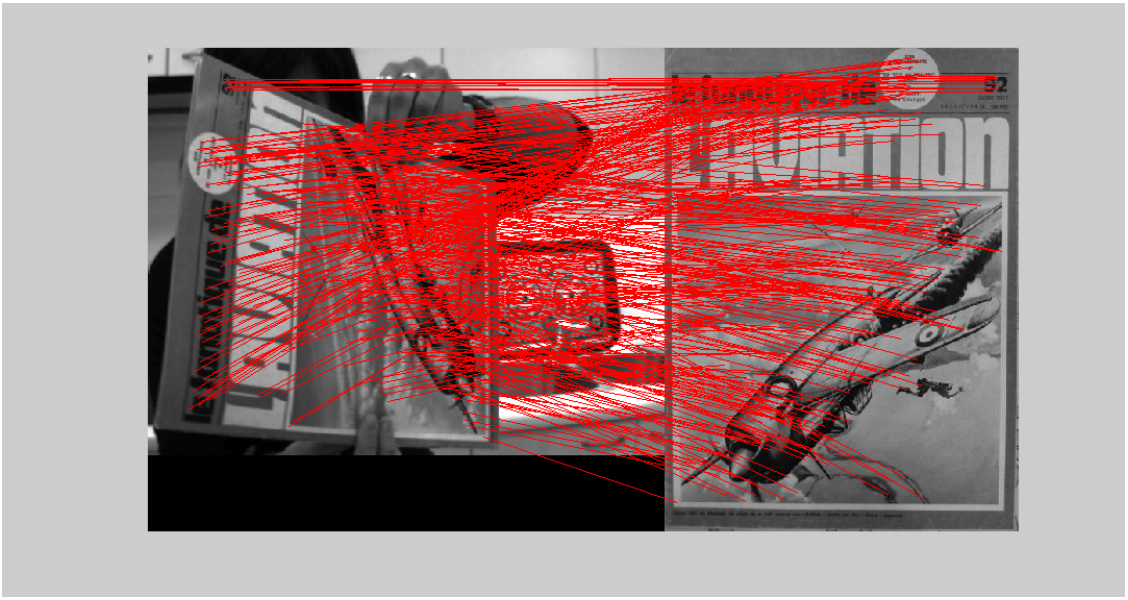
Si se obsevan las figuras 7.42 y 7.43, se puede apreciar que al comienzo el número de "inliers" va creciendo hasta alcanzar el máximo cuando la relación entre la distancia y el tamaño del objeto está en torno a $1{,}5$. A partir de este momento, el número de "inliers" comienza a decrecer hasta anularse por completo cuando la relación es de 8 aproximadamente, al igual que pasaba con el modelo de cámara afín. Sin embargo, en este caso, el número medio de "inliers" normalizados es mucho mayor que el de "outliers" incluso para distancias cortas. El hecho de que el número de "inliers" al comienzo sea menor no se debe a fallos en la detección (pues se puede apreciar que aunque el número sea menor, es prácticamente igual al número de correspondencias iniciales) sino a que parte del objeto queda fuera de la imagen para ciertas combinaciones de ángulos de rotación. A partir de una cierta distancia, el número de "inliers" comienza a decrecer debido a la disminución de la escala, como pasaba en el caso anterior.

En la figura 7.44 se muestra el porcentaje de detección. A diferencia del caso anterior, el sistema es capaz de detectar los objetos incluso a corta distancia. El porcentaje de detección se mantiene aproximadamente en el 100 % hasta un valor de $6{,}5$ en la relación entre la distancia y el tamaño del objeto, decreciendo rápidamente hasta llegar a ser cero para una relación de 8. La rápida disminución del porcentaje

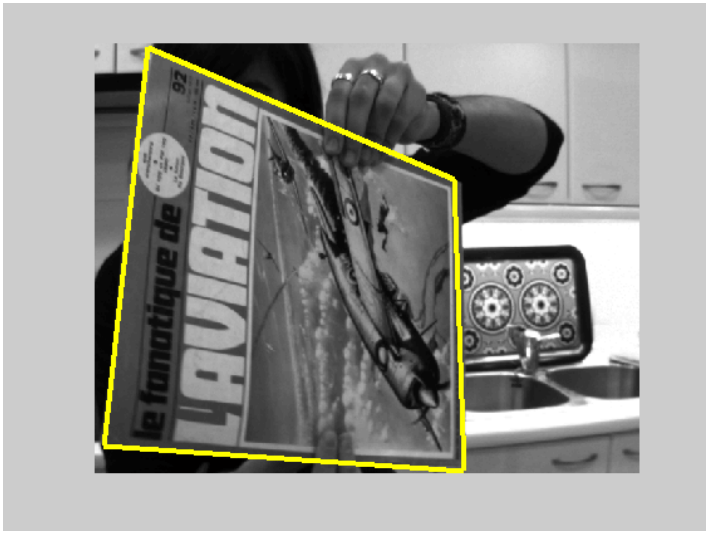
de detección se debe a que el factor de escala disminuye tanto que el número de correspondencias llega a ser nulo o no supera el valor mínimo necesario para considerar que la solución válida.

Respecto al error, en la figura 7.45 se puede observar que independientemente de la distancia entre la cámara y el objeto, si el sistema es capaz de encontrar una solución, el error medio no es superior a $1{,}1$ píxeles. Los errores mayores se producen para relaciones entre distancia y tamaño del objeto menores de 2. Esto se debe a que a corta distancia, las deformaciones proyectivas se acentúan más y para ciertas combinaciones de valores de los ángulos de rotación (cuando toman valores extremos dentro de su rango de variación) el sistema de detección es un poco más sensible (pero aun así, el error es despreciable). Para distancias mayores, el error esta en torno a 0.4 píxeles.

En las figuras 7.46 y 7.47 se muestran los mismos ejemplos de las figuras 7.39 y 7.41 respectivamente, con la diferencia de que ahora se estima la matriz H considerando que las transformaciones son proyectivas. Se puede apreciar como el sistema detecta el objeto correctamente incluso a poca distancia (con el modelo afín, el objeto no se detectaba correctamente) y que el error de estimación es muy pequeño pues los perfiles obtenidos a partir de la matriz H se adaptan perfectamente al contorno del objeto.

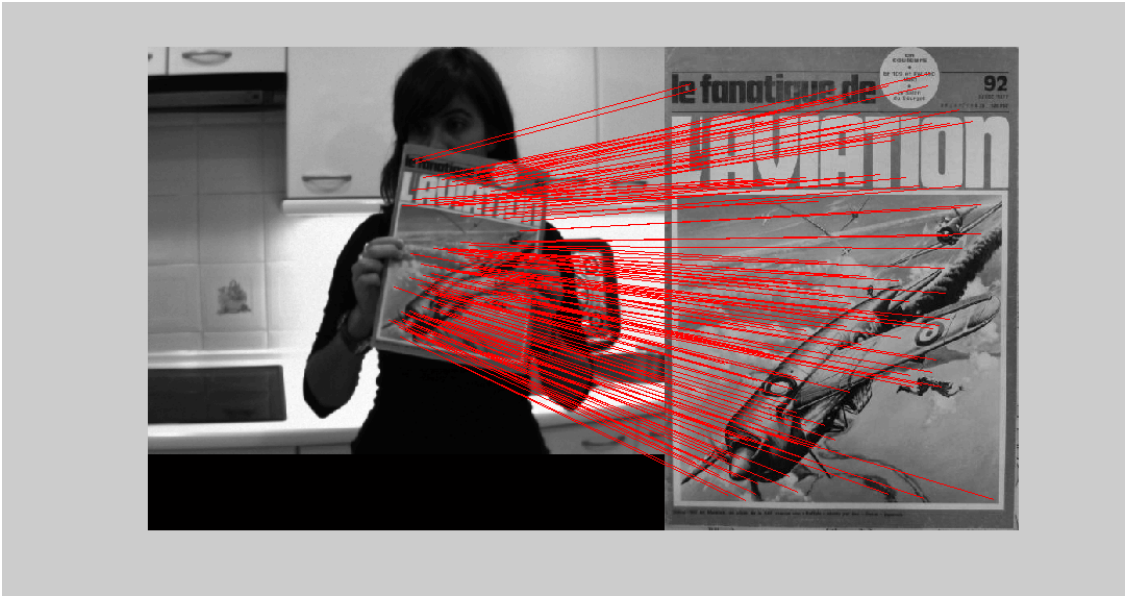


(a)



(b)

Figura 7.46: **Resultados y simulaciones.** *En estas gráficas se muestra un ejemplo de detección utilizando el modelo de cámara proyectiva cuando el objeto se encuentra a poca distancia de la cámara. En la imagen se puede apreciar como el perfil estimado del objeto se adapta perfectamente al objeto. (a) Conjunto de "inliers" detectados por RANSAC. (b) Perfil estimado del objeto detectado.*



(a)

(b)

Figura 7.47: **Resultados y simulaciones.** *En estas gráficas se muestra un ejemplo de detección utilizando el modelo de cámara proyectiva cuando el objeto se encuentra alejado de la cámara. En la imagen se puede apreciar como el perfil estimado del objeto se adapta perfectamente al objeto. (a) Conjunto de "inliers" detectados por RANSAC. (b) Perfil estimado del objeto detectado.*

- **Comparación de ambos sistemas de detección** - Se van a representar en las mismas gráficas el error y el porcentaje de detección de cada uno de los casos (para la aproximación afín y para el modelo de cámara proyectiva).

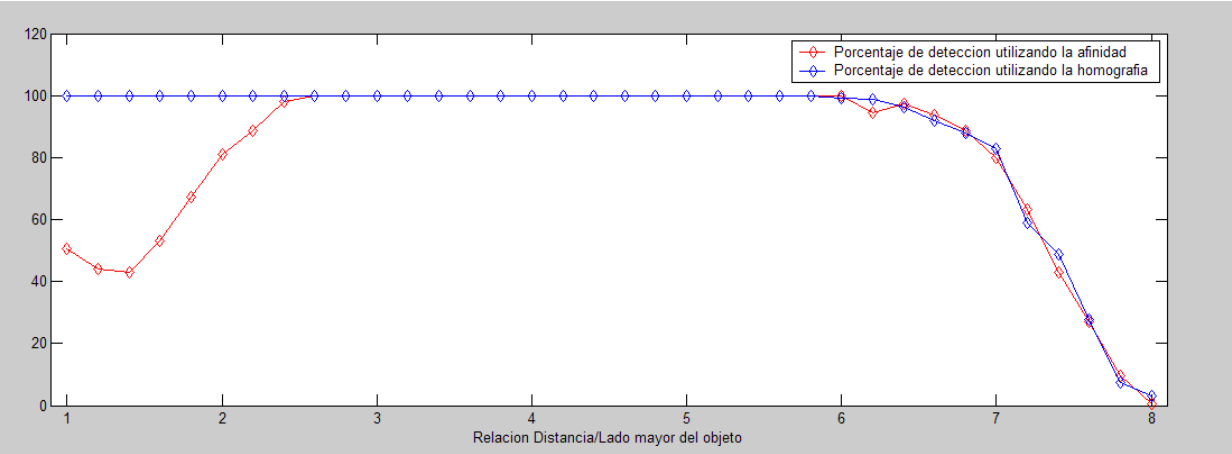


Figura 7.48: **Resultados y simulaciones.** *En esta gráfica se representa el porcentaje de detección para cada modelo propuesto. Con rojo se ha representado el porcentaje de detección para el caso en que se utiliza la aproximación afín y en azul, el modelo general.*

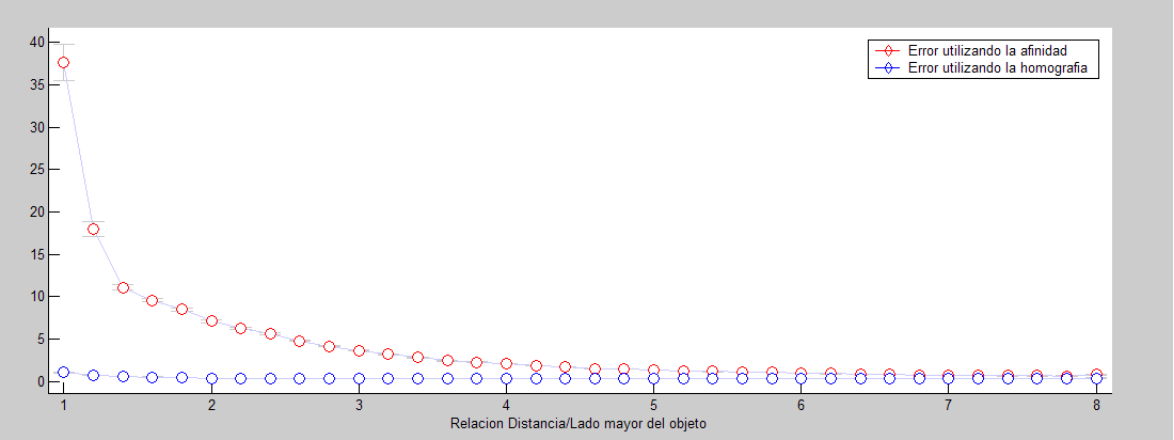


Figura 7.49: **Resultados y simulaciones.** *Comparación del error en la detección para la aproximación afín (en rojo) y para el modelo de cámara proyectiva (en azul) en función de la relación distancia/tamaño del objeto.*

Observando las gráficas anteriores, se puede comprobar como los resultados que se obtienen utilizando el modelo de cámara afín son similares a los del caso general siempre que se trabaje en unas condiciones concretas. Si la relación entre la distancia y el tamaño del objeto es superior a 3 ó 4 (en función de si queremos un sistema más o menos preciso) se puede aplicar la aproximación pues el error de aproximación en estos casos es despreciable y el hecho de estimar la matriz de homografía

no aporta ninguna mejora significativa. Únicamente hace que el tiempo de computo sea mayor. En la siguiente gráfica se muestran los tiempos de ejecución en función del número de objetos a estimar. Se puede observar que el tiempo de ejecución para el caso de la homografía crece más rápidamente a medida que el número de objetos a detectar es mayor .

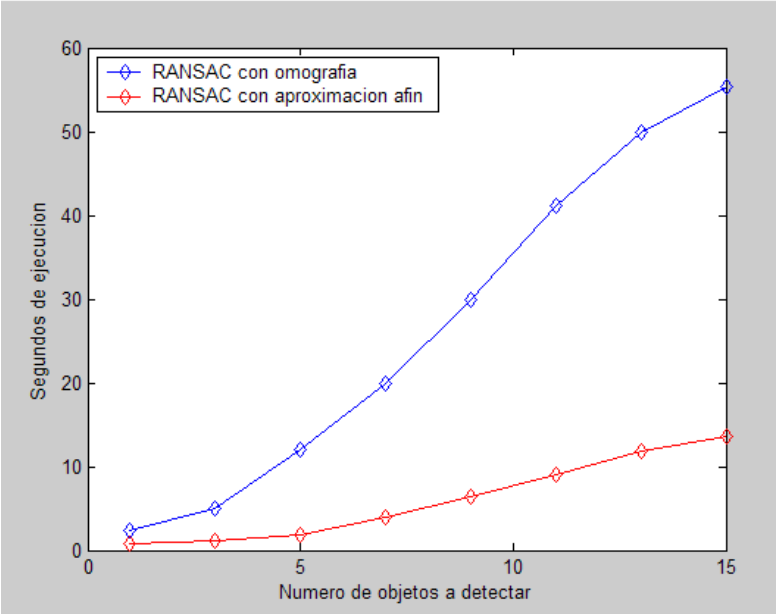
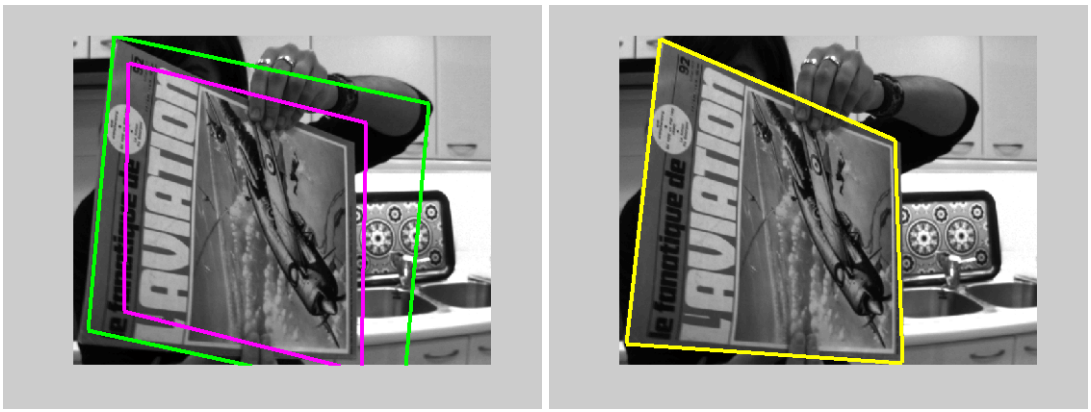


Figura 7.50: **Resultados y simulaciones.** *Tiempos de ejecución en función del número de objetos a estimar y del tipo de solución que se calcula (matriz de afinidad u homografía).*

Cuando la distancia crece considerablemente la capacidad de detección disminuye, independientemente del método utilizado. Si por el contrario la distancia a la cámara es pequeña, no hay más remedio que utilizar el método general si se desea que el error de estimación sea despreciable.

Una de las condiciones que se cumplen en el “Espacio Inteligente” es que los objetos (planares o tridimensionales) son observados por las cámaras del entorno a una distancia lo suficientemente grande para considerar que las deformaciones que sufren son afines. Si además se garantiza que la orientación de los objetos tridimensionales no va a sufrir grandes variaciones (es decir, que la vista del objeto utilizada para la generación del patrón permanezca en todo momento visible por la cámara) se puede considerar que dicho objeto tiene un comportamiento similar a uno plano. El la figura **??** se muestra un ejemplo de detección, utilizando la aproximación afín, realizado en el “Espacio Inteligente” del Departamento de Electrónica de la Universidad de Alcalá (Ispace). Desde una de las cámaras del Ispace se ha captado una imagen del entorno en el que se encuentran un robot (agente controlable), una persona (agente autónomo) y otra serie de objetos estáticos repartidos a lo largo del espacio. En este ejemplo, el sistema es capaz de detectar a la persona que se encuentra en el entorno.



(a) (b)

Figura 7.51: **Resultados y simulaciones.** *En estas gráficas se muestra un ejemplo de detección utilizando el modelo de cámara proyectiva cuando el objeto se encuentra alejado de la cámara. En la imagen se puede apreciar como el perfil estimado del objeto se adapta perfectamente al objeto. (a) Conjunto de "inliers" detectados por RANSAC. (b) Perfil estimado del objeto detectado.*

(a) (b)

Figura 7.52: **Resultados y simulaciones.** *En estas gráficas se muestra un ejemplo de detección utilizando el modelo de cámara proyectiva cuando el objeto se encuentra alejado de la cámara. En la imagen se puede apreciar como el perfil estimado del objeto se adapta perfectamente al objeto. (a) Conjunto de "inliers" detectados por RANSAC. (b) Perfil estimado del objeto detectado.*

(a)

(b)

Figura 7.53: **Resultados y simulaciones.** *En estas gráficas se muestra un ejemplo de detección, utilizando el modelo de cámara afín, realizada en el "Espacio Inteligente del Departamento de Electrónica de la Universidad de Alcalá. En concreto, se quiere detectar a una persona que se encuentra en él. (a) Imagen tomada desde una cámara del "Espacio Inteligente". (b) Perfil superpuesto de la persona detectada.*

7.1.3. Estudio del error en función del grado de oclusión del objeto

Una de las ventajas que presenta el uso del método SIFT es que el número de descriptores que se encuentran en una sola imagen es muy alta. Por tanto, si el modelo de apariencia está compuesto por un alto número de descriptores, la probabilidad de detectar dicho objeto en una imagen es alta incluso cuando gran parte del mismo se encuentre ocluido (el número de "inliers" mínimo para obtener una solución válida de la matriz H es bajo comparado con el número de descriptores que puede tener el modelo de apariencia).

En este apartado se va a analizar la influencia del grado de oclusión de un objeto en la detección del mismo. Para ello, se ha realizado un experimento similar al explicado en el apartado anterior:

- Se ha tomado una imagen de un objeto planar y se ha obtenido su modelo de apariencia. Se ha considerado que los descriptores están repartidos de forma uniforme por todo el objeto.

- A dicha imagen se le ha aplicado una serie de transformaciones sintéticas (afines para el caso de cámara afín y proyectivas para el caso general en el que la cámara es proyectiva) variando de forma aleatoria y controlada la escala, la rotación y la deformación proyectiva dentro de unos rangos de valores. A su vez, se ha variado el grado de oclusión de cada imagen transformada y se ha superpuesto sobre otra imagen de fondo (la finalidad de la foto de fondo es generar "outliers" en el "matching" inicial).

- Si RANSAC es capaz de detectar el objeto, se evalúa el error que se comete en la estimación de la matriz H. Para determinar dicho error se va a utilizar el mismo método descrito en el apartado anterior (mediante distancias Euclídeas entre las proyecciones reales y las proyecciones obtenidas tras aplicar la matriz H).

- Se va a representar varias gráficas que van a permitir analizar la influencia del grado de oclusión del objeto en la detección del mismo. Dichas gráficas son las siguientes:

  - Gráfica de la media y varianza del error. El error se calcula únicamente con las imágenes donde se ha detectado el objeto.
  - Gráfica de la probabilidad de detección en función del grado de oclusión.
  - Media del número de "inliers" y "outliers" en las imágenes en función del grado de oclusión.

- **Estudio del error utilizando la aproximación de cámara afín** - Se va a considerar que el objeto esta situado a suficiente distancia de la cámara de forma que las deformaciones que sufre dicho objeto se aproximan a deformaciones afines. De esta forma, para realizar el experimento, se le va a aplicar de forma sintética distintas

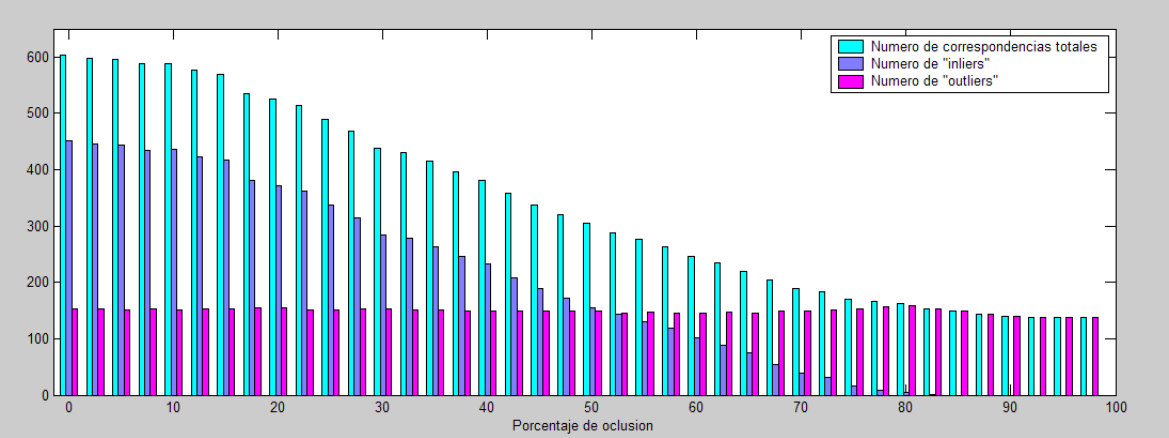


Figura 7.54: **Resultados y simulaciones.** *En está gráfica se representa el número medio de descriptores encontrados en la imagen junto con el número de "inliers" y de "outliers" totales.*

transformaciones afines a un objeto y se va a evaluar el error que se comete al utilizar RANSAC y la aproximación de cámara afín en la estimación de la matriz afín.

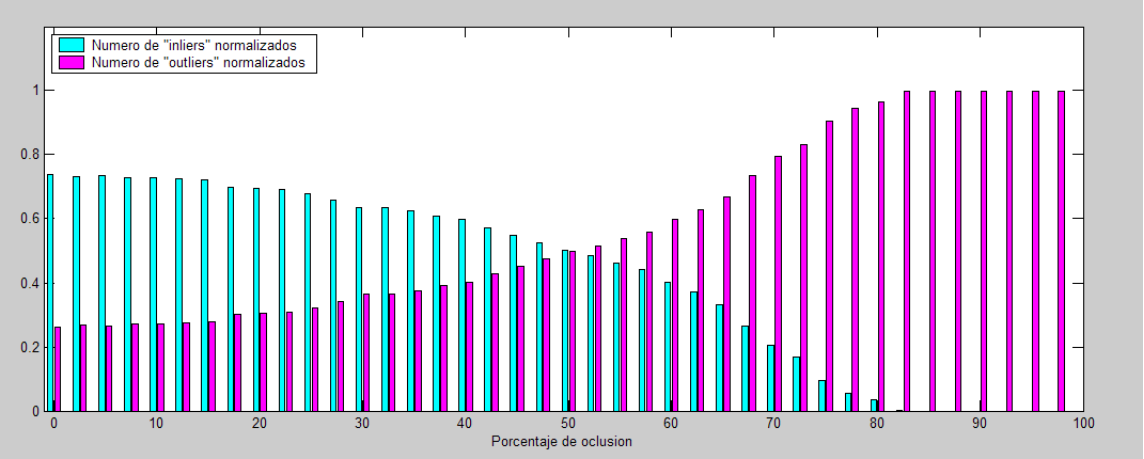


Figura 7.55: **Resultados y simulaciones.** *En está gráfica se representa el número medio de "inliers" y de "outliers" normalizados respecto del número total de descriptores encontrados en las imágenes.*

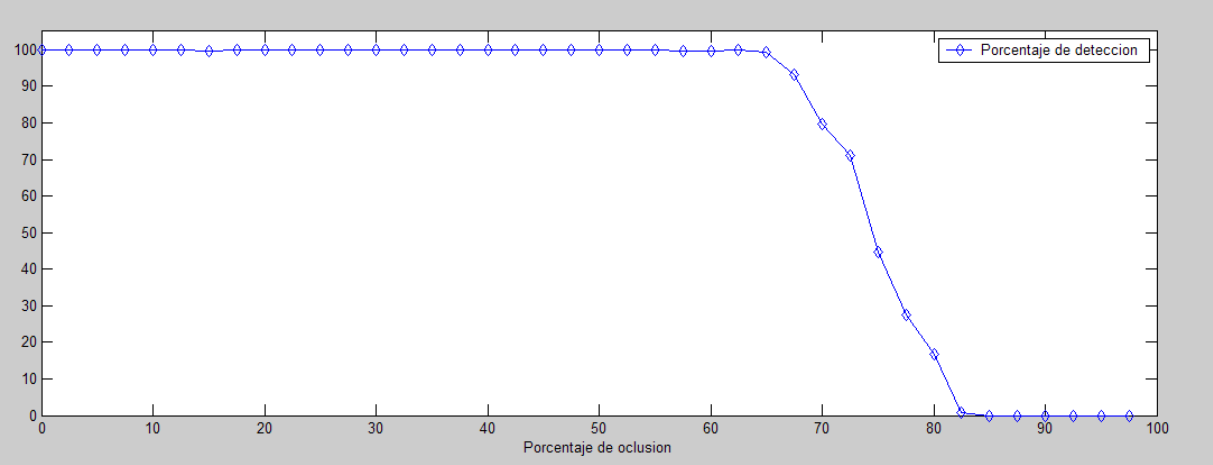


Figura 7.56: **Resultados y simulaciones.** *En esta gráfica se representa el porcentaje de detección en función del grado de oclusión.*

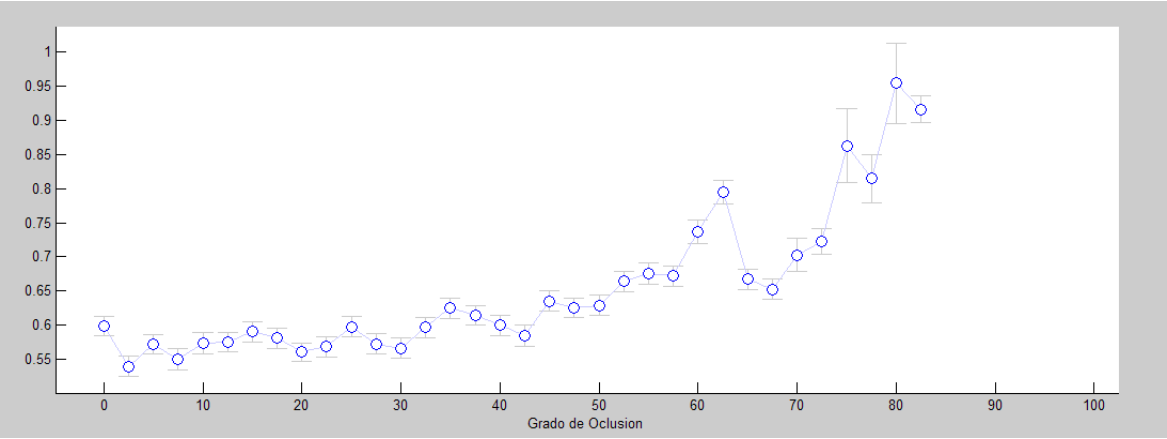


Figura 7.57: **Resultados y simulaciones.** *En esta gráfica se representa la media y la varianza de error que se comete en la estimación de la matriz H en función del grado de oclusión del objeto.*

En la figura 7.54 se puede ver como evolucionan el número de “inliers” y de “outliers” en función del grado de oclusión del objeto. A medida que el objeto se encuentra más ocluido, el número medio de descriptores SIFT que se encuentran en la imagen pertenecientes al objeto es menor por lo que el número medio de “inliers” estimados por RANSAC disminuye. El número de “outliers” permanece más o menos constante pues se corresponden con los “matching” entre los puntos del patrón y el fondo de la imagen. La proporción de “inliers” respecto del número total de

correspondencias en la imagen es mucho mayor que la de "outliers" para grado de oclusión de hasta un 50 %. Aproximadamente, para un 50 % de oclusión, el número de "inliers" se iguala al de "outliers", llegándose a anular prácticamente el número medio de "inliers" para un 80 % de oclusión.

En la gráfica 7.56 se puede observar como a partir de un 65 % aproximadamente el porcentaje de detección decae rápidamente, llegando a ser nulo para grados de oclusión mayores del 80 %. Esto e debe a que el grado de oclusión ya empieza a ser alto y el número de descriptores SIFT encontrados en la imagen pertenecientes al objeto es nulo o no supera en ocasiones el número mínimo necesario para considerar que la solución de la matriz H es correcta o incluso (recordemos que solo tendremos en cuenta las soluciones dadas por RANSAC con un número de "inliers" superior al 0.3 % del número total de descriptores del patrón).

El error medio se ha evaluado teniendo en cuenta sólo las imágenes en las que se ha detectado el objeto. Podemos observar en el figura 7.57 que el error que se comete es prácticamente insignificante y se mantiene en torno a un valor de $0{,}6$ píxeles. Para grados de oclusión mayores del 60 %, el error aumenta ligeramente pero siempre manteniendose por debajo de 1 píxel de error. Esto se debe a que el número de "inliers" disminuye mucho y por tanto, es lógico que la matriz H sea menos precisa pues se tienen menos datos redundantes para recalcular la matriz H. Aun así, el error es muy bajo.

Con todo esto se puede concluir que con la combinación de los descriptores SIFT y el algoritmo de RANSAC se consigue un sistema capaz de detectar objetos de forma robusta (con probabilidades de detección en torno al 100 %) incluso cuando el 70 % del objeto se encuentra oculto.

- **Estudio del error utilizando el modelo de cámara proyectiva** - El análisis que se va a realizar ahora es el mismo que en el caso anterior. La única diferencia es que ahora se va a considerar el caso general en el cual el modelo de la cámara es proyectiva y por tanto, las transformaciones que sufrirán los objetos serán proyectivas.

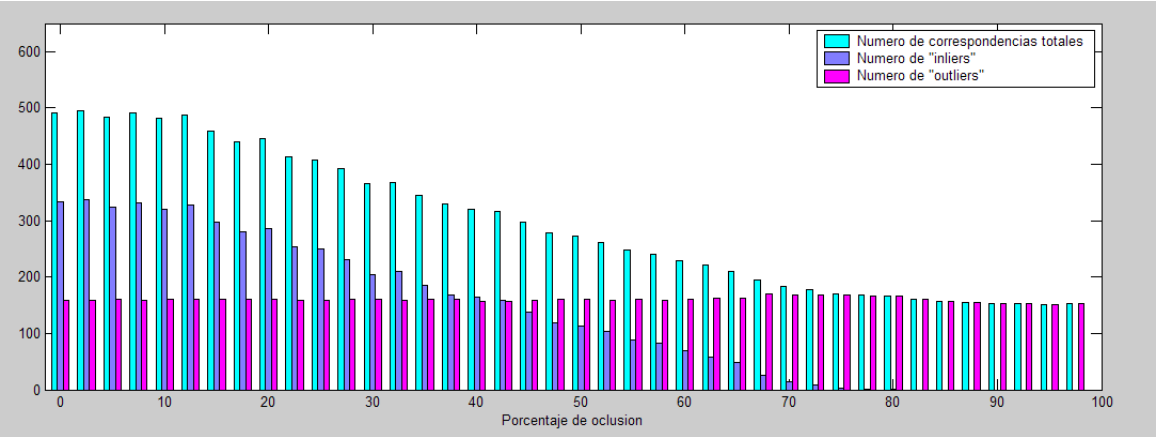


Figura 7.58: **Resultados y simulaciones.** *En está gráfica se representa el número medio de descriptores encontrados en la imagen junto con el número de "inliers" y de "outliers" totales.*

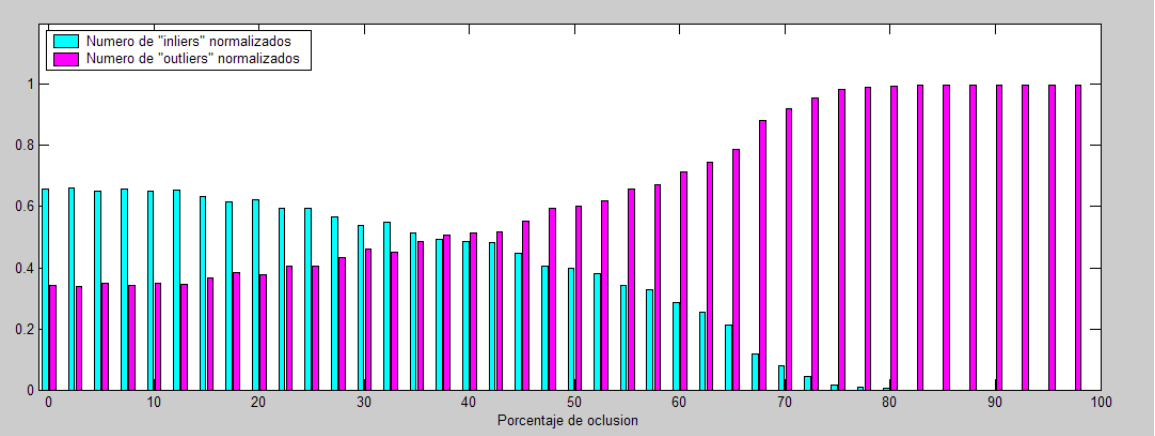


Figura 7.59: **Resultados y simulaciones.** *En está gráfica se representa el número medio de "inliers" y de "outliers" normalizados respecto del número total de descriptores encontrados en las imágenes.*

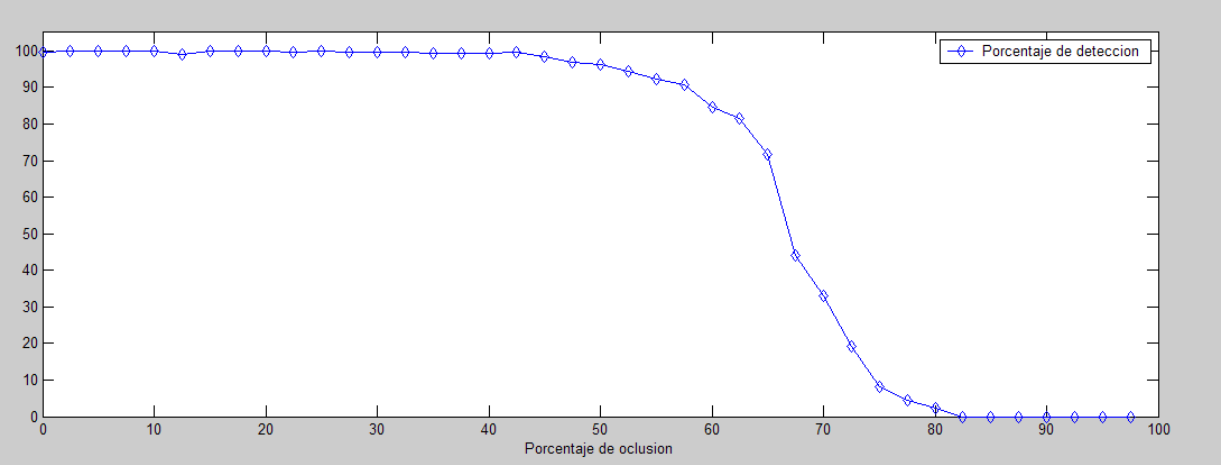


Figura 7.60: **Resultados y simulaciones.** *En esta gráfica se representa el porcentaje de detección en función del grado de oclusión.*

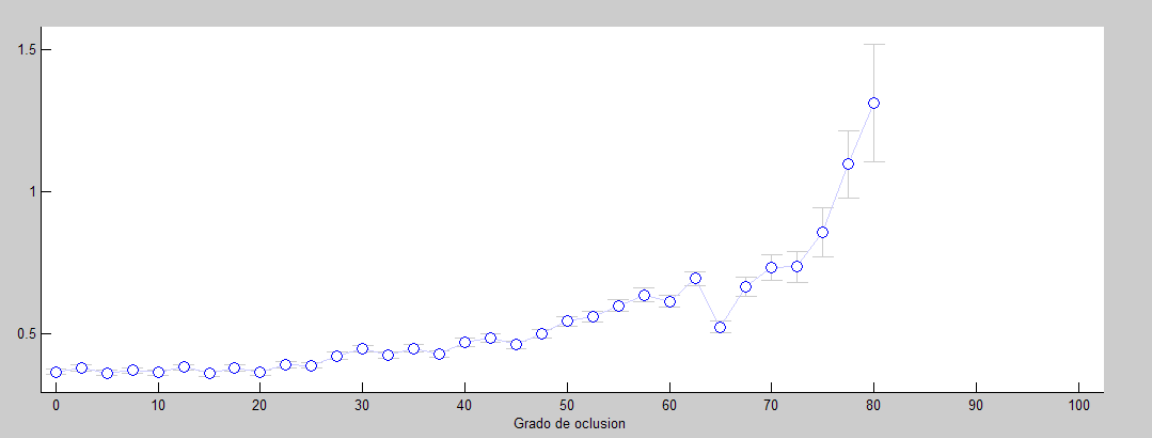


Figura 7.61: **Resultados y simulaciones.** *En esta gráfica se representa la media y la varianza de error que se comete en la estimación de la matriz H en función del grado de oclusión del objeto.*

Las conclusiones que se pueden sacar observando las gráficas anteriores son similares a las del caso de cámara afín. El número de "inliers" detectados es alto en comparación con el número de "outliers" para grados de oclusión inferiores al 40 %. A partir de este valor, la proporción de "inliers" va disminuyendo considerablemente llegando a ser prácticamente nulo en torno al 75 % u 80 %.

La probabilidad de detección es alta hasta aproximadamente un 60 % de oclusión decayendo rápidamente hasta anularse prácticamente para oclusiones mayores del

80 %. Esto se debe a que el número de "inliers" ha disminuido considerablemente llegando a ser en ocasiones inferior al número mínimo de puntos necesarios para considerar que la solución obtenida por RANSAC es válida.

Respecto al error, las conclusiones que se sacan son similares a las del caso anterior. Cuando el sistema detecta un objeto, el error que se comete es muy bajo independientemente del grado de oclusión. En este caso el error oscila entre $0{,}4$ y menos de $1{,}5$ píxeles de error.

Al igual que antes, el sistema es capaz de detectar objetos de forma robusta incluso cuando gran parte del objeto se encuantra ocluido (el porcentaje de detección esta por encima del 70 % para oclusiones de hasta el 65 %). Hay que resaltar que los resultados obtenidos para el caso general son un poco peores que los resultados para la aproximación afín. Esto se debe principalmente al algoritmo de RANSAC. Para el caso de cámara afín, el umbral del valor de distancia t es mucho más restrictivo que para el caso afín y cuando el número de correspondencias es muy bajo por el aumento del grado de oclusión, es posible que no todos estos puntos se adapten a la solución dada por RANSAC por lo que el número de "inliers" al final suele ser un poco inferior.

7.1.4. Estudio del error en función de la escala y el ángulo de deformación proyectiva

Dos factores que influyen notablemente en la detección robusta de objetos es la escala y el ángulo de deformación proyectiva. A medida que la distancia entre el objeto y la cámara aumenta (es decir, a medida que la escala disminuye) el número de descriptores SIFT de la imagen pertenecientes al objeto va disminuyendo. De la misma forma, a medida que la deformación proyectiva es mayor, la detección empeora. Además, es lógico pensar que los efectos de ambos factores en la detección no son independientes.

Para evaluar la influencia de la escala y la deformación proyectiva se va a realizar un experimento similar a los descritos en los apartados anteriores:

- Se toma una imagen de objeto planar del que se obtiene su modelo de apariencia.
- A dicho objeto se le va a aplicar una serie de transformaciones sintéticas variando de forma aleatoria el ángulo de rotación de Euler en el eje z (recordemos que el eje z se corresponde con el vector normal al objeto) y la posición del objeto en la imagen. La escala y el ángulo de deformación se va a variar de forma controlada, tomando todas las combinaciones posibles para ambos factores dentro de un rango de valores determinado.
- Para obtener el error de estimación, se va a utilizar de nuevo el error de reproyección.
- Las gráficas que se van a obtener son las mismas que las del apartado anterior. La única diferencia es que los resultados se van a representar en función de la escala y el ángulo de deformación proyectiva.

- **Estudio del error utilizando la aproximación de cámara afín** - En el capítulo 6 se vio que la matriz de afinidad queda definida por 6 parámetros:
  - K_x, K_y $\rightarrow$ son los factores de escala en el eje x y en el eje y respectivamente.
  - θ $\rightarrow$ es el ángulo de rotación de Euler en el eje z del sistema de referencia del objeto planar.
  - d_x, d_y $\rightarrow$ son las componentes del vector de desplazamiento en el eje x y en el eje y respectivamente.
  - s $\rightarrow$ es el parámetro que representa la deformación que se produce debido a los efectos de perspectiva en los que se rompe la ortogonalidad de los ejes. El parámetro s está relacionado con el ángulo de deformación entre la imagen y el eje ortogonal de la siguiente manera:

$$s = \tan \alpha$$

Utilizando esta parametrización, la matriz de afinidad queda de la siguiente forma:

$$H = \begin{pmatrix} K_x & 0 \\ s & K_y \end{pmatrix} \begin{pmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{pmatrix}$$

Para realizar esta simulación, los dos parámetros que nos interesan son la escala (se va a considerar que $K_x = K_y$) y el parámetro s. Se va a variar su valor generando todas las combinaciones posibles entre ambos parámetros. El rango de variación que se va a aplicar es el siguiente:

$$K \in [0{,}2, 1{,}1]$$

$$s \in [0, 0{,}85] \longrightarrow \alpha \in [0, 40]\text{grados}$$

Las gráficas obtenidas son las que se muestran a continuación:

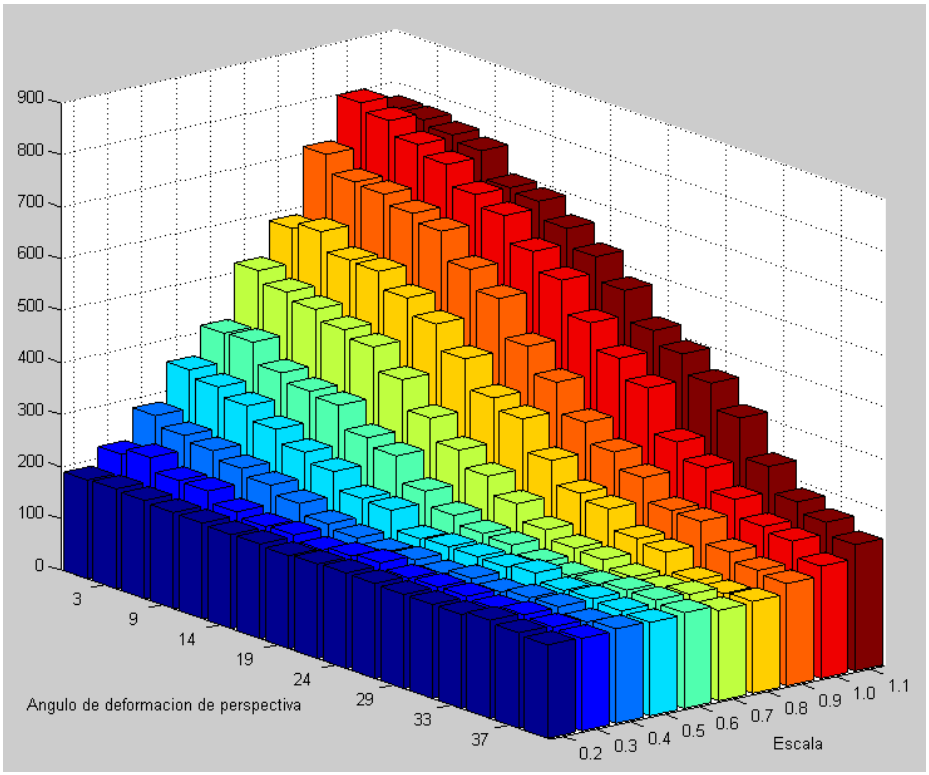


Figura 7.62: **Resultados y simulaciones.** *En está gráfica se representa el número medio de correspondencias iniciales encontradas en las imágenes para la aproximación afín en función de la escala y el ángulo de deformación proyectiva.*

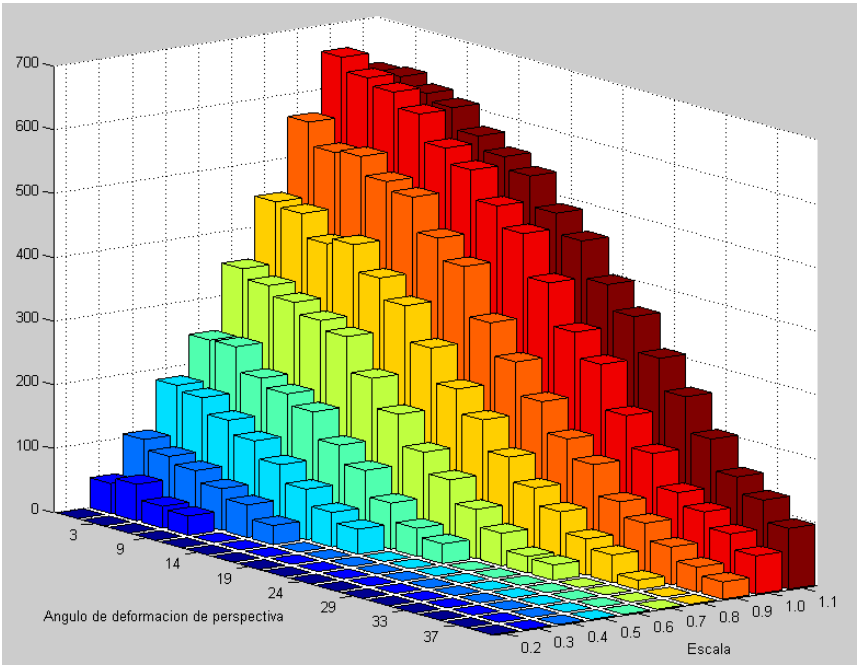


Figura 7.63: **Resultados y simulaciones.** *En está gráfica se representa el número medio de "inliers" encontrados en la imagen para un modelo de cámara afín en función de la escala y el ángulo de deformación proyectiva .*

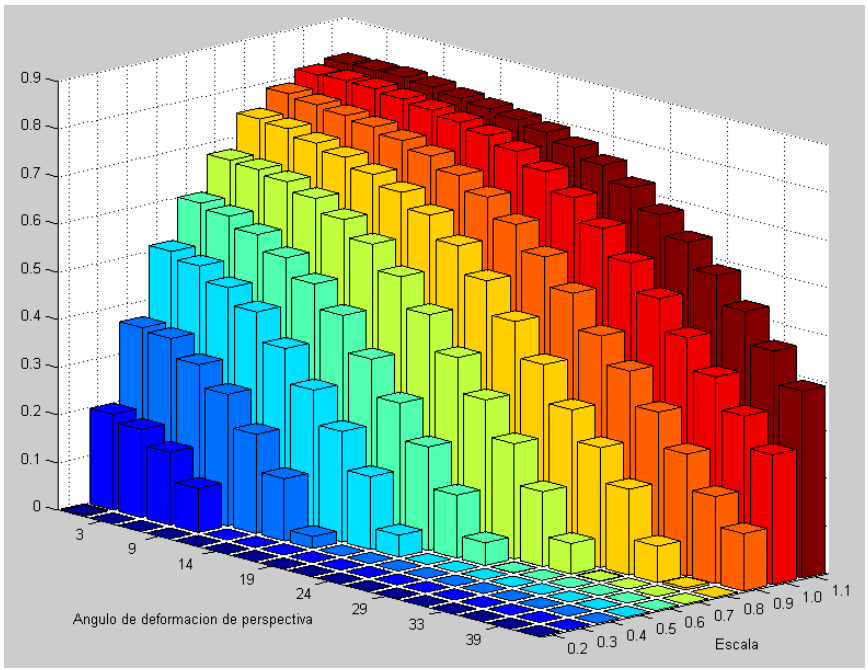


Figura 7.64: **Resultados y simulaciones.** *En está gráfica se representa el número medio de "inliers" y de "outliers" normalizados respecto del número total de descriptores encontrados en las imágenes.*

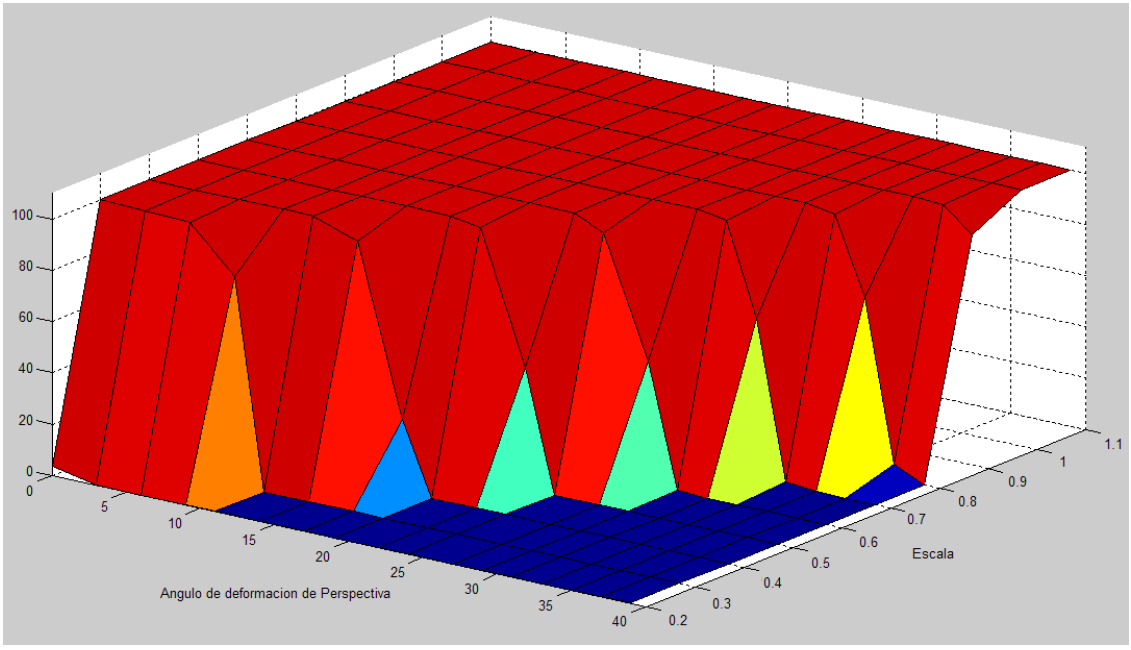


Figura 7.65: **Resultados y simulaciones.** *En esta gráfica se representa el porcentaje de detección en función de la escala y el ángulo de deformación proyectiva.*

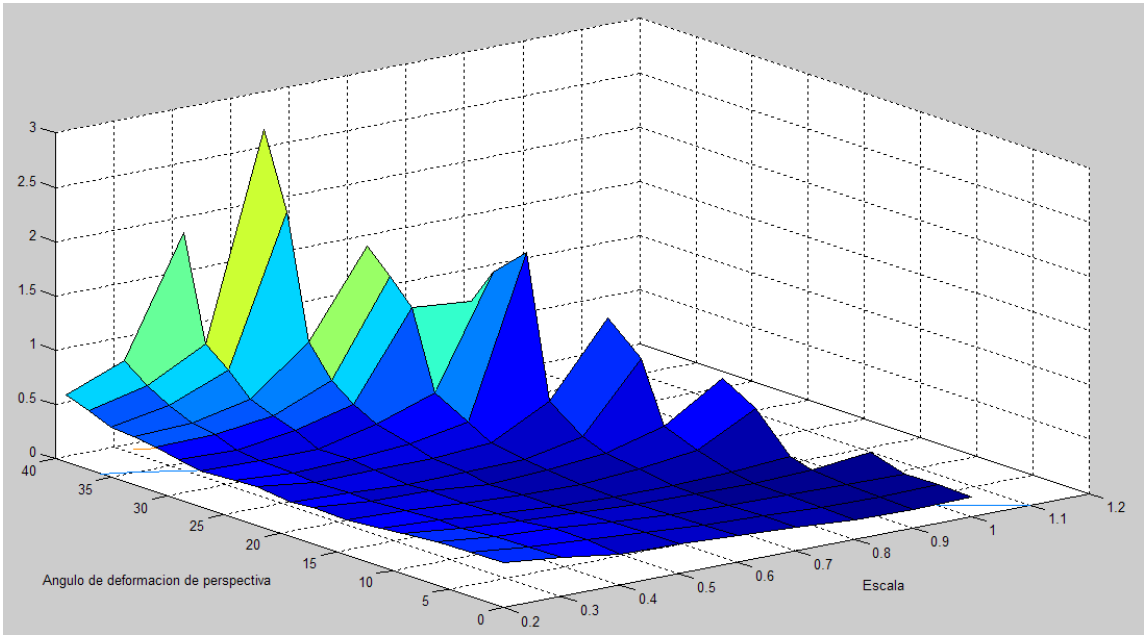


Figura 7.66: **Resultados y simulaciones.** *En esta gráfica se representa la media de error que se comete en la estimación de la matriz H en función de la escala y el ángulo de deformación proyectiva.*

En la figura 7.63 se puede apreciar como el número de correspondencias iniciales decrece hasta alcanzar un valor constante al disminuir la escala del objeto. Los descriptores SIFT tienen una fuerte dependencia con la escala (al fin y al cabo, este método se basa en la búsqueda de puntos invariantes aplicando distintas escalas a la imagen y cuando la escala disminuye mucho, la resolución del objeto empeora) y al disminuir la escala, el número de descriptores que se detectan en una imagen decrece, llegando a ser nulo cuando la escala sobrepasa un cierto valor mínimo (en esta situación, todos los puntos del "matching" inicial se corresponden con "outlier")

Lo mismo ocurre con la deformación proyectiva. A medida que el ángulo de deformación aumenta, el número de correspondencias iniciales decae pues el número de descriptores detectados en la imagen disminuye y además, su valor se degrada bastante. Estos descriptores se obtienen con la información local de una área reducida en torno al punto de interés, por tanto, mientras la deformación proyectiva no sobrepase un cierto valor, se puede considerar que esta área reducida se ve afectada únicamente por transformaciones afines.

Además, se puede observar que los efectos que producen ambos parámetros en la detección de objetos no son independientes. Para una escala de $0{,}3$, el porcentaje de detección decae bruscamente hasta ser nulo para ángulo de deformación mayores de unos 15 grados aproximadamente (se corresponde con un valor de s cercano a $0{,}3$). A media que aumenta la escala, el ángulo permisible que garantiza una detección robusta va aumentando, pudiéndose aplicar deformaciones de hasta 40 grados para un factor de escala del $0{,}8$

Respecto al error de detección, se puede observar que si el sistema es capaz de detectar el objeto, el error de reproyección es muy pequeño (no supera el píxel de error) Cuando la deformación del objeto debido tanto a la escala como a la propia deformación proyectiva es grande, el porcentaje de detección decae bruscamente y el error de estimación de los objetos que el sistema es aun capaz de detectar aumenta un poco (en torno a 3 píxeles de error) Esto se debe a que no solo se degrada el valor de los descriptores sino que también se pierde exactitud a la hora de obtener su localización.

- **Estudio del error utilizando el modelo de cámara proyectiva** - Ahora se va a considerar que el objeto puede sufrir deformaciones proyectivas. Para realizar la simulación, se va a considerar que el objeto puede sufrir únicamente rotaciones en el eje x y en el eje z. El ángulo de Euler en el eje z será aleatorio mientras que para el eje x se va a considerar que el ángulo de Euler puede tomar valores desde 0 a 60 grados. A su vez se va a considerar que el factor de escala varía entre 0.2 y 1.2.

  Las gráficas obtenidas son las que se muestran a continuación:

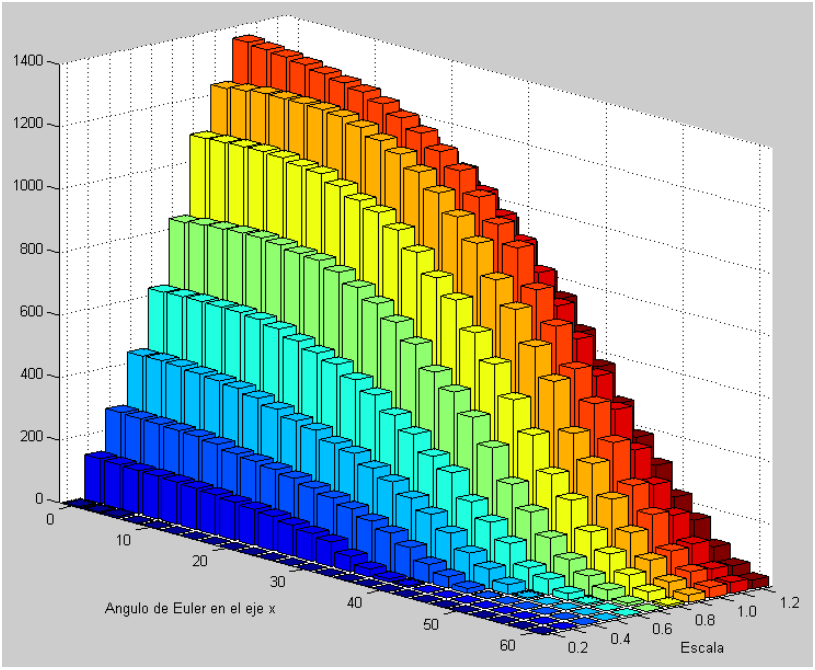


Figura 7.67: **Resultados y simulaciones.** *En está gráfica se representa el número medio de correspondencias iniciales encontradas en las imágenes para la aproximación afín en función de la escala y el ángulo de deformación proyectiva.*

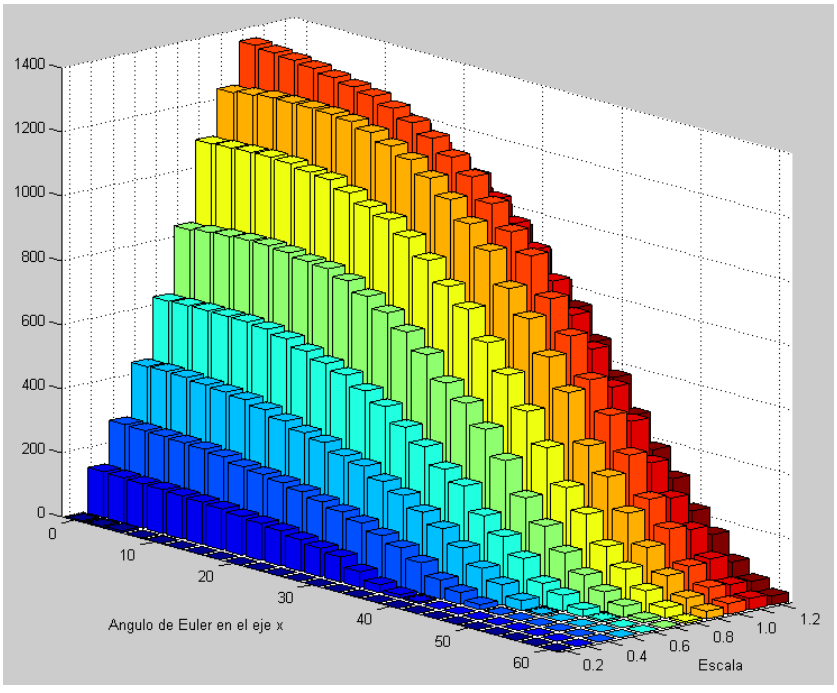


Figura 7.68: **Resultados y simulaciones.** *En está gráfica se representa el número medio de "inliers" encontrados en la imagen para un modelo de cámara afín en función de la escala y el ángulo de deformación proyectiva .*

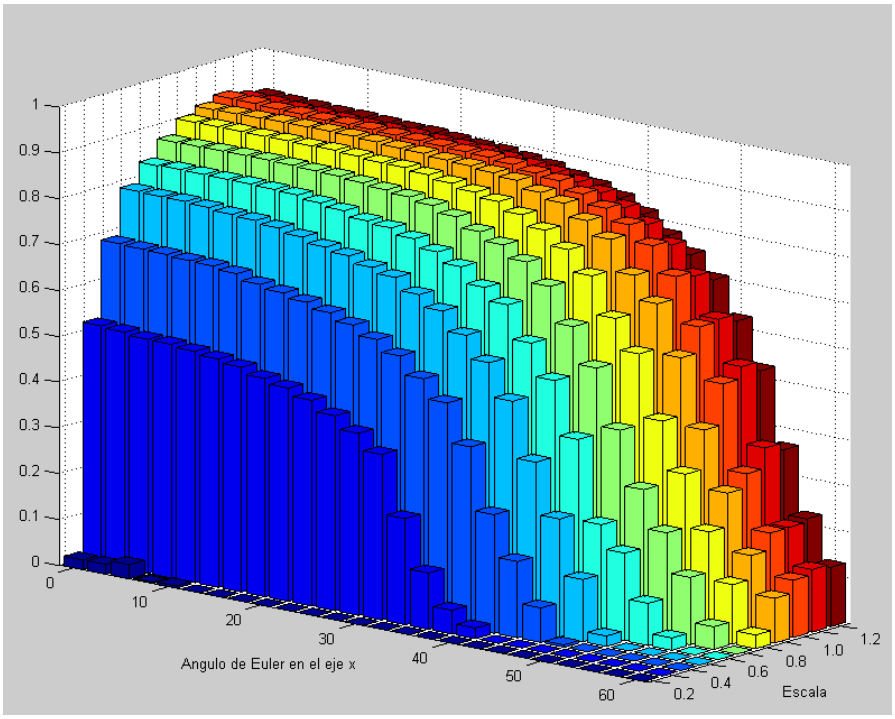


Figura 7.69: **Resultados y simulaciones.** *En está gráfica se representa el número medio de "inliers" y de "outliers" normalizados respecto del número total de descriptores encontrados en las imágenes.*

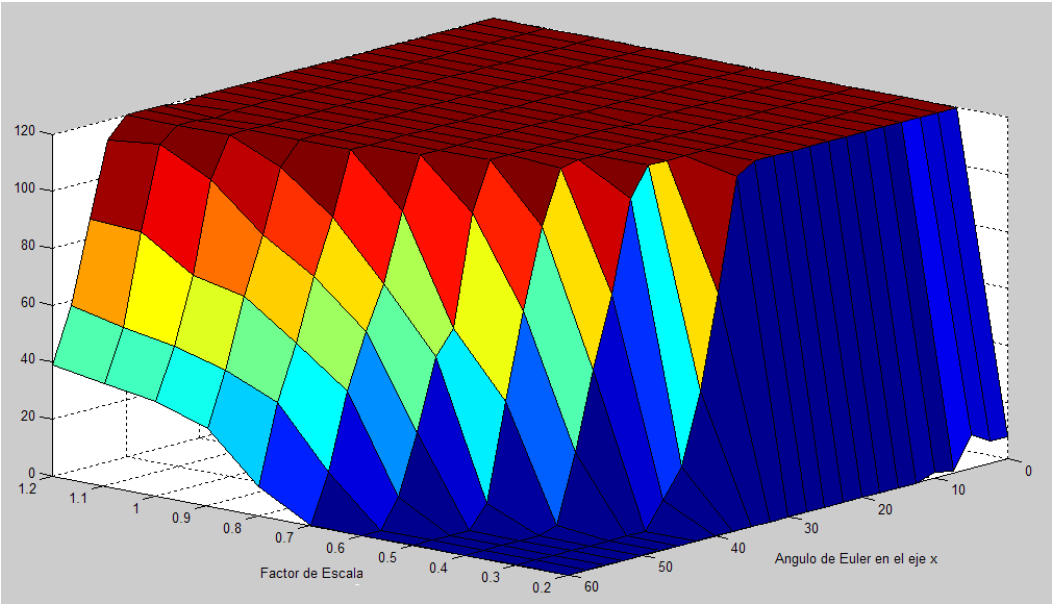


Figura 7.70: **Resultados y simulaciones.** *En esta gráfica se representa el porcentaje de detección en función de la escala y el ángulo de deformación proyectiva.*

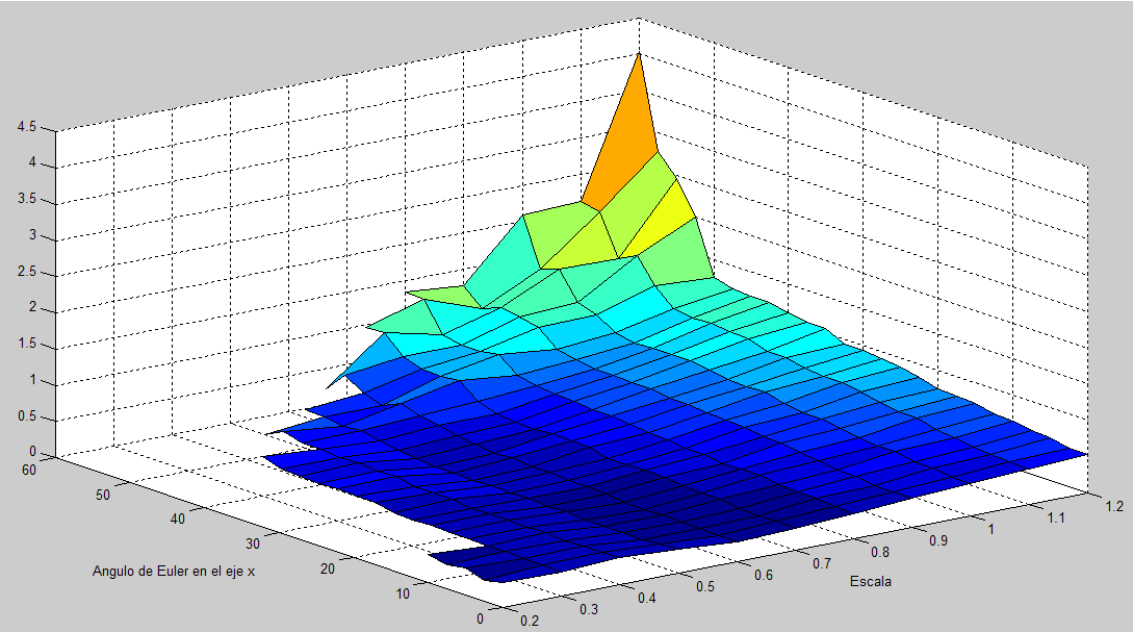


Figura 7.71: **Resultados y simulaciones.** *En esta gráfica se representa la media y la varianza de error que se comete en la estimación de la matriz H en función de la escala y el ángulo de deformación proyectiva.*

Los resultados obtenidos son similares al caso anterior. A medida que disminuye la escala, el número de descriptores disminuye por lo que la probabilidad de detección decrece. Lo mismo ocurre a medida que aumenta el ángulo de Euler.

Cuando el factor de escala está en torno a $0{,}3$, el ángulo máximo de rotación que se puede aplicar al objeto para garantizar una detección con una probabilidad alta esta en torno a los 30 grados. A medida que aumenta la distancia, el ángulo aumenta. Para una escala de 0.9, se pueden aplicar ángulos de 60 grados garantizando una detección del 100 %. Estos resultados son bastante buenos, teniendo en cuenta que el ángulo máximo que se puede aplicar al objeto es 90 grados (en esta situación el objeto deja de ser visible pues su posición será perpendicular al plano imagen).

En la gráfica del
error de reproyección, se puede ver que, al igual que antes, el error de estimación es pequeño (en torno a medio píxel) Este error va aumentando a medida que la deformación del objeto aumenta (para los casos límites de escala y ángulo de rotación donde el porcentaje de detección decae bruscamente).

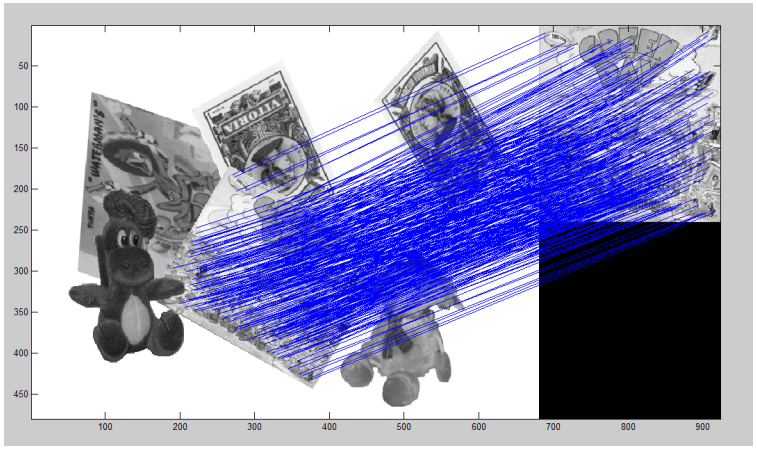
7.1.5. Estudio del error en función del ruido

El ruido es otro factor que puede influir notablemente en la detección. Al introducir ruido en una imagen, el valor de los descriptores cambian con respecto al valor que tendrían en condiciones normales. Esto puede afectar al "matching" inicial. También es posible que el ruido influya en la posición que ocupan los descriptores en la imagen y por tanto el error al calcular la matriz H aumente.

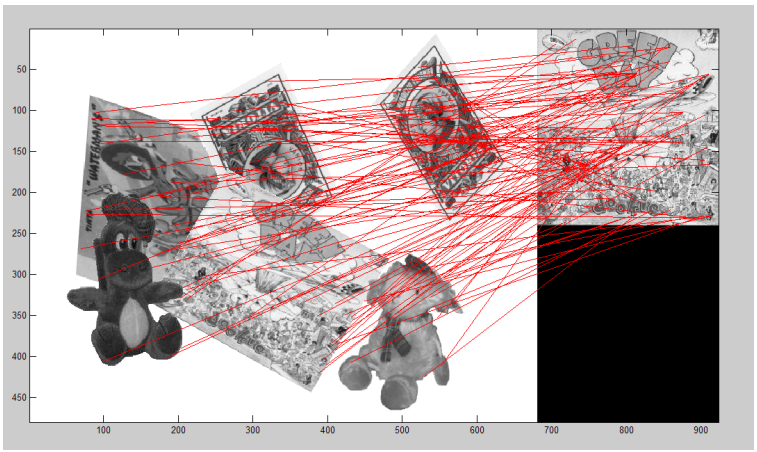
Para analizar la influencia del ruido, se ha realizado el siguiente experimento:

- Se ha tomado una imagen de objeto planar del que se ha obtenido su modelo de apariencia.
- Al igual que en el resto de experimentos comentados con anterioridad, se le ha aplicado una serie de transformaciones sintéticas a la imagen patrón (afines para el caso de cámara afín y proyectivas para el caso general de cámara proyectiva) e incluimos un fondo en el que aparecen otros objetos.
- A cada imagen generada se le va a añadir ruido blanco gaussiano aditivo de distinta varianza, desde 0 a 45 (se va a trabajar con imágenes en escala de grises de 0 a 255).
- Para evaluar el error que se comete con RANSAC a la hora de estimar la matriz H, se va a utilizar el mismo método que en los apartados anteriores. Las gráficas que se van a obtener son las siguientes:
  - Gráfica de la media y varianza del error. El error se calcula únicamente con las imágenes donde se ha detectado el objeto.
  - Gráfica de la probabilidad de detección en función de la varianza de ruido.
  - Media del número de "inliers" y "outliers" en las imágenes en función de la varianza de ruido.

- **Estudio del error utilizando la aproximación de cámara afín** - Para este caso, se va a aplicar transformaciones afines al objeto que se desea detectar. En las figuras 7.72 y 7.73 se muestran dos ejemplos utilizados en el experimento, la primera de ellas se corresponde con una varianza de ruido nula y la segunda con una $\sigma = 30$.



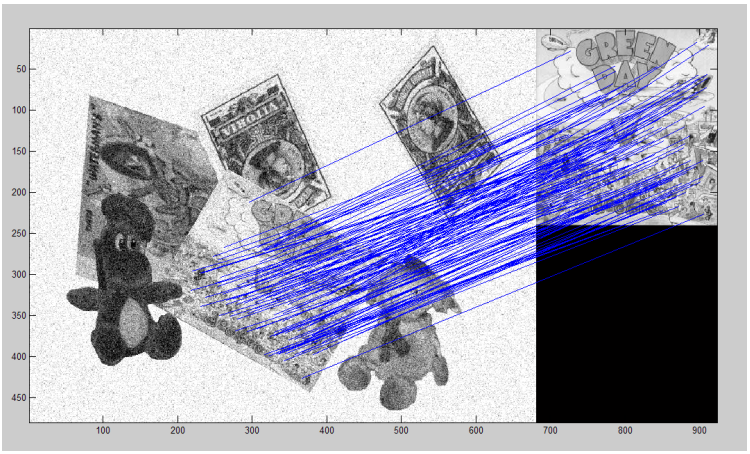
(a)



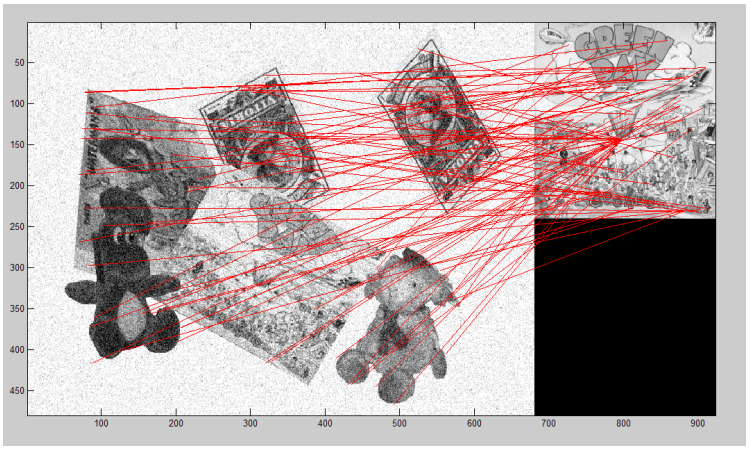
(b)

(c)

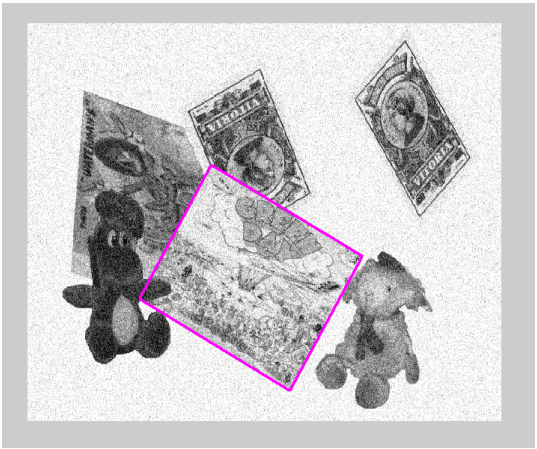
Figura 7.72: **Resultados y simulaciones.** *En estas gráficas se muestra un ejemplo de la influencia del ruido en la detección de objetos. En concreto, el ruido aplicado en este caso es de varianza nula. (a) Conjunto de "inliers" detectados por RANSAC. (b) Falsas correspondencias. (c) Estima del contorno del objeto detectado.*



(a)



(b)



(c)

Figura 7.73: **Resultados y simulaciones.** *En estas gráficas se muestra un ejemplo de la influencia del ruido en la detección de objetos. En concreto, el ruido aplicado en este caso es de varianza igual a 30. (a) Conjunto de "inliers" detectados por RANSAC. (b) Falsas correspondencias. (c) Estima del contorno del objeto detectado*

Las gráficas que hemos obtenido para evaluar el comportamiento del algoritmo RANSAC para la aproximación de cámara afín son las siguientes:

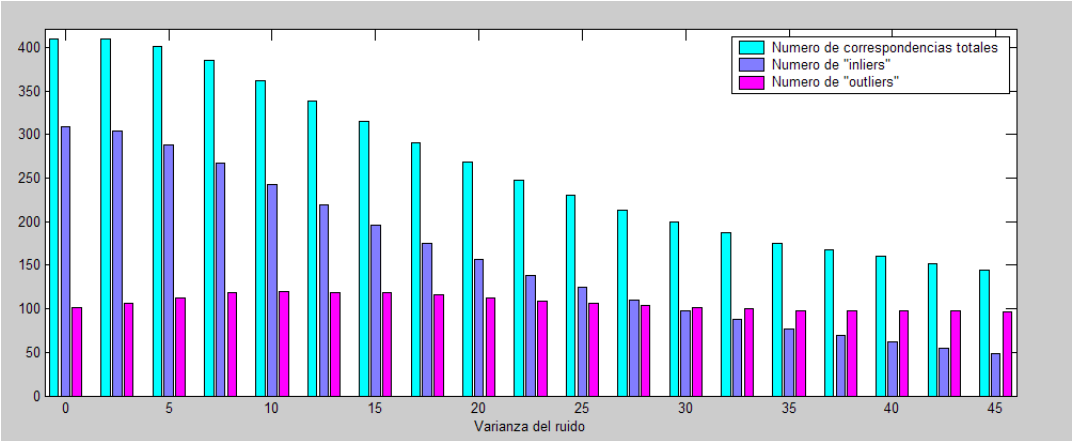


Figura 7.74: **Resultados y simulaciones.** *En está gráfica se representa el número medio de descriptores encontrados en la imagen junto con el número de "inliers" y de "outliers" totales.*

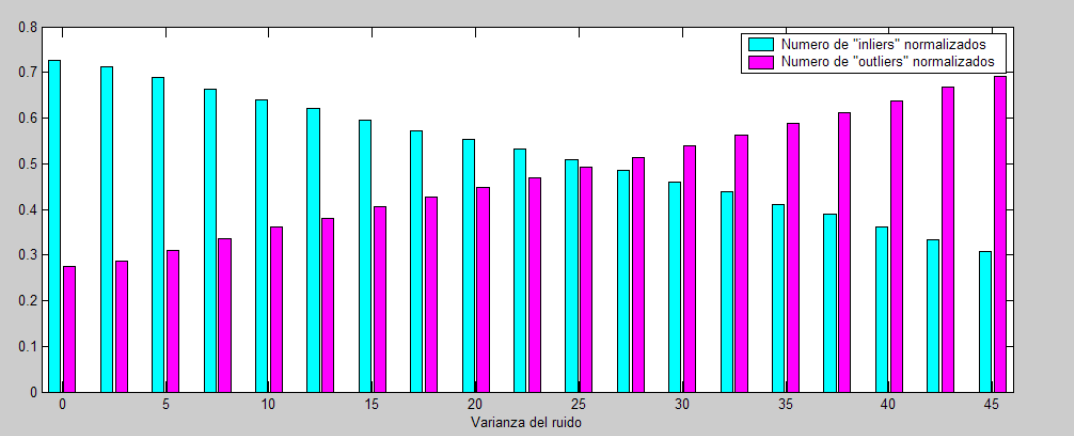


Figura 7.75: **Resultados y simulaciones.** *En está gráfica se representa el número medio de "inliers" y de "outliers" normalizados respecto del número total de descriptores encontrados en las imágenes.*

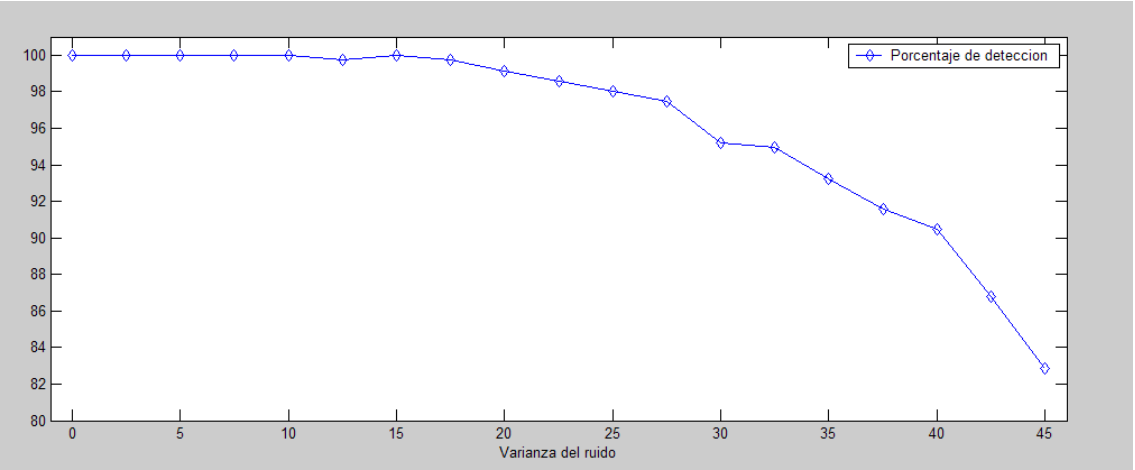


Figura 7.76: **Resultados y simulaciones.** *En esta gráfica se representa el porcentaje de detección en función de la varianza de ruido.*

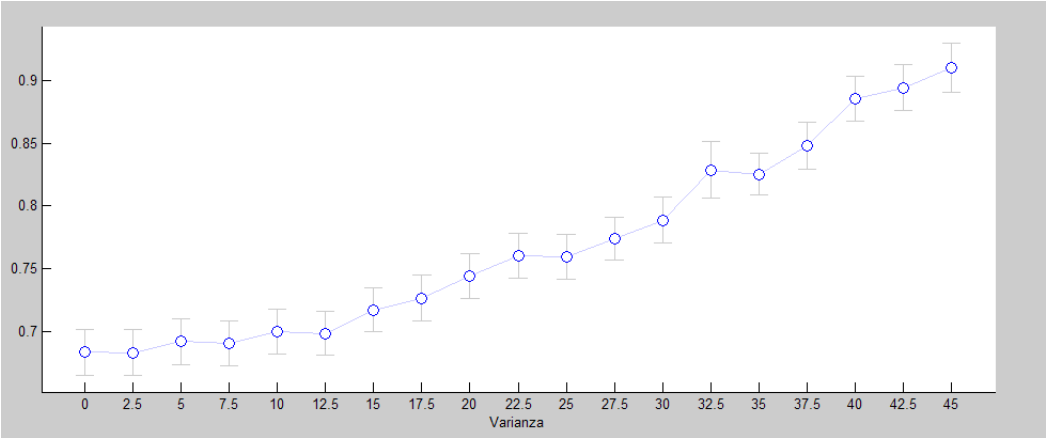


Figura 7.77: **Resultados y simulaciones.** *En esta gráfica se representa la media y la varianza de error que se comete en la estimación de la matriz H en función de la varianza de ruido.*

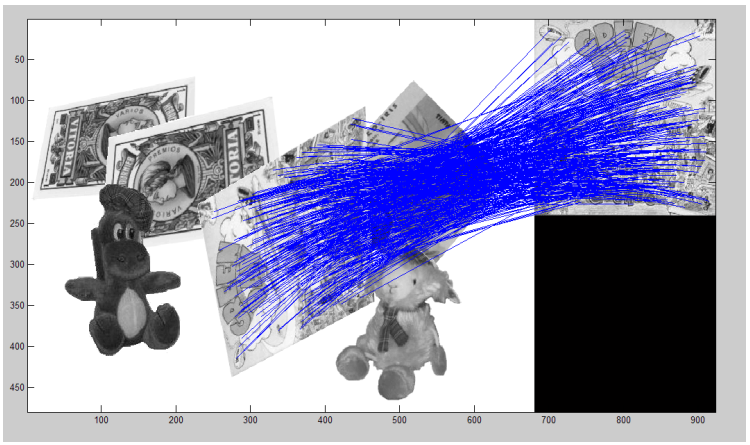
En la gráfica 7.74 se puede observar la evolución del número de “inliers” y de “outliers” a media que aumenta la varianza de ruido. Para valores de σ bajos, la proporción de “inliers” respecto del número total de correspondencias encontradas en la imagen es mucho mayor que la de “outliers” (aproximadamente para valores de varianza superiores a 25 la proporción de “inliers” es menor), por lo que la probabilidad de detección es alta, aproximadamente del 100 %. Comparando las figuras 7.72 y 7.73 se puede apreciar la disminución de “inliers” al aumentar el ruido.

Sería lógico pensar que a medida que aumenta la varianza, la influencia del ruido en los descriptores SIFT es muy grande de forma que el valor de los mismos se desviará de su valor real. En esta situación, seguramente el número de correspondencias iniciales correctas sería muy pequeño o incluso nulo comparado con un caso en el que el ruido es de varianza casi nula, por lo que la probabilidad de detección decaería. Sin embargo, en los resultados obtenidos se puede apreciar que el método SIFT presenta un comportamiento muy bueno frente al ruido. Es verdad que el número de correspondencias iniciales pertenecientes al objeto que se desea detectar va disminuyendo pero aun así, el número es bastante mayor comparado con el mínimo necesario para detectar el objeto. Debido a esto, el porcentaje de detección es alto incluso para los casos en los que la σ es grande (por ejemplo, para $\sigma = 45$, que se corresponde con una señal de ruido considerable teniendo en cuenta que trabajamos con imágenes en escala de grises de 0 a 255, el porcentaje de detección es superior al 80 %).

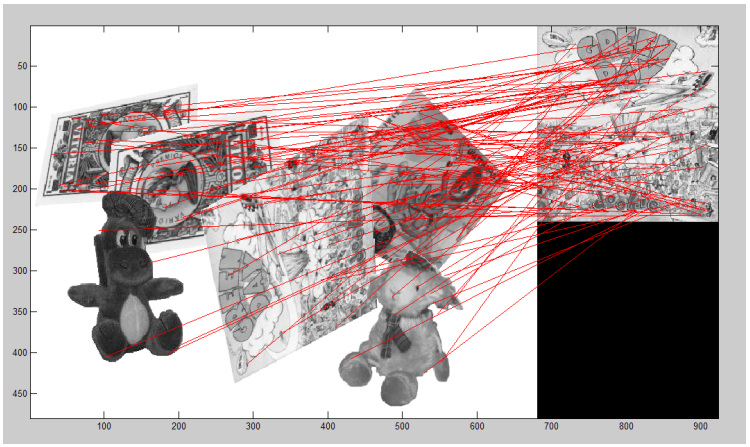
Además del porcentaje de detección, tenemos que tener en cuenta el error que se comete en la estimación de la matriz de afinidad. Al igual que pasaba en otros experimentos, el error que se comete es muy bajo. Aunque a medida que aumenta la señal de ruido, el error crece, éste se mantiene por debajo de 1 píxel.

Por tanto, con este experimento se puede concluir que el comportamiento de los descriptores SIFT frente al ruido es muy bueno. El número de descriptores que se detectan inicialmente con el método SIFT va disminuyendo con el ruido pero aun así, el número es bastante alto para realizar la detección. Además, los descriptores que encuentra resultan ser robustos.

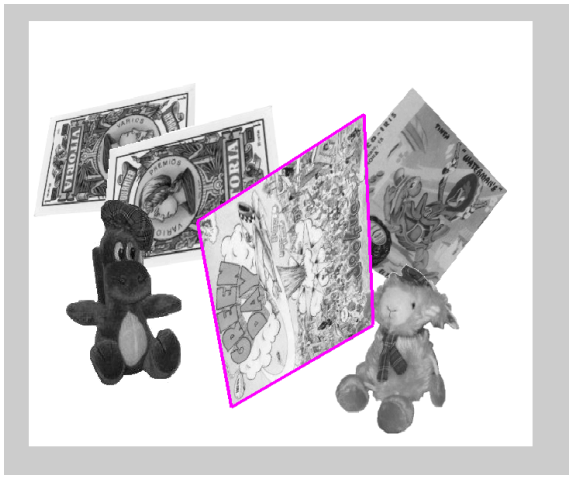
- **Estudio del error utilizando el modelo de cámara proyectiva** - Ahora se va a analizar el caso más general en el cual se calcula mediante RANSAC la matriz de homografía de los objetos. Al igual que antes, en las figuras 7.78 y 7.79 se muestran dos ejemplos utilizados en el experimento, pero ahora se les ha aplicado a los objetos deformaciones proyectivas. La primera de ellas se corresponde con una varianza de ruido nula y la segunda con una $\sigma = 35$.



(a)

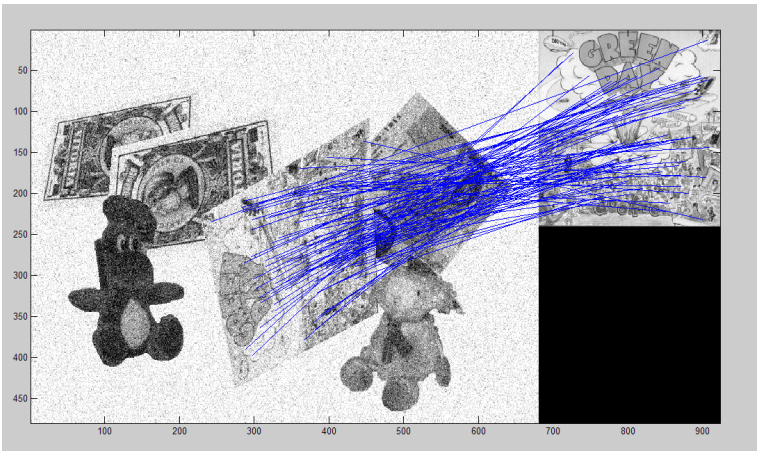


(b)

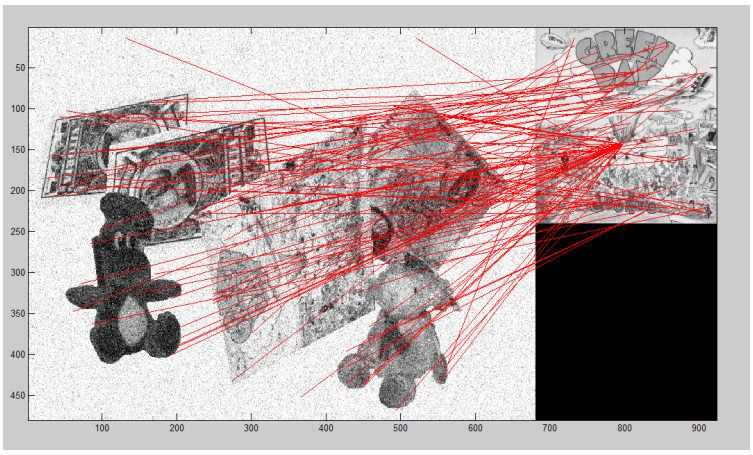


(c)

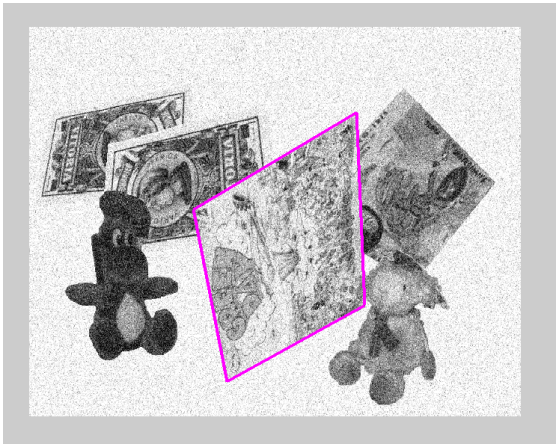
Figura 7.78: **Resultados y simulaciones.** *En estas gráficas se muestra un ejemplo de la influencia del ruido en la detección de objetos. En concreto, el ruido aplicado en este caso es de varianza nula. (a) Conjunto de "inliers" detectados por RANSAC. (b) Falsas correspondencias. (c) Estima del contorno del objeto detectado.*



(a)



(b)



(c)

Figura 7.79: **Resultados y simulaciones.** *En estas gráficas se muestra un ejemplo de la influencia del ruido en la detección de objetos. En concreto, el ruido aplicado en este caso es de varianza igual a 35. (a) Conjunto de "inliers" detectados por RANSAC. (b) Falsas correspondencias. (c) Estima del contorno del objeto detectado*

Las gráficas que hemos obtenido para evaluar el comportamiento del algoritmo RANSAC para el caso general de cámara proyectiva son las siguientes:

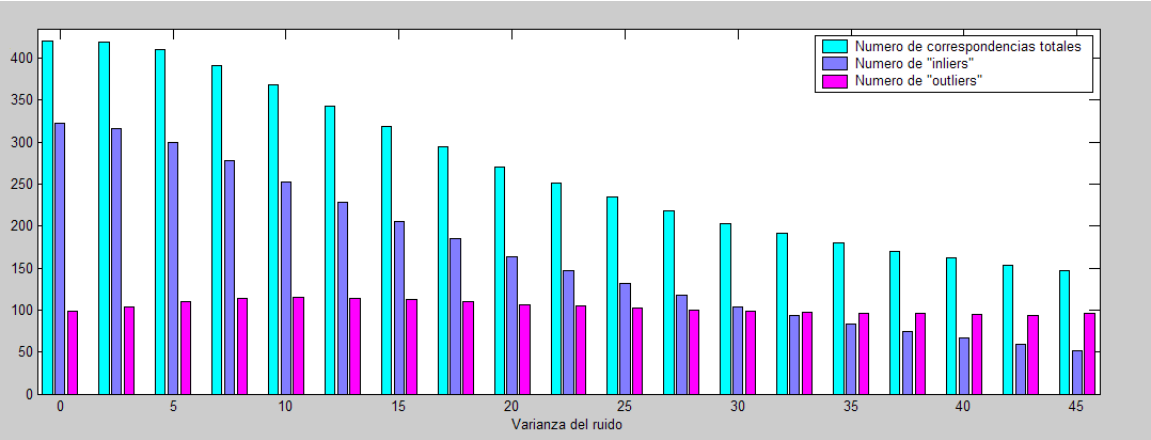


Figura 7.80: **Resultados y simulaciones.** *En está gráfica se representa el número medio de descriptores encontrados en la imagen junto con el número de "inliers" y de "outliers" totales. Se ha utilizado RANSAC junto con un modelo de cámara proyectiva.*

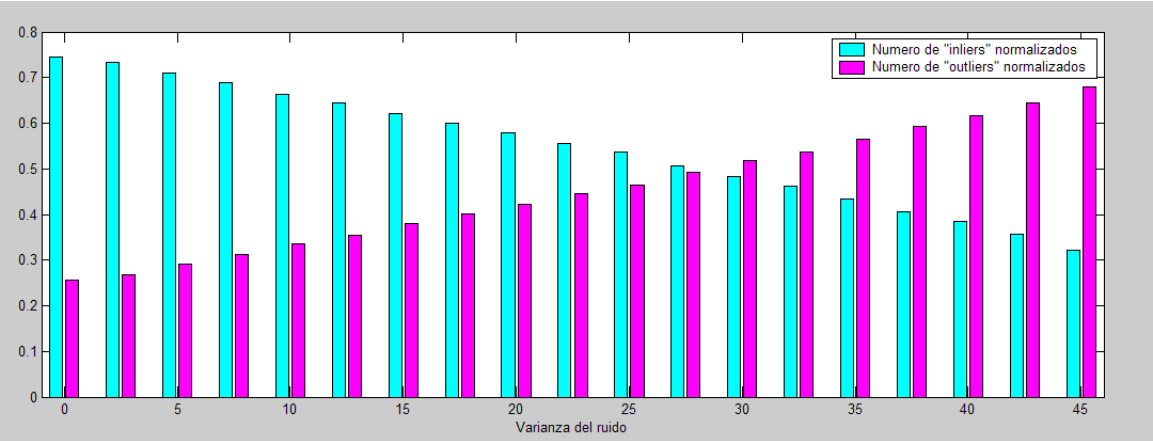


Figura 7.81: **Resultados y simulaciones.** *En está gráfica se representa el número medio de "inliers" y de "outliers" normalizados respecto del número total de descriptores encontrados en las imágenes. Se ha utilizado RANSAC junto con un modelo de cámara proyectiva.*

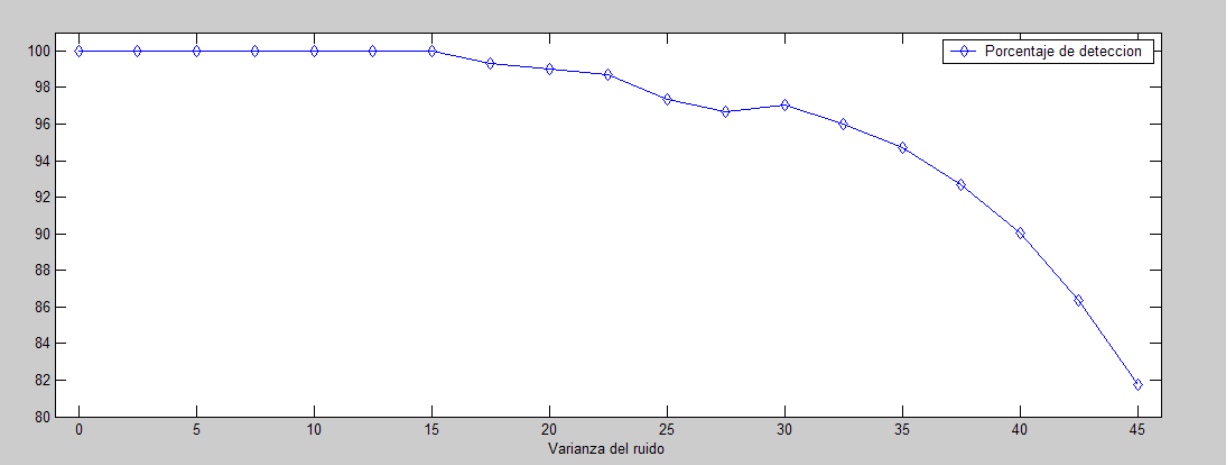


Figura 7.82: **Resultados y simulaciones.** *En esta gráfica se representa el porcentaje de detección en función de la varianza de ruido.*

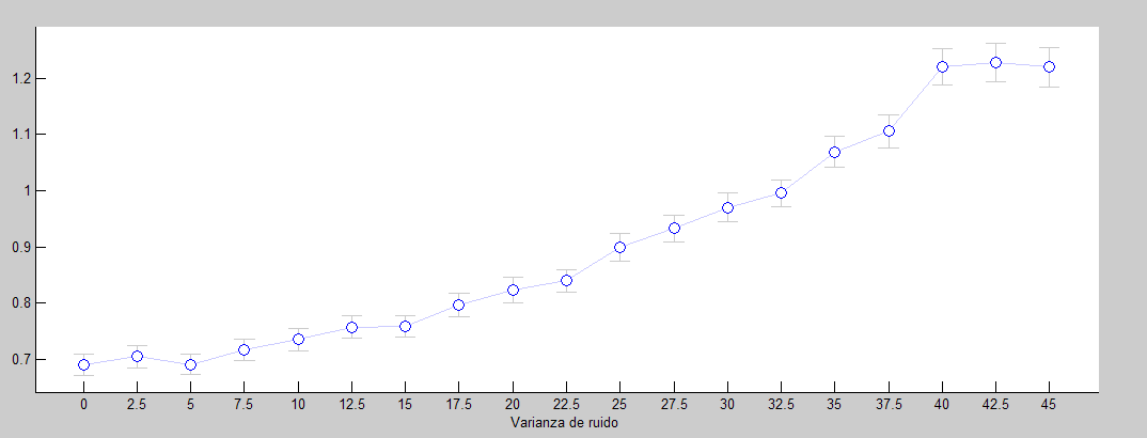


Figura 7.83: **Resultados y simulaciones.** *En esta gráfica se representa la media y la varianza de error que se comete en la estimación de la matriz H en función de la varianza de ruido.*

Al igual que pasaba para el caso de afinidad, se obtienen muy buenos resultados en la detección de objetos en imágenes con varianza de ruido alta. El número de descriptores decrece al aumentar el ruido, pero aun así, sigue siendo lo suficientemente alto como para detectar los objetos. Además, de los resultados obtenidos se puede deducir que, en presencia de ruido, los descriptores que se consiguen detectar presentan bastante inmunidad, pues el método SIFT es capaz de emparejar puntos correctamente en el proceso de "matching" por lo que el valor de cada descriptor

detectado no debe desviarse mucho de su valor real.

Debido a todo esto, el porcentaje de detección es alto incluso para los casos en los que la σ es grande. Al igual que pasaba antes, para un valor de $\sigma = 45$, el porcentaje de detección es superior al $80\,\%$.

Además, el error medio que se produce en la estimación de la matriz H es pequeño pues el error de reproyección calculado en función de la varianza de ruido es bastante bajo. Aunque a medida que aumenta la señal de ruido, el error crece, éste se mantiene por debajo de $1{,}2$ píxeles. Con esto también se puede concluir que la posición de los descriptores no varía con el ruido.

7.1.6. Detección de múltiples objetos

Hasta ahora en todos los experimetos que se han realizado, se han utilizado imágenes en las que sólo aparecía un único objeto. Se ha podido demostrar que el método RANSAC proporciona muy buenos resultados en la detección de un sólo objeto incluso en los casos en los que el número de "outliers" es elevado. A continuación, se va a mostrar varios ejemplos en los que hay más de un objeto repetido en la imagen para avaluar el comportamiento del algoritmo de RANSAC en estos casos. Hay que remarcar que se está usando la modificación propuesta en el apartado **??** del capítulo **??**.

Las imágenes que se van a utilizar para las simulaciones están tomadas a una distancia en la que se puede utilizar la aproximación afín. Puesto que lo que se quiere evaluar en estos ejemplos es el comportamiento del algoritmo RANSAC para múltiples objetos, se va a utilizar únicamente el método de RANSAC junto con la aproximación afín, pues los resultados con ambos métodos son similares (el algoritmo de RANSAC es el mismo, lo único que varía es la forma de calcular la solución de las matrices).

A continuación, se muestran dos ejemplos. Además de las imágenes donde se muestran las correspondencias iniciales, los "inliers", los "outliers" y los perfiles generados en el proceso de reproyección, se va a representar una gráfica en la que se muestra el número de "inliers" detectados en cada iteración del algoritmo de RANSAC, el número total de "inliers´´ detectados hasta el momento y el número total de puntos considerados como falsas correspondencias después de cada iteración.

Con ambos ejemplos se puede comprobar que el algoritmo de RANSAC para múltiples objetos funciona correctamente. En la primera iteración de cada ejemplo se puede ver que el algoritmo es capaz de detectar un objeto con un porcentaje de "outliers" superior al 50 %. Aunque este porcentaje es mucho menor al finalizar la detección, hay que tener en cuenta que en cada iteración en la que se encuentra una solución, los "inliers" de los objetos restantes de la imagen que aun no han sido detectados son también "outliers".

- **Primer ejemplo**

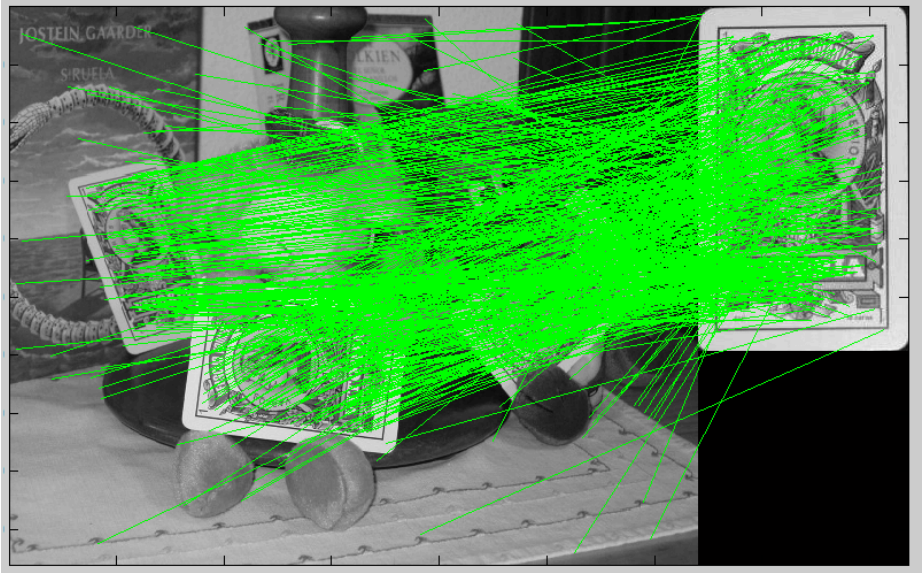


Figura 7.84: **Resultados y simulaciones.** *En esta imagen se muestra el "matching" inicial obtenido por SIFT. Cada par de correspondencias entre un punto del patrón y la imagen se representa con una recta de color verde .*

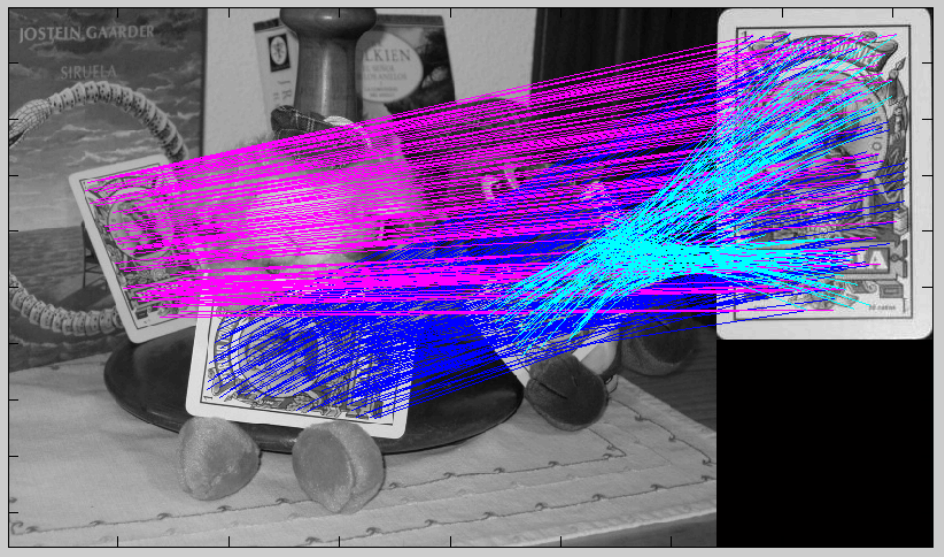


Figura 7.85: **Resultados y simulaciones.** *En esta imagen se muestran los "inliers" tras aplicar RANSAC junto con la aproximación afín. Cada color representa un nuevo objeto*

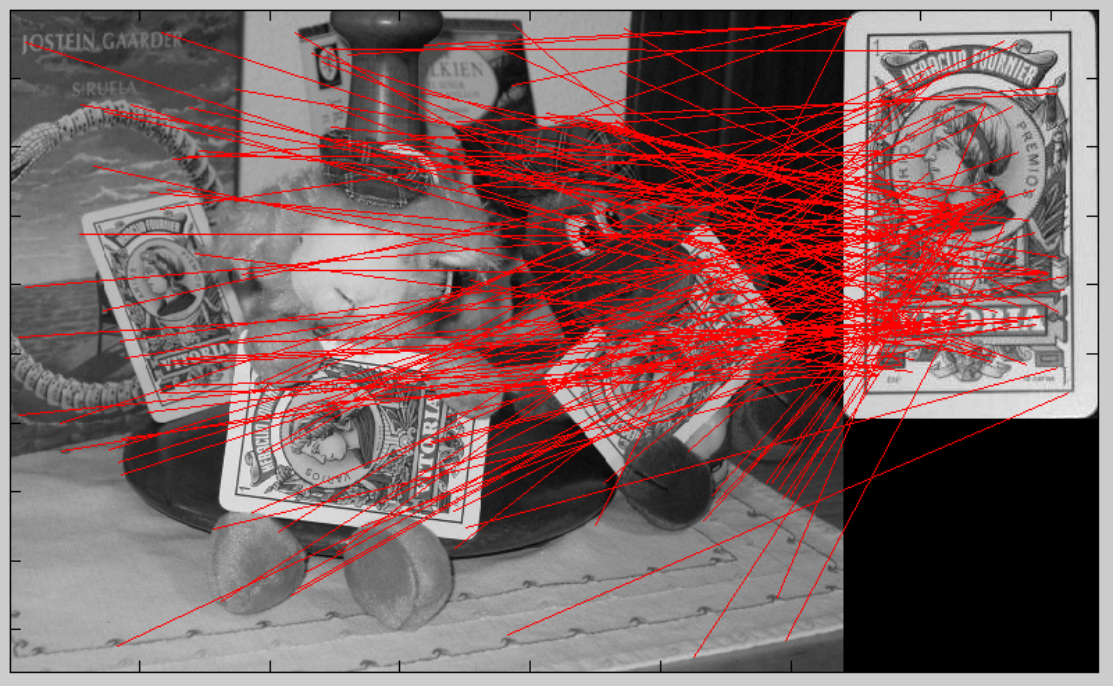


Figura 7.86: **Resultados y simulaciones.** *En esta imagen se muestran los "outliers" tras aplicar RANSAC junto con la aproximación afín.*

Figura 7.87: **Resultados y simulaciones.** *En esta imagen se muestra el perfil de reproyección de cada objeto detectado tras aplicar RANSAC junto con la aproximación afín.*

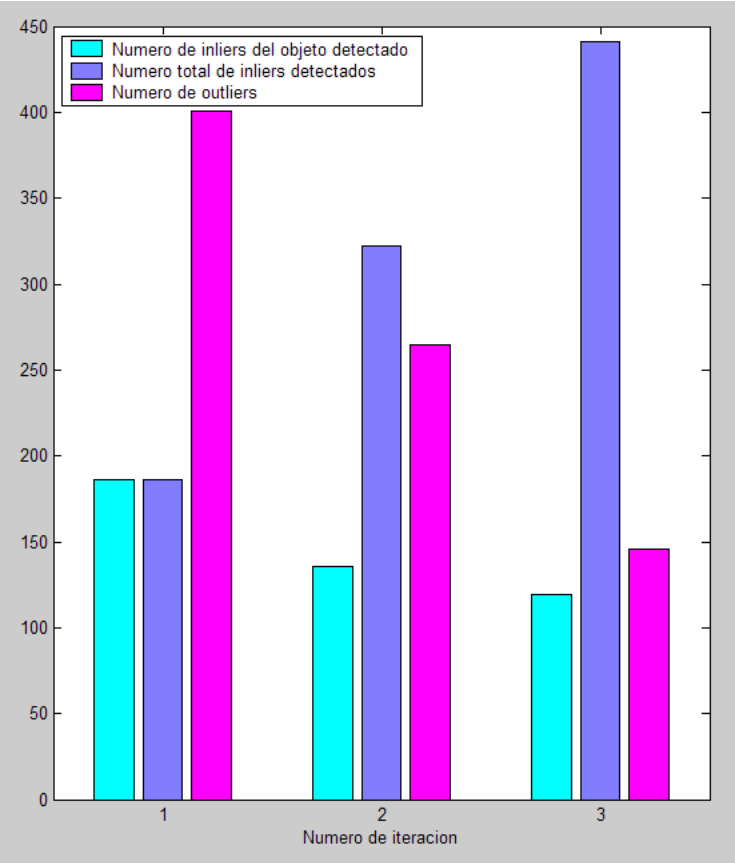


Figura 7.88: **Resultados y simulaciones.** *En esta imagen se muestran el número de "inliers" detectados en cada iteración válida de RANSAC, junto con el número de "outliers" y el número total de "inliers".*

- **Segundo ejemplo**

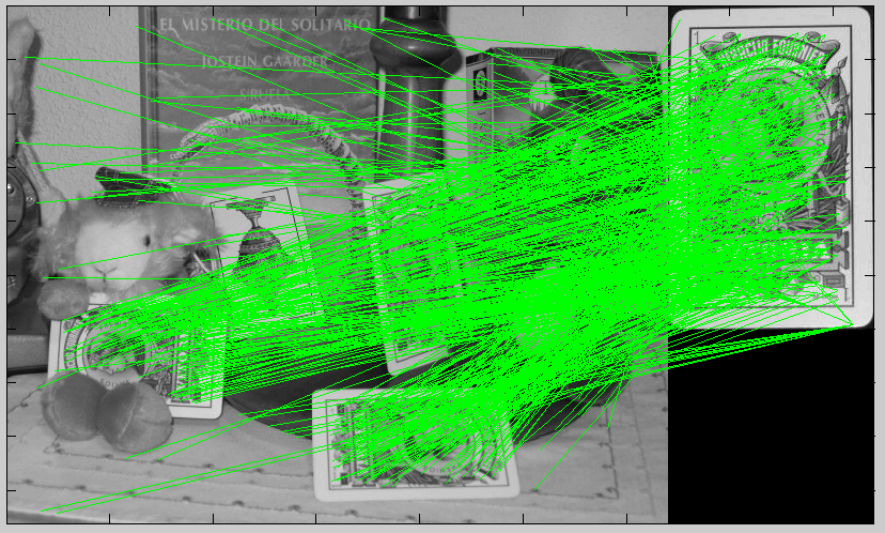


Figura 7.89: **Resultados y simulaciones.** *En esta imagen se muestra el "matching" inicial obtenido por SIFT. Cada par de correspondencias entre un punto del patrón y la imagen se representa con una recta de color verde.*

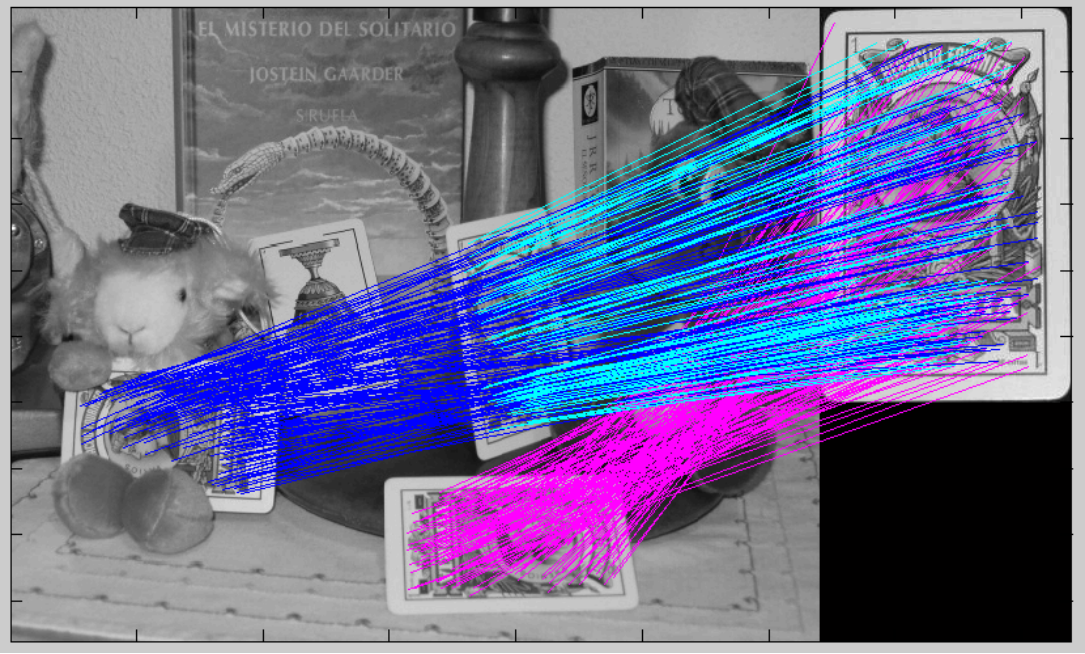


Figura 7.90: **Resultados y simulaciones.** *En esta imagen se muestran los "inliers" tras aplicar RANSAC junto con la aproximación afín. Cada color representa un nuevo objeto*

Figura 7.91: **Resultados y simulaciones.** *En esta imagen se muestran los "outliers" tras aplicar RANSAC junto con la aproximación afín.*

Figura 7.92: **Resultados y simulaciones.** *En esta imagen se muestra el perfil de reproyección de cada objeto detectado tras aplicar RANSAC junto con la aproximación afín.*

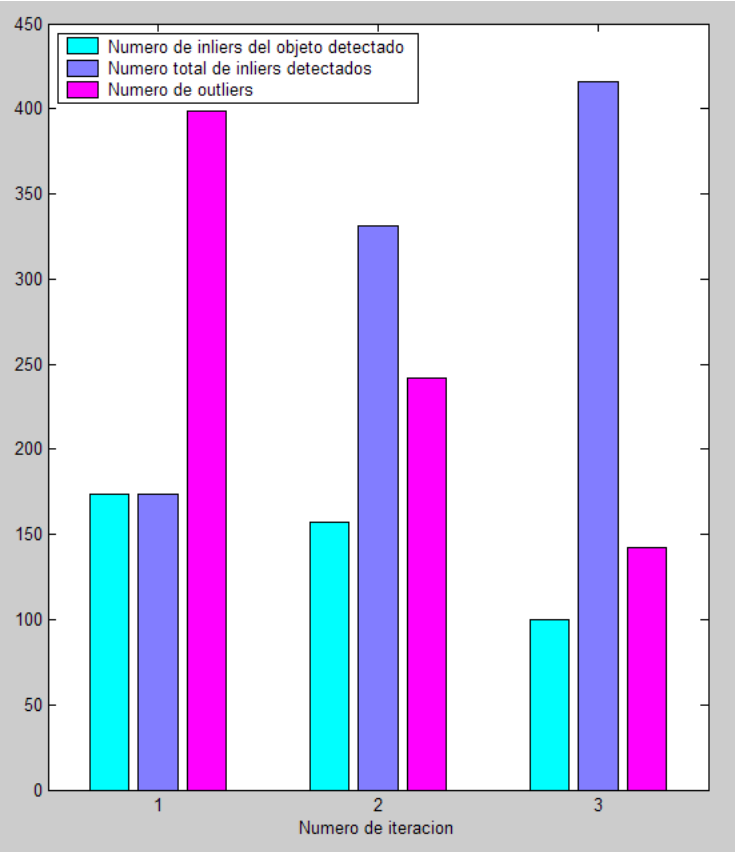


Figura 7.93: **Resultados y simulaciones.** *En esta imagen se muestran el número de "inliers" detectados en cada iteración válida de RANSAC, junto con el número de "outliers" y el número total de "inliers".*

Parte III

Pliego de condiciones

Pliego de Condiciones

Para el correcto funcionamiento de los distintos sistemas de detección desarrollados en este proyecto es necesario disponer del material y del software descrito a continuación:

a) Hardware

- Ordenador portátil Pentium M. 1.7 GHz con 1 GB de RAM o de mejores prestaciones.
- Al menos una cámara compatible con la norma IEEE1394 (pedir datos a dani)
- Estructura de soporte para las cámaras.
- Plataforma giratoría para generar el patrón tridimensional.
- Patrón de calibración para las cámaras.

b) Software

- Sistema operativo Windows XP o GNU/Linux Ubuntu 3.4 con kernel o superior.
- Matlab 7.0 o superior compatible con el sistema operativo utilizado.
- Aplicación que permita controlar cámaras digitales por el puerto IEEE-1394 y capturar imagenes. Por ejemplo, Coriander (junto con sus dependencias correspondientes).

Parte IV

Presupuesto

Capítulo 8

Presupuesto del Proyecto

En este apartado podemos ver el desglose de todos los costes relacionados con el desarrollo del proyecto. Podemos dividir los costes en los referentes a especificaciones y requerimientos técnicos, y en los referentes a horas de trabajo humano en el desarrollo del proyecto.

8.1. Costes de ejecución material

Los costes de ejecución incluye tres elementos: el coste de equipos, el coste del software y el coste de mano de obra por el tiempo empleado en el proyecto.

8.1.1. Costes de equipos

CONCEPTO	PRECIO UNITARIO	CANTIDAD	SUBTOTAL
Ordenador portátil Pentium M. 1,7 GHz 1GB RAM	650,00€	1	650,00€
Cámara Digital FireWire DFK 21BF04	350 €	1	350€
Trípode	24.99 €	1	24,99 €
Plataforma giratoria	560 €	1	560 €
Tablero de 274×300 mm de fibra de carbono	60 €	1	60 €

Subtotal: ..... **1.644,99 €**

8.1.2. Costes de software para el desarrollo del proyecto

CONCEPTO	PRECIO UNITARIO	CANTIDAD	SUBTOTAL
Sistema operativo Linux: Ubuntu 6.06	0 €	1	0 €
Windows XP Home Edition (incluido en el portatil)	0 €	1	0 €
Coriander 1.0.1	0 €	1	0 €
Matlab 7.0	500,00 €	1	500,00 €

Subtotal: ..... **500,00 €**

8.1.3. Costes de software para la elaboración de la documentación

CONCEPTO	PRECIO UNITARIO	CANTIDAD	SUBTOTAL
MiKTeX	0 €	1	0 €
TeXnicCenter	0 €	1	0 €
Gimp 2.2	0 €	1	0 €
SmartDraw 7	155,95 €	1	155,95 €

Subtotal: ..... **155,95 €**

8.1.4. Costes por tiempo empleado

FUNCIÓN	PRECIO UNITARIO	Nº HORAS	SUBTOTAL
Ingeniería	25,00 €	1080	27.000,00 €
Mecanografiado	15,00 €	240	3.600 €

Subtotal: ..... **30.600,00 €**

A la hora de realizar el cálculo de horas de desarrollo del proyecto, se ha estimado una dedicación de 6 horas durante 8 meses.

8.1.5. Costes total del presupuesto de ejecución material

CONCEPTO	SUBTOTAL
Coste de equipos	1.644,99 €
Coste de software para el desarrollo del proyecto	500,00 €
Coste de software para el desarrollo de la documentación	155,95 €
Coste por tiempo empleado	30.600,00 €

Subtotal: ..... **32.900,94 €**

8.2. Gastos generales y beneficio industrial

Los gastos generales y beneficio industrial son los gastos obligados que se derivan de la utilización de las instalaciones de trabajo más el beneficio industrial. Se estima un porcentaje del 16 % sobre el coste de ejecución material

CONCEPTO	SUBTOTAL
Gastos generales y beneficio industrial	5.264,15 €

Subtotal: ..... **5.264,15 €**

8.3. Importe total del presupuesto

CONCEPTO	SUBTOTAL
Costes total del presupuesto de ejecución material	32.900,94 €
Gastos generales y beneficio industrial	5.264,15 €

TOTAL: ..... **38.165,09 €**

IVA 16 %: ..... **6.106,41 €**

TOTAL IVA INCLUIDO: ..... **44.271,50 €**

El Importe Total del proyecto suma la cantidad de:

Cuarenta y Cuatro Mil Doscientos Setenta y Un Euros, con Cincuenta Céntimos

Alcalá de Henares a 16 de Diciembre de 2007

Fdo: Amaia Santiago Pé
Ingeniero de Telecomunicación

Bibliografía

[Ballard and Brown, 1982] Ballard, D. H. and Brown, C. M. (1982). *Computer Vision*. Prentice Hall.

[Basagoitia et al., 2004] Basagoitia, E., Pizarro, D., Novo, I., and Mazo, M. (2004). Multiple camera calibration using point correspondences oriented to intelligent spaces. In *TELEC'04 International Conference*.

[Beis and Lowe, 1997] Beis, J. S. and Lowe, D. G. (1997). Shape indexing using approximate nearest-neighbour search in high-dimensional spaces. In *CVPR*, pages 1000–1006.

[Coen, 1998] Coen, M. H. (1998). Design principles for intelligent environments. In *AAAI '98/IAAI '98: Proceedings of the fifteenth national/tenth conference on Artificial intelligence/Innovative applications of artificial intelligence*, pages 547–554, Menlo Park, CA, USA. American Association for Artificial Intelligence.

[Duda and Hart, 1971] Duda, R. O. and Hart, P. E. (1971). Use of the hough transformation to detect lines and curves in pictures. Technical report, AI Center, SRI International. SRI Project 8259 Comm. ACM, Vol 15, No. 1.

[Fischler and Bolles, 1981] Fischler, M. A. and Bolles, R. C. (1981). Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395.

[Friedman et al., 1977] Friedman, J. H., Bentley, J. L., and Finkel, R. A. (1977). An algorithm for finding best matches in logarithmic expected time. *ACM Trans. Math. Softw.*, 3(3):209–226.

[Hartley and Zisserman, 2004] Hartley, R. and Zisserman, A. (2004). *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition.

[Hashimoto and Lee, 2002] Hashimoto, H. and Lee, J.-H. (2002). Intelligent space – concept and contents. *Advanced Robotics*, 16(3):265–280.

[Hashimoto et al., 2003] Hashimoto, H., Lee, J.-H., and Ando, N. (2003). Self-identification of distributed intelligent networked device in intelligent space. In *ICRA*, pages 4172–4177.

[Hough, 1962] Hough, P. (1962). Method and means for recognizing complex patterns. In *US Patent*.

[Koenderink, 1984] Koenderink, J. J. (1984). The structure of images. *Biological Cybernetics*, V50(5):363–370.

[Lee et al., 2005] Lee, J.-H., Morioka, K., and Hashimoto, H. (2005). Intelligent space and mobile robots. In *The Industrial Information Technology Handbook*, pages 1–15.

[Lindeberg, 1994] Lindeberg, T. (1994). Scale-space theory: A basic tool for analysing structures at different scales. *J. of Applied Statistics*, 21(2):224–270.

[Lowe, 1999] Lowe, D. G. (1999). Object recognition from local scale-invariant features. In *ICCV*, pages 1150–1157.

[Lowe, 2004] Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. In *International Journal of Computer Vision*, volume 20, pages 91–110.

[Mikolajczyk, 2002] Mikolajczyk, K. (2002). *Detection of local features invariant to affines transformations*. PhD thesis, INPG, Grenoble.

[Pentland, 1996] Pentland, A. (1996). Smart rooms. *Scientific American*.

[Pizarro et al., 2005] Pizarro, D., Santiso, E., and Mazo, M. (2005). Simultaneous localization and structure reconstruction of mobile robots with external cameras. In *ISIE05*.

[Shafer et al.,] Shafer, S., Krumm, J., Brumitt, B., Meyers, B., Czerwinski, M., and Robbins, D. The new easyliving project at microsoft research. In *Proceedings, Joint DARPA/NIST Smart Spaces Workshop*.

[Villadangos et al., 2005] Villadangos, J. M., Ureña, J., Mazo, M., Hernandez, A., Alvarez, F., Garcia, J. J., Marziani, C. D., and Alonso., D. (2005). Improvement of ultrasonic beacon-based local position system using multi-access techniques. In *WISP05*, page CDROM Edition.

[Weiser, 1993a] Weiser, M. (1993a). Parc builds a world saturated with computation. *Science (AAAS)*, pages 3–11.

[Weiser, 1993b] Weiser, M. (1993b). Some computer science problems in ubiquitous computing. *Communications of the ACM*.

[Weiser, 1999] Weiser, M. (1999). The computer for the 21st century. *SIGMOBILE Mob. Comput. Commun. Rev.*, 3(3):3–11.

[Witkin, 1983] Witkin, A. P. (1983). Scale-space filtering. In *IJCAI*, pages 1019–1022.