Departamento de Informática y Sistemas

Procesado de Vídeo Digital

Digital Video Processing

Tesis Doctoral
Ph.D. Thesis

Autor: Pedro Henríquez Castellano

Director: Luis Álvarez León

Las Palmas de Gran Canaria
Marzo 2013

A mi familia

Agradecimientos

En primer lugar me gustaría hacer un agradecimiento muy especial a mis padres por todo el esfuerzo que han hecho para que pudiese conseguir este objetivo y por el gran apoyo que siempre he recibido por su parte.

Mi más sincero agradecimiento al director de la tesis Luis Álvarez por su dedicación, orientación, seguimiento y continua supervisión de este trabajo de investigación.

En todos los años que he estado trabajando en el CTIM he disfrutado siempre de un buen ambiente de trabajo. Quisiera agradecer a varios miembros del CTIM su inestimable ayuda en muchos momentos a lo largo de mi estancia en el centro: Luis Mazorra, Agustín Salgado, Javier Sánchez, Miguel Alemán, Luis Gómez, Agustín Trujillo y Karl Krissian. Además, he tenido la suerte de conocer a varios buenos compañeros que nunca dudaban en ayudar a los demás y debatir cualquier tema en la hora de la fruta. Los recuerdo aquí en varios grupos; AMI: Carlos, Daniel, Javier y Jesús; SIG: Adaya, Airam, Kilian y Nelson; Capaware: Antonio, David y Pedro; ComC: José.

Para finalizar, dar las gracias también a los familiares y amigos que siempre están ahí, en especial a Judit por su apoyo, comprensión y ayuda con las presentaciones.

Índice general

Introducción

En los últimos años se ha visto incrementado el uso del vídeo digital de alta definición, como por ejemplo en la televisión ($HDTV$) o en el cine. Este formato de vídeo aporta nuevas características, como son el número de líneas de barrido, la claridad de imagen, mayor profundidad de campo y nivel de detalle. Esto hay que sumarlo a que las cámaras capaces de grabar en alta definición, ofrecen mayor control técnico.

Este tipo de producción, además de ser interesante para el usuario, también lo es desde el punto de vista científico-tecnológico, dado que, por ejemplo para la producción de varios eventos es posible utilizar una gran variedad de tipos de cámaras, puntos de vista, diferentes montajes y combinaciones de las tomas de todas las cámaras siguiendo la acción, pudiendo retransmitirlo todo en alta calidad consiguiendo que el espectador se sienta inmerso en lo que está viendo. Por otra parte, hay eventos en los que los gráficos por computador juegan un papel importante, ya que habitualmente se incluyen gráficos para ofrecer información extra al espectador. Trabajar con vídeo digital facilita la inserción de gráficos virtuales y efectos especiales.

En esta tesis se estudian varios procedimientos para procesar el vídeo digital con el fin de llevar a cabo aplicaciones como las comentadas anteriormente. Uno de los procesos importantes en este contexto es la calibración de cámaras, que permite obtener los parámetros y posición de la cámara con la que se ha grabado el vídeo. Esta valiosa información permite el desarrollo de aplicaciones para mejorar la compresión de la escena, como añadir gráficos, mediciones, cambiar el punto de vista, seguimiento de objetos de interés, etc.

La calibración de cámaras, que puede llegar a ser un proceso costoso, se divide en varias etapas. Dichas fases implican la ejecución de otros procedimientos también importantes dentro del campo de la visión por computador, como son la segmentación, detección de primitivas, seguimiento de objetos y corrección de la distorsión de lentes.

El trabajo realizado está compuesto por contribuciones a la literatura en cada una

de las fases necesarias para el procesado de vídeo digital con calibración de cámaras. Como aplicación de todos los métodos propuestos se estudia también la inserción de gráficos virtuales en secuencias de vídeo.

En primer lugar, se trabaja con la segmentación de imágenes y detección de primitivas en la escena, procesos necesarios para calibrar la cámara. Se estudia cómo localizar con precisión las primitivas en imágenes de alta definición y en escenarios reales. Este tipo de imágenes plantea ciertas dificultades como el cambio de iluminación en distintas zonas de la imagen o el número de píxeles de grosor que puede llegar a tener una primitiva.

Cuando se cuenta con un número suficiente de primitivas detectadas en una imagen se pueden recuperar los parámetros de la cámara. En este trabajo se propone un método de calibración de cámaras aisladas a partir de las primitivas. Por otra parte, cuando se trata de procesar un vídeo, hay que tener en cuenta otras características y requerimientos. Por ello, se realizó el estudio de la calibración de cámaras de vídeo montadas en un trípode. El método propuesto aprovecha las características geométricas del trípode para mejorar la obtención de los parámetros de la cámara.

Una cuestión importante cuando se procesa vídeo digital es el tiempo de procesado, ya que muchas veces se necesita en retransmisiones en directo, repeticiones instantáneas o en resúmenes que se publican poco tiempo después de la emisión. Por lo tanto, se ha estudiado cómo mejorar este tiempo de procesado sin pérdida en la precisión, y se han propuesto nuevos métodos de seguimiento de primitivas y calibración en tiempo real.

Otro de los aspectos que influyen en la precisión de los resultados de la calibración es la distorsión de la lente. Ésta puede provocar errores en la estimación de los parámetros de la cámara y por consiguiente errores en las aplicaciones posteriores. Se ha realizado un estudio del efecto de dicha distorsión de lentes en las secuencias de vídeo, del cual se han obtenido nuevos modelos matemáticos para su estimación y corrección.

Después de realizada la calibración de la cámara, se dispone de información suficiente para insertar gráficos virtuales en secuencias de vídeo. Se ha abordado este tema para probar la calidad de los métodos propuestos en este documento de tesis, debido a que esta aplicación requiere rapidez en el procesado del vídeo a la vez que precisión en los resultados de calibración.

Al añadir gráficos a una secuencia, el ojo humano puede percibir pequeñas vibraciones durante el vídeo debidas a pequeños errores de precisión en la calibración de la cámara. Para corregir este tipo de perturbaciones, se propone añadir un método de suavizado al proceso de seguimiento de la cámara.

El trabajo desarrollado en esta tesis doctoral forma parte de las líneas de investigación del grupo de Análisis Matemático de Imágenes dirigido por el profesor Luis Álvarez León e inscrito en el Centro de I+D de Tecnologías de la Imagen. Las actividades desarrolladas han sido financiadas por:

- El Cabildo de Gran Canaria a través del programa de becas predoctorales "Becas de investigación en temas de interés para la isla de Gran Canaria".
- La Universidad de Las Palmas de Gran Canaria a través del programa de ayudas "Plan de formación del personal investigador".
- El proyecto de investigación "I3MEDIA: Tecnologías para la creación y gestión automatizada de contenidos audiovisuales", financiado por CENIT Ministerio de Industria, Turismo y Comercio, Mediaproducción S.L., cuyo responsable por la ULPGC es D. Luis Alvarez León.
- El proyecto de investigación "Modelización matemática de los procesos de calibración de cámaras de vídeo", financiado por el Ministerio de Ciencia e Innovación, subprograma de proyectos de investigación fundamental no orientada, convocatoria 2010. (Ref: MTM2010-17615), dirigido por D. Luis Alvarez León.

Contenido de la tesis

El documento está dividido en ocho capítulos. En el primero de ellos se hace un recorrido por el estado del arte de los problemas que se estudian en este trabajo. Después, se dedica un capítulo completo a cada uno de los problemas tratados, en los cuales se describen los métodos y aportaciones realizadas. El último capítulo está destinado a comentar las principales conclusiones obtenidas de los diferentes trabajos desarrollados durante la realización de la tesis. A continuación, se resume el contenido de cada capítulo:

- Capítulo 1, *Estado del arte:*
  El capítulo contiene la descripción del estado del arte relativo a varios temas importantes en el procesado de vídeo digital. Empezando por el problema de la calibración de cámaras, un proceso para el cual es necesario realizar varias tareas previamente, como son la detección de bordes, segmentación de la imagen y extracción de primitivas. Otro problema que se aborda, y que también afecta a la calibración de cámaras, es la corrección de la distorsión de lentes. Finalmente se describe la inserción de gráficos virtuales en secuencias de vídeo. En todas las secciones se hace mención de los métodos más relevantes de la literatura, haciendo hincapié en algunas de las técnicas más innovadoras propuestas en los últimos años y que han servido de referencia a otros métodos posteriores.

- Capítulo 2, *Localización de primitivas en la escena:*
  En este capítulo se trata el problema de la localización de primitivas en una imagen. Se estudia detectar líneas y círculos en escenarios reales, donde aparecen dificultades como los cambios de iluminación o la gran cantidad de píxeles que proporcionan los vídeos de alta definición.

- Capítulo 3, *Calibración de cámaras aisladas a partir de primitivas:*
  El trabajo realizado en este capítulo se centra en el desarrollo de un procedimiento para calibrar cámaras a partir de la información que se obtiene de una imagen aislada. Para ello se utilizan las primitivas localizadas en la imagen, que proporcionan la información necesaria para llevar a cabo la estimación de la posición de la cámara.

- Capítulo 4, *Calibración de cámaras montadas en un trípode:*
  Aquí se estudian los cambios que se realizan en los procesos de calibración cuando hay que procesar una secuencia de vídeo proveniente de una cámara montada sobre un trípode. Se propone un nuevo modelo que simplifica la calibración de secuencias de vídeo, aprovechando las características del trípode.

- Capítulo 5, *Seguimiento de primitivas en una secuencia de vídeo:*
  Se propone un nuevo método para el seguimiento de primitivas durante una secuencia de vídeo. Este proceso es necesario para realizar la calibración de cámaras en una secuencia utilizando solamente información obtenida de la misma.

- Capítulo 6, *Variación del modelo de distorsión de la lente en una secuencia de vídeo:*
  En este capítulo se estudia el efecto que tienen los cambios en la distancia focal sobre el modelo de distorsión de la lente. Se proponen modelos matemáticos para estimar el modelo de distorsión de la lente cuando éste se ve modificado a causa de las variaciones en la distancia de enfoque o del *zoom*.

- Capítulo 7, *Suavizado del movimiento de una cámara en una secuencia de vídeo:*
  Al calibrar una secuencia de forma individual fotograma a fotograma de forma independiente, pueden producirse ciertas discontinuidades en los resultados. En este trabajo se propone un método para suavizar estos resultados y eliminar posibles vibraciones que puedan aparecer cuando se utilizan los resultados de la calibración para otras aplicaciones, como por ejemplo al insertar gráficos en la secuencia.

- Capítulo 8, *Inserción de gráficos en escenas reales de escenarios deportivos:*
  Este capítulo está dedicado a describir la técnica propuesta con la que se añade contenido virtual a una secuencia de vídeo usando la calibración de cámaras a partir de primitivas, además se explica el proceso de sincronización de cámaras y el procedimiento de renderizado.

- Capítulo 9, *Conclusiones y trabajo futuro:*
  Se exponen las conclusiones finales de la tesis comentando los problemas encontrados durante su desarrollo y destacando las aportaciones realizadas a la literatura en el trabajo recogido en este documento. También se comentan las posibilidades de trabajo futuro en esta misma línea.

Las principales aportaciones

- *Localización de primitivas en la imagen:*
  Se propone una técnica para extraer correctamente de una imagen líneas y sus centros usando operadores morfológicos. El método funciona incluso con imágenes entrelazadas y cambios de iluminación dentro de las mismas. (Publicación 1). Los resultados de esta aportación se ilustran en la página web: http://www.ctim.es/demo107.

- *Calibración de cámaras a partir de primitivas de la imagen:*
  El método propuesto calibra automáticamente una cámara a partir de una sola imagen de un escenario donde los objetos de interés están en un plano. Esta técnica usa como información las líneas y círculos que dividen una cancha deportiva. Después se minimiza una función de error para obtener la mejor transformación de perspectiva, y finalmente se recupera la posición de la cámara. (Publicaciones 6 y 9).

- *Calibración de cámaras montadas en un trípode:*
  Proponemos un método novedoso para la calibración de cámaras en escenarios planos, que se ejecuta en tiempo real. Este método se basa en la geometría de un trípode, una estimación inicial de la posición de la cámara y en un procedimiento de seguimiento de primitivas. (Publicaciones 5 y 8). Los resultados de esta contribución se ilustran en http://www.ctim.es/demo106 y en http://www.ctim.es/demo108.

- *Seguimiento de primitivas en un vídeo:*
  Se ha propuesto un nuevo método para el seguimiento de primitivas basado en un *CART(Classification and Regression Tree)*. El procedimiento usa una estimación de la homografía y líneas y círculos como primitivas. (Publicación 2). Los resultados de esta aportación se ilustran en la página web: http://www.ctim.es/demo105.

- *Variación del modelo de distorsión en una secuencia de vídeo:*
  Se propone un nuevo modelo matemático que estudia la variación del modelo de distorsión cuando se cambia el *zoom* de la cámara. El nuevo modelo está basado en una aproximación polinómica y una minimización de una función de error.(Publicación 4). En la página web http://www.ctim.es/demo101 se ilustran los resultados de esta contribución.

- *Suavizado del movimiento de una cámara en una secuencia de vídeo:*
  Se introduce a la calibración de cámaras de vídeo una restricción de suavizado. La función de calibración se obtiene como el mínimo de una función de energía.

Se estudia la existencia de un mínimo de dicha función, al igual que las soluciones de las ecuaciones de Euler-Lagrange asociadas. (Publicación 3).En la página web http://www.ctim.es/demo102 se ilustran los resultados de esta contribución.

- *Inserción de gráficos en imágenes de escenarios deportivos:* El método propuesto usa solamente información de la imagen para llevar a cabo la calibración de la cámara. Después, se realiza una sincronización entre la cámara real y virtual, que hace posible el renderizado final de los gráficos en la secuencia de vídeo. (Publicación 7). Los resultados de la aportación se ilustran en la dirección: http://www.ctim.es/demo104.

Publicaciones realizadas

En esta sección se hará una breve descripción de las publicaciones que se han realizado en el contexto de esta tesis.

1. M. Alemán-Flores, L. Alvarez, P. Henriquez, L. Mazorra: **Morphological thick line center detection.** In: 7th International Conference on Image Analysis and Recognition, Lecture Notes in Computer Science V. 6111 (2010) 71-80.

   Se ha propuesto un método morfológico para detectar los centros de líneas en las imágenes, teniendo en cuenta que pueden estar en partes de la imagen con distinta iluminación y contraste. También es capaz de detectar los centros de líneas de distinto grosor. El método aporta mejor precisión en la detección de centros de líneas presentes en las imágenes de alta definición de escenarios reales, como son los vídeos de partidos de fútbol.

2. L. Alvarez, P. Henriquez, J. Sánchez: **CART application to image primitive tracking.** CAEPIA 11: Conferencia de la Asociación Española para la Inteligencia Artificial, (2011).

   Se propone un nuevo método para el seguimiento de primitivas en una secuencia de vídeo, el cual está basado en un *CART (Classification and Regression Tree)*. El procedimiento usa líneas y círculos como primitivas. Se estiman los parámetros del *CART* usando un proceso de aprendizaje basado en los canales *RGB* de la imagen. La calidad del seguimiento de las primitivas con el árbol de decisión se valida por medio de los porcentajes de error obtenidos al clasificar imágenes y la comparación con otras técnicas. Se presenta también cómo puede incluirse este método en el proceso de calibración de cámaras y cómo acelera la ejecución del mismo.

3. L. Alvarez, L. Gomez, P. Henriquez, L. Mazorra: **A variational approach to camera motion smoothing.** In: Differential Equations and Applications - DEA, 4 (2011) 555-564.

   Se estudia el problema variacional que se deriva de la calibración de vídeo con restricción de suavizado. Por calibración de cámara se entiende, estimar la localización, orientación y *zoom* de la cámara para cada fotograma del vídeo. En este caso, se trabaja con cámaras montadas en un trípode, y para cada fotograma capturado en el instante t, la calibración queda definida por 3 parámetros: $P(t)$ *(PAN)* y $T(t)$ *(TILT)* que representan la rotación del trípode sobre el eje vertical y horizontal respectivamente, y $Z(t)$ *(CAMERA ZOOM)* el *zoom* de la lente de la cámara. La función de calibración $t \to \mathbf{u}(t) = (P(t), T(t), Z(t))$ se obtiene como

el mínimo de una función de energía $I[\mathbf{u}]$. Por ello, en esta aportación se estudia la existencia de un mínimo de dicha función de energía a la vez que las soluciones de las ecuaciones de Euler-Lagrange asociadas.

4. L. Alvarez, L. Gomez, P. Henriquez: **Zoom dependent lens distortion mathematical models.** Journal of Mathematical Imaging and Vision, 44(3) (2012) 480-490.

   Se proponen nuevos modelos matemáticos para estudiar la variación de la distorsión cuando se modifica el *zoom* de la lente. Los nuevos modelos están basados en una aproximación polinómica para tener en cuenta la variación de los parámetros de distorsión radial a lo largo del rango de *zoom* de la lente y la minimización de un error global de la energía al medir la distancia entre secuencias de alineaciones de puntos distorsionados y líneas rectas después de corregir la distorsión. Para validar el rendimiento del método, se realizan experimentos con imágenes de un patrón de calibración y con vídeos de eventos deportivos.

5. L. Alvarez, P. Henriquez, L. Mazorra: **Mathematical models for the calibration of cameras mounted on a tripod using primitive tracking.** In: 9th International Conference on Image Analysis and Recognition, Lecture Notes in Computer Science V. 7324 (2012) 304-311.

   Proponemos un nuevo modelo matemático para la calibración de secuencias de vídeo cuando la cámara está montada en un trípode. Una de las novedades es que no se supone que el centro de rotación del trípode y el centro de proyección de la cámara sean el mismo punto. La calibración está basada en la geometría del trípode y en el seguimiento de primitivas. La calidad del proceso de calibración se ha validado insertando elementos virtuales en la secuencia de vídeo.

6. L. Alvarez, L. Gomez, P. Henriquez, L. Mazorra: **Automatic camera pose recognition in planar view scenarios.** 17th Iberoamerican Congress on Pattern Recognition, Lecture Notes in Computer Science V. 7441 (2012) 406-413.

   Se ha propuesto un método cuyo objetivo principal es reconocer automáticamente la posición de la cámara a partir de una sola imagen de un escenario plano. Se ha aplicado esta técnica a escenarios de eventos deportivos usando como información la líneas y círculos que dividen las diferentes partes del terreno de juego. Con estas primitivas de la cancha, se define una función de error que se minimiza para obtener la mejor transformación de perspectiva (homografía), haciendo coincidir una cancha real con su proyección en la imagen. De dicha homografía se recupera la posición y orientación de la cámara en el espacio 3D.

7. M. Alemán-Flores, L. Alvarez, P. Henriquez, A. Trujillo: **Augmented reality in sport scenarios using cameras mounted on a tripod.** Technical Report - Centro de Tecnologías de la Imagen, 2012.

Se aborda el problema de insertar contenido virtual en una secuencia de vídeo. El método que se propone usa solamente información de la imagen. En él se realiza un seguimiento de primitivas, calibración de cámaras, sincronización de las cámaras real y virtual, y por último el renderizado para añadir los gráficos virtuales al vídeo real. La sincronización de la cámara real y la virtual, y el renderizado se llevan a cabo usando funciones de OpenGL *(Open Graphic Library)*. Para ilustrar el rendimiento y la calidad del método propuesto, se ha validado insertando elementos virtuales en vídeos de alta definición.

8. L. Alvarez, L. Gomez, P. Henriquez, J. Sánchez: **Real-time camera motion tracking in planar view scenarios.** Enviado a Journal of Real-Time Image Processing, 2013.

   Se propone un nuevo método para el seguimiento en tiempo real del movimiento de una cámara en escenarios planos. El método se basa en la geometría del trípode, una estimación inicial de la posición de la cámara realizada en el primer fotograma y un procedimiento de seguimiento de primitivas. Este proceso usa líneas y círculos como primitivas, las cuales son extraídas aplicando un CART (árbol de clasificación y regresión). El método propuesto se aplicó a vídeos de partidos de fútbol grabados en alta definición. Los resultados de los experimentos prueban que la propuesta puede procesar vídeos de alta definición en tiempo real.

9. M. Alemán-Flores, L. Alvarez, L. Gomez, P. Henriquez: **Camera Calibration in Sport Event Scenarios.** Enviado a Pattern Recognition, 2013.

   El objetivo principal de este artículo es calibrar la cámara en escenarios deportivos, tales como campos de fútbol, baloncesto o canchas de tenis usando solamente una imagen. En estos casos, las referencias principales que se pueden usar para calibrar la cámara son las líneas y los círculos que delimitan las regiones del terreno de juego. El primer problema que se aborda es la extracción de las primitivas de la imagen, teniendo en cuenta también el caso de zonas sombreadas. A partir de estas primitivas, se localiza automáticamente la cancha deportiva en la escena, estimando la homografía que hace coincidir la cancha real con su proyección en la imagen. Por último, de esta homografía se recuperan los parámetros de calibración de la cámara (distancia focal, posición y orientación en el espacio 3D). Se presentan varios experimentos que ilustran la precisión de la calibración obtenida.

Abstract

In the last years, the use of high definition video has been increased, thanks to the television (HDTV) or the cinema. This video format provides new features, such as the number of scan lines, the image clearness, higher depth field and detail level. Moreover, the modern cameras which are capable of recording HD video, offer a better technical control.

This kind of production, apart from being interesting for the common user, is also interesting from the scientist and technological point of view. For example, in the production of some events, a great variety of cameras can be used, different points of view can be selected, different camera editing and combinations of shots can be done to follow the action, and finally it is broadcasted in high definition involving the viewer in what is being watched. On the other hand, there are events where the computer graphics are important, because they are usually inserted to offer extra information to the spectator. Working with digital video makes easier the special effects and virtual graphics insertion.

In this dissertation we study some procedures to process the digital video with the aim of performing the previously commented applications. One of the important processes in this context, is the camera calibration. It allows to obtain the camera parameters and its position from an image or a video sequence. This useful information makes possible the development of many applications to improve the scene comprehension, such as graphic insertion, measurements, changing the point of view, object tracking, etc...

Camera calibration is divided in several tasks. These procedures are also important issues inside the computer vision field. The processes are segmentation, primitive extraction, tracking and lens distortion correction.

This work is a collection of contributions to the literature in each needed stage for the digital video processing with camera calibration. As an application of the proposed methods, we also studied the virtual graphic insertion in video sequences.

First of all, we work with image segmentation and image primitive extraction. It is studied how to locate the primitives in high definition images of real scenarios. This sort of images present certain difficulties as illumination changes in different image regions, or the primitive thickness (more pixels than in low definition images).

When we have detected enough image primitives, the camera parameters can be recovered. In this work, we propose a camera calibration method using the primitives from only one image. Moreover, when we have to process a video, there are some different features and requirements. Therefore, we also study the calibration of cameras mounted on a tripod, and we propose a method which is based on the geometry of the tripod to improve the camera parameters calculation through a video sequence.

On the other hand, when a digital video is processed, one important point is the processing time. We have studied how to improve this time without losing accuracy, and we have proposed new methods to perform primitive tracking and calibration in real-time.

One issue which influences the accuracy of the calibration results, is the lens distortion. It can be the cause of some errors in the calibration and therefore, errors in the applications which use calibration information as input. For this reason, we have studied the effect of the lens distortion in video sequences. We have obtained new mathematical models to estimate and correct the lens distortion taking into account the focal length variations.

With the information provided by the video sequence calibration results, we have studied how to insert virtual graphics into the video sequence. In this way, we can test the quality of the methods we have proposed in this dissertation, because this kind of application requires an accurate and fast camera calibration computation.

When inserting virtual content into the sequence, the human eye can notice small vibrations during the sequence due to small calibration errors. To correct this type of perturbations, we propose adding a smoothing method in the camera calibration process.

The work developed in this Ph.D. thesis is a part of the research lines of *"Análisis Matemático de Imágenes"* research group, lead by Prof. Luis Álvarez León and registered in *"Centro de I+D de Tecnologías de la Imagen"*. The developed activities have been funded by:

- Cabildo de Gran Canaria by the research and teaching grant programme "Becas de investigación en temas de interés para la isla de Gran Canaria".

- Universidad de Las Palmas de Gran Canaria by the funding programme "Plan de formación del personal investigador".

- Research project "I3MEDIA: Tecnologías para la creación y gestión automatizada de contenidos audiovisuales", funded by CENIT Ministerio de Industria, Turismo y Comercio (Spanish Government), Mediaproducción S.L., lead by D. Luis Alvarez León in the ULPGC part.

- Research project "Modelización matemática de los procesos de calibración de cámaras de vídeo", funded by Ministerio de Ciencia e Innovación (Spanish Government), non oriented fundamental research projects subprogramme 2010. (Ref: MTM2010-17615), lead by D. Luis Alvarez León.

Thesis organisation

The work is organised in eight chapters. The first chapter is the state of art review of the studied problems. The chapters from 2 to 8 contain the different problems studied in this work and the contributions we have proposed for them. The last chapter shows the main conclusions obtained. Straightaway, we show a small summary of each chapter:

- Chapter 1, *State of art:*
  This chapter contains the state of art description corresponding to some important digital video processing issues. Starting with the camera calibration problem, a process which is divided several stages. These stages are tasks like edge detection, segmentation, primitive extraction and lens distortion correction. It is also described the virtual graphics insertion into video sequences. In every section of the chapter, we mention the most important methods from the bibliography.

- Chapter 2, *Image primitive location:*
  In this chapter we deal with the problem of image primitive localisation. We study the detection of lines and circles in real scenarios, where we can find some difficulties such as illumination changes or the big amount of pixels provided by a high definition image.

- Chapter 3, *Camera calibration from image primitives:*
  The problem studied in this chapter is how to perform camera calibration with the information obtained from only one image. We use the primitives located in the image, which provide the required information to estimate the camera pose.

- Chapter 4, *Calibration of cameras mounted on a tripod:*
  We study the changes in the calibration methods when we have to deal with cameras mounted on a tripod. We propose a new mathematical model which simplifies the video camera calibration using the tripod features.

- Chapter 5, *Primitive tracking in a video sequence:*
  A new method for primitive tracking through a video sequence is described in this chapter. This procedure is needed to perform the camera calibration using only the information obtained from the sequence.

- Chapter 6, *Lens distortion model variation in a video sequence:*
  The lens distortion model may vary through a video sequence if the zoom parameter is modified. In this chapter we study this problem and we present mathematical models to estimate the lens distortion model when the model varies owed to the zoom changes.

- Chapter 7, *Camera motion smoothing in a video sequence:*
  When calibrating a video sequence with a frame by frame method, it can appear some discontinuities in the results. In this work we explain a method to smooth the results and removing the vibrations which can appear when virtual graphics are inserted into the sequence.

- Chapter 8, *Graphic insertion into scenes of real sport scenarios:*
  This chapter is focused in describing the technique to insert virtual content in a video sequence using the camera calibration results. We explain the camera synchronisation process and the render procedure.

- Chapter 9, *Conclusions and future work:*
  The final conclusions are presented in this chapter as well as the future work.

Main contributions

- *Image primitive location:*
  We propose a technique to properly extract the image thick lines and their centres using mathematical morphological operators. The method works even with interlaced images and illumination changes. (Publication 1). The results of this contribution are shown in the web: http://www.ctim.es/demo107.

- *Camera calibration from image primitives:*
  The proposed method calibrates automatically the camera from a single image

of a planar view scenario. This technique uses as information the white lines and circles dividing the different parts of a sport court. Then we minimise a loss function to obtain the best perspective transformation, and finally we recover the camera pose. (Publications 6 and 9).

- *Calibration of cameras mounted on a tripod:*
  We present a novel method for real-time camera calibration in planar view scenarios. This method relies on the geometry of a tripod, an initial estimation of camera pose for the first video frame and a primitive tracking procedure. (Publications 5 and 8). The results of this contribution are shown in http://www.ctim.es/demo106 and in http://www.ctim.es/demo108.

- *Primitive tracking in a video sequence:*
  This is a new method for image primitive tracking based on a CART (Classification and Regression Tree). Primitive tracking procedure uses a homography estimation and lines and circles as primitives. (Publication 2). The results are shown in the webpage: http://www.ctim.es/demo105.

- *Lens distortion model variation in a video sequence:*
  We propose new mathematical models to study the variation of lens distortion models when the zoom setting is changed. The new models are based on a polynomial approximation and the minimisation of a global error energy. (Publication 4). The results are shown in the webpage: http://www.ctim.es/demo101.

- *Camera motion smoothing in a video sequence:*
  We perform a video camera calibration with smoothing constraint. The calibration function is obtained as the minima of an energy function. We study the existence of minima of such energy function as well as the solutions of the associated Euler-Lagrange equations. (Publication 3). In the web http://www.ctim.es/demo102 the results of this contribution are shown.

- *Graphic insertion into scenes of real sport scenarios:*
  The method we propose uses just image information, to perform the primitive tracking and the camera calibration. Then, there is a real and virtual camera synchronisation and finally the virtual content rendering in the real video sequence. (Publication 7). The results are shown online in: http://www.ctim.es/demo104.

Publications

This section contains a small description of the publications done in the context of this thesis.

1. M. Alemán-Flores, L. Alvarez, P. Henriquez, L. Mazorra: **Morphological thick line center detection.** In: 7th International Conference on Image Analysis and Recognition, Lecture Notes in Computer Science V. 6111 (2010) 71-80.

   In this paper, we analyse this issue in real situations where we have to deal with some additional difficulties, such as the thick line distortion produced by interlaced broadcast video cameras or large shaded areas in the scene. We propose a technique to properly extract the thick lines and their centres using mathematical morphological operators. In order to illustrate the performance of the method, we present some experiments in real images.

2. L. Alvarez, P. Henriquez, J. Sánchez: **CART application to image primitive tracking.** CAEPIA 11: Conferencia de la Asociación Española para la Inteligencia Artificial, (2011).

   We present a new method for image primitive tracking based on a CART (Classification and Regression Tree). Primitive tracking procedure uses lines and circles as primitives. We have applied the proposed method to sport event scenarios, specifically, soccer matches. We estimate CART parameters using a learning procedure based on RGB image channels. In order to illustrate its performance, it has been applied to real HD (High Definition) video sequences and some experiments are shown. The quality of the primitives tracking with the decision tree is validated by the percentage error rates obtained and the comparison with other techniques as a morphological method. We also present applications of the proposed method to camera calibration and graphic object insertion in real video sequences.

3. L. Alvarez, L. Gomez, P. Henriquez, L. Mazorra: **A variational approach to camera motion smoothing.** In: Differential Equations and Applications - DEA, 4 (2011) 555-564.

   We study a variational problem derived from a computer vision application: video camera calibration with smoothing constraint. By video camera calibration we mean to estimate the location, orientation and lens zoom-setting of the camera for each video frame taking into account image visible features. To simplify the problem we assume that the camera is mounted on a tripod, in such case, for each frame captured at time t, the calibration is provided by 3 parameters : $P(t)$ (PAN) and $T(t)$ (TILT) which represent the tripod vertical and horizontal axis rotation respectively, and $Z(t)$ (CAMERA ZOOM) the camera lens zoom setting. The calibration function $t \to \mathbf{u}(t) = (P(t), T(t), Z(t))$ is obtained as the minima of an energy function $I[\mathbf{u}]$. In this paper we study the existence of minima of such energy function as well as the solutions of the associated Euler-Lagrange equations.

4. L. Alvarez, L. Gomez, P. Henriquez: **Zoom dependent lens distortion mathematical models.** Journal of Mathematical Imaging and Vision, 44(3) (2012) 480-490.

   We propose new mathematical models to study the variation of lens distortion models when the zoom setting is changed. The new models are based on a polynomial approximation to account for the variation of the radial distortion parameters through the range of zoom lens settings and the minimisation of a global error energy measuring the distance between sequences of distorted aligned points and straight lines after lens distortion correction. To validate the performance of the method we present experimental results on calibration pattern images and on sport event scenarios using broadcast video cameras. We obtain, experimentally, that using just a second order polynomial approximation of lens distortion parameter zoom variation, the quality of lens distortion correction is as good as the one obtained individually frame by frame using independent lens distortion model for each frame.

5. L. Alvarez, P. Henriquez, L. Mazorra: **Mathematical models for the calibration of cameras mounted on a tripod using primitive tracking.** In: 9th International Conference on Image Analysis and Recognition, Lecture Notes in Computer Science V. 7324 (2012) 304-311.

   In this paper we present new mathematical models for video sequence calibration when cameras are mounted on a tripod. One of the main novelties is that tripod rotation centre and camera projection centre are not supposed to be the same. The calibration is based on the geometry of the tripod and a primitive tracking procedure which uses lines and circles as primitives. For the extraction of primitive information, we use a CART (Classification and Regression Tree). We have applied the method proposed to sport event scenarios, specifically, soccer matches. In order to illustrate its performance, it has been applied to real HD (High Definition) video sequences and some experiments are shown. The quality of the camera calibration procedure is validated by inserting virtual elements in the video sequence.

6. L. Alvarez, L. Gomez, P. Henriquez, L. Mazorra: **Automatic camera pose recognition in planar view scenarios.** 17th Iberoamerican Congress on Pattern Recognition, Lecture Notes in Computer Science V. 7441 (2012) 406-413.

   The proposed method recognises automatically the camera pose from a single image of a planar view scenario. We apply this technique to sport event scenarios using as information the white lines and circles dividing the different parts of the sport court. Using these court primitives we define a loss function that we minimise to obtain the best perspective transformation (homography) matching the actual sport court with its projection in the image. From such homography we

recover the camera pose (position and orientation in the 3D space). We present experiments in simulated and real sport scenarios.

7. M. Alemán-Flores, L. Alvarez, P. Henriquez, A. Trujillo: **Augmented reality in sport scenarios using cameras mounted on a tripod.** Technical Report - Centro de Tecnologías de la Imagen, (2012).

   In this paper we address the problem of inserting virtual content in a video sequence. The method we propose uses just image information. We perform primitive tracking, camera calibration, real and virtual camera synchronisation and finally rendering to insert the virtual content in the real video sequence. To simplify the calibration step we assume that cameras are mounted on a tripod (which is a common situation in practise). The primitive tracking procedure, which uses lines and circles as primitives, is performed by means of a CART (Classification and Regression Tree). Finally, the virtual and real camera synchronisation and rendering is performed using OpenGL (Open Graphic Library) functions. In order to illustrate its performance, it has been applied to real HD (High Definition) video sequences, specifically, soccer matches. The quality of the proposed method is validated by inserting virtual elements in such HD video sequence.

8. L. Alvarez, L. Gomez, P. Henriquez, J. Sánchez: **Real-time camera motion tracking in planar view scenarios.** Journal of Real-Time Image Processing, (2013). (Submitted)

   This is a novel method for real-time camera motion tracking in planar view scenarios. This method relies on the geometry of a tripod, an initial estimation of camera pose for the first video frame and a primitive tracking procedure. This process uses lines and circles as primitives, which are extracted applying CART (Classification and Regression Tree). We have applied the proposed method to HD (high definition) videos of soccer matches. Experimental results prove that our proposal can be applied to processing high definition video in real time. Finally, to illustrate the quality of the camera motion tracking, we validate the procedure by inserting virtual content in the video sequence.

9. M. Alemán-Flores, L. Alvarez, L. Gomez, P. Henriquez: **Camera Calibration in Sport Event Scenarios.** Pattern Recognition, (2013). (Submitted)

   The main goal of this paper is the design of a novel and robust methodology for calibrating cameras from a single image in sport scenarios, such as a soccer field, or a basketball or tennis court. In these sport scenarios, the only references we use to calibrate the camera are the lines and circles delimiting the different regions. The first problem we address is the extraction of image primitives also considering the challenging problems of shaded regions and lens distortion. From these primitives, we automatically recognise the location of the sport court in

the scene by estimating the homography which matches the actual court with its projection onto the image. This is achieved even when only a few primitives are available. Finally, from this homography, we recover the camera calibration parameters. In particular, we estimate the focal length as well as the position and orientation in the 3D space. We present some experiments on models and real courts which illustrate the accuracy of the proposed methodology.

Conclusions

The aim of this dissertation is the development of digital video processing methods. In this document we present works which contribute to the literature in some important procedures related with digital video processing, such as, camera calibration, primitive detection, lens distortion correction and graphic insertion into video sequences.

The first presented work was a new technique for image thick line centres extraction, based on morphological operators. The proposed method works properly even in complex scenarios where we have to deal with interlaced broadcast images or large shaded areas. We observed that most of the significant thick line centres are extracted from the images, and the amount of spurious false thick lines is small. Moreover, these false detections could be easily removed in a post-processing stage where we search for straight lines and ellipses in the image based on the extracted thick line centres.

One of the main works is the proposal of camera calibration methods. First of all, we started studying the problem of camera calibration from only one image, and afterwards, we tackled the methods for calibrating video cameras.

For the problem of camera calibration from only one image, we developed a method which can be applied in planar view scenarios. In this kind of scenarios some difficulties can be found, because there are situations where only a few primitives to estimate the camera pose are visible . With the proposed method, we demonstrated that if there is a minimum number of visible primitives in the scene, it is possible to calculate the transformation from the image plane to reference plane. The method is based on building candidate homographies using the extracted lines from the image. Then, we have to find the homography which minimises the loss function. From that homography we can recover the camera parameters, focal length and extrinsic parameters.

As an extension of the previous work, we worked on a method for video camera calibration. Specifically, in scenarios where the camera is mounted on a tripod, which

is a very common situation in the sport events broadcasting. We studied the tripod geometry, and its features strongly simplifies the problem and allow to calibrate frames where the standard techniques fail. In the proposed model, one of the main novelties is that the tripod rotation centre and camera projection centre are not supposed to be the same. This is important due to, in professional cameras, the distance between the tripod rotation centre and camera proyection centre is significant.

When dealing with video sequences, we can use certain features to improve the camera calibration through the frame sequence. Therefore, we have performed a camera motion tracking procedure which relies on a homography estimation and a primitive tracking. For the primitive tracking, we proposed a new method based on a CART (Classification and Regression Tree). The decision tree is built by means a learning process which uses a training set formed by the classes obtained in one frame segmentation. We experimented with HD videos, where the proposal demonstrated to be fast and accurate classifying the primitives and the background. The maximum classification error was 0.16 %. On the other hand, the combination of the homography estimation and primitive tracking improves the processing time without losing accuracy. To test the procedure efficiency, we applied it in some experiments, specifically in HD videos of soccer matches. The results were precise and fast, the average processing time of a HD frame is only 5 milliseconds. The reason why the processing time is small is that the computation of a decision tree is very fast and the primitive tracking method is local. That is, it is not necessary processing all the image pixels, we only need to analyse a pixel neighbourhood near the primitive location, according to an estimation of its location obtained from the primitive location in previous frames.

We have to take into account the effects of the lens distortion in the digital video processing, because the lens distortion model may vary through the video sequence. In this work, we have developed new mathematical models for the lens distortion variation in cameras with zoom. Such mathematical models are based on a second order polynomial approximation and the minimisation of a global error energy. The approximation accounts for the variation of the radial distortion parameters through the range of zoom lens settings, and the global error is calculated measuring the distance between sequences of distorted aligned points and straight lines after lens distortion correction.

As an application of the proposed methods, we worked on the insertion of virtual elements into video sequences. Specifically with sport scenarios and using cameras mounted on a tripod. We have chosen this kind of experiments because these applications require an accurate and fast video calibration estimation. In our approach, when the calibration is done, we can synchronise the virtual camera with the real camera. We obtain the virtual camera parameters from the real camera parameters. Finally, we render the graphics using OpenGL due to it provides an easy virtual camera ma-

nagement and an optimised graphic processing on the graphic card. We have inserted virtual elements in different positions and heights into the HD videos used in the other experiments in this thesis.

Sometimes, when virtual graphics are inserted into a video, small perturbations over time of $P(t)$, $T(t)$, and $Z(t)$ values produce small oscillations in the camera motion disturbing the observer. To remove such perturbations we have proposed a new variational approach to smooth the camera movement through the video sequence. By using the proposed variational technique we strongly reduce such disturbing oscillations affecting the included graphic objects.

Capítulo 1

Estado del arte

Tradicionalmente la producción televisiva se hacía para ser vista generalmente en televisores de 19", hoy en día existen varias opciones incluyendo la definición estándar (*SD*), la televisión de alta definición (*HDTV*), *Internet*, teléfonos móviles y *PDA*. Esto significa que el proceso de producción debe replantearse, ya que por ejemplo, la producción hecha para *HDTV* es diferente que la de *SD*. Hay que tener en cuenta que si se produce para teléfonos móviles, el monitor será muy pequeño, o si se emite por *Internet* el tamaño del vídeo debe ser adecuado para la velocidad de transmisión en la red. El tener que producir salidas de vídeo para ser visualizados con distinta calidad de imagen, hace necesario un procesado de vídeo por computador, tanto para editar el tamaño como la calidad, para que puedan ser utilizados en diferentes soportes de difusión. Todo esto genera nuevos procesos de producción. Una nueva forma de producción es la televisión interactiva o por *Internet*. Esta televisión permite enlaces a páginas *web*, foros, mensajes instantáneos, correo electrónico, biografías de deportistas y entrenadores, variedad de cámaras y estadísticas. Se puede ver este tipo de emisión tanto por televisión como por teléfonos móviles e *Internet*. Por ello necesita también de un procesado de vídeo especial para su difusión. Otra producción reciente es la *HDTV*, que cambia con respecto a la *SD* debido a que la *HDTV* es un formato en auge y tiene diferentes características. Como son el número de líneas de barrido, la claridad de imagen, mayor profundidad de campo, se pueden percibir más detalles y las cámaras ofrecen mayor control técnico. Una de las aplicaciones más interesantes de la *HDTV* es la producción de eventos deportivos, ya que la calidad de imagen que ofrece dicho formato hace que el espectador esté inmerso en la acción. Además de ser interesante para el usuario, también lo es desde el punto de vista científico-tecnológico, dado que, para la producción de un evento deportivo se utiliza una gran variedad de tipos de cámaras, puntos de vista, y diferentes montajes y combinaciones de las tomas

de todas las cámaras siguiendo la acción. Por otra parte, en este tipo de producciones son importantes los gráficos por computador, ya que en estos eventos se incluyen gráficos habitualmente para ofrecer información extra al espectador. Normalmente la producción de vídeo digital para su emisión pasa por las siguientes fases:

1. Capturar en formato digital las imágenes desde el sistema de adquisición (habitualmente una cámara).
2. Transmitir a un ordenador el vídeo capturado para su edición.
3. Editar el vídeo seleccionando escenas de interés, etc.
4. Procesar el vídeo resultante para facilitar la interpretación de la escena. Lo cual puede incluir la calibración de la cámara para situarla en el espacio, la detección de objetos de interés en la escena (por ejemplo jugadores en un campo de fútbol), la detección de movimiento en la escena, etc.
5. Incluir efectos especiales en el vídeo, como la eliminación de ruido, añadir gráficos (por ejemplo elementos publicitarios), la eliminación de objetos indeseables, etc.
6. Mezclar con el audio.
7. Transmitir el vídeo procesado para su visualización.

Como el trabajo de tesis doctoral se centra en las fases 4 y 5, se explican cada una de ellas. La fase 4, procesado de vídeo, dentro de la cual se pueden realizar varias tareas. Como pueden ser los procedimientos de mejora de vídeo, que están basados en el procesado tridimensional y espacio-temporal de una señal. Dentro de estos procesos de mejora se usan métodos de convolución y de filtrado. Una de las técnicas para mejorar el vídeo capturado usando convolución es el muestreado. Por ejemplo el muestreado espacio-temporal, en el que destaca el vídeo progresivo (sin entrelazado). Este método evita el parpadeo siendo más fácil realizar estimación de movimiento y esquemas de compensación en el vídeo. Por otra parte están los métodos de filtrado, que puede ser intrafotograma o interfotograma. El filtrado es apropiado para eliminar ruido de las imágenes. Dentro de la fase 4 también se encuentran los métodos de calibración de cámaras. Para realizar la calibración hay que reconstruir una escena real para obtener la perspectiva, posición de la cámara y otros parámetros.

Una vez finalizada la fase 4, el procesado del vídeo y el calibrado de las cámaras, ya se puede comenzar con la inserción de efectos especiales en el vídeo (fase 5). Gracias a la información que proporciona el calibrado de las cámaras se pueden proyectar elementos sintéticos en la imagen real. Todas estas técnicas deben tener en cuenta

que las retransmisiones deportivas suelen ser en directo, con lo cual, el tiempo de procesado del vídeo es importante. Por ello, es interesante que el procesado se haga con programación multihilo o paralela. Aprovechando la capacidad de computación de los nuevos procesadores con múltiples núcleos se reduce el tiempo de procesamiento del vídeo.

Profundizando un poco más en la inserción de gráficos, se puede observar que muchos de los objetos sintéticos se añaden a las imágenes reales mediante modelos 3D, que tienen un coste computacional elevado. Esto no es beneficioso para la rapidez con la que se obtiene un resultado. Así que se usan técnicas basadas en imágenes, que son una alternativa potente a las técnicas basadas en la geometría. Dichas técnicas se enmarcan en el modelado basado en imágenes (*image-based modeling and rendering, IBR*). Algunas de estas técnicas son: *plenoptic modeling, lumigraph*, mosaicos concéntricos, imágenes multiperspectiva, métodos de transferencia, *view morphing*, *3D warping*, modelo de textura mapeada, *view-dependent geometry* y *view-dependent texture*.

Como se ha podido observar, la producción televisiva actual necesita de un procesado del vídeo para poder aprovechar bien las nuevas tecnologías y nuevas formas de retransmisión de los eventos. Dadas las características de dichos eventos, la edición del vídeo debe ser no lineal. Este tipo de edición consiste en el volcado de todo lo que se graba, directamente en el computador. Luego se realizan las operaciones de edición necesarias y finalmente se copia a un fichero de salida con un formato determinado para su difusión.

En este capítulo se va a entrar más en detalle del estado del arte de ciertos procedimientos mencionados anteriormente, y que forman parte del procesado de vídeo digital. Concretamente se estudia la calibración de cámaras (Sección 1.1), la extracción de líneas en las imágenes (Sección 1.2), corrección de la distorsión de la lente (Sección 1.3), y la inserción de gráficos en escenas reales (Sección 1.4).

1.1. Calibración de cámaras

Por calibración de cámaras se entiende la estimación de la rotación y traslación de la cámara en el espacio 3D (parámetros extrínsecos), así como los parámetros intrínsecos de la cámara (*zoom*, *aspect ratio* del píxel y la proyección del punto principal de la imagen). Para comprender las técnicas de calibración es necesario tener conocimientos de geometría proyectiva y modelos de cámaras basados en ella. En esta sección, primero se hace una pequeña introducción de los modelos matemáticos en los

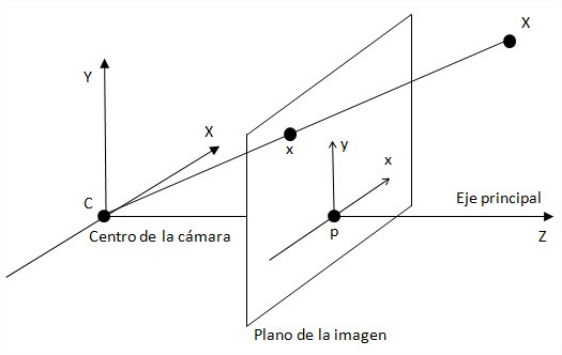

Figura 1.1: Geometría del modelo *pinhole*. C es el centro de la cámara y p el punto principal.

que se basa la calibración de cámaras y los parámetros importantes, y luego se recorre el estado del arte de las técnicas de calibración de cámaras.

1.1.1. Modelo de cámara

El modelo más comúnmente utilizado es el modelo *pinhole*, es lineal y se considera ideal y que no presenta distorsión de la lente. El modelo *pinhole* está representado en la Figura 1.1.

Sea el centro de proyección el origen de un sistema de coordenadas Euclídeas, y considerando el plano $z = f$, que se denomina plano de la imagen o plano focal. Con el modelo *pinhole*, un punto en el espacio con coordenadas $X = (X, Y, Z)^T$ se proyecta a un punto en el plano de la imagen donde una línea une el punto X con el centro de proyección a través del plano de la imagen. Esto se ilustra en la Figura 1.1. Por triángulos similares, se puede calcular que el punto $(X, Y, Z)^T$ se proyecta en el punto $(fX/Z, fY/Z, f)^T$ en el plano de la imagen. Ignorando la última coordenada de la imagen, la Ecuación (1.1) describe la proyección central a partir de las coordenadas del mundo a las de la imagen. Es decir, se realiza un mapeado desde el espacio Euclídeo R^3 al espacio Euclídeo R^2.

$$(X, Y, Z)^T \longrightarrow (fX/Z, fY/Z, f)^T \tag{1.1}$$

Como se observa en la Figura 1.1, el centro de proyección es denominado como el centro de la cámara, también conocido como centro óptico. La línea que sale desde ese

punto y es perpendicular al plano de la imagen es el eje principal o rayo principal de la cámara, y el punto donde el rayo principal se encuentra con el plano de la imagen se llama punto principal. Aquí se asume que el origen de coordenadas en el plano de la imagen está en el punto principal y que la proporción de las dimensiones del píxel es 1 (el píxel es cuadrado). En la práctica puede que esto no ocurra, y la proyección cambiaría a:

$$(X, Y, Z)^T \longrightarrow (fX/Z + x_c, rfY/Z + y_c)^T . \tag{1.2}$$

Esto se puede expresar en coordenadas homogéneas de la siguiente manera:

$$\begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \longrightarrow \begin{pmatrix} fX + zx_c \\ fY + zy_c \\ Z \end{pmatrix} = \begin{bmatrix} f & 0 & x_c & 0 \\ 0 & r \cdot f & y_c & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}. \tag{1.3}$$

Se puede separar la matriz A, que se conoce como matriz de calibración o matriz de parámetros intrínsecos:

$$A = \begin{bmatrix} f & 0 & x_c \\ 0 & r \cdot f & y_c \\ 0 & 0 & 1 \end{bmatrix}. \tag{1.4}$$

donde f representa la distancia focal de la lente. r es el *ratio* entre el ancho y el alto del píxel (generalmente r está cercano a 1.) (x_c, y_c) es el punto principal dado por la proyección ortogonal del foco en el plano de la cámara (habitualmente es próximo al punto medio de la imagen).

Los parámetros extrínsecos son la rotación y traslación que habría que aplicar a un punto del mundo real para obtener las coordenadas de ese punto en la imagen. Esta relación se describe a continuación:

$$X_{cam} = \begin{bmatrix} R & -RC' \\ 0 & 1 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} = \begin{bmatrix} R & -RC' \\ 0 & 1 \end{bmatrix} X. \tag{1.5}$$

Donde X_{cam} son las coordenadas en la imagen, X las coordenadas del mundo real, R una matriz de rotación y C' las coordenadas del centro de la cámara en el sistema de coordenadas del mundo real. Poniéndolo todo junto se obtiene la siguiente fórmula:

$$x = AR\,[I| - C']\,X. \tag{1.6}$$

Se deduce la matriz de proyección en un modelo *pinhole*:

$$P = AR\,[I|-C']\,. \tag{1.7}$$

Esta matriz de proyección tiene nueve grados de libertad, tres por A (f, x_c, y_c), tres de R y los últimos tres de C'. Por tanto, para poder definir la matriz de proyección, habrá que disponer de al menos 5 puntos en correspondencia para obtener el sistema de ecuaciones. Tener un punto en correspondencia, es tener la información relativa a las coordenadas de ese punto tanto en la imagen como en el mundo real. Si se tiene un escenario de medidas conocidas y puntos de referencia que estarán siempre dentro de un conjunto de posibilidades, se pueden obtener varios puntos en correspondencia. Un caso ideal de calibración donde se pueden encontrar fácilmente puntos en correspondencia es la calibración de una cámara a partir de un patrón de calibración.

1.1.2. Calibración de una cámara a partir de un patrón de calibración

Se sabe que cuando se toma una fotografía de un escenario plano (en este caso un patrón de calibración), las posiciones de las primitivas (líneas, círculos, etc.) en la imagen están dadas por una transformación de perspectiva (homografía) de su posición real. En otras palabras, si se considera un patrón plano compuesto por cualquier conjunto de primitivas, con sus dimensiones reales, entonces existe una homografía que hace corresponder el patrón real de las primitivas con su proyección en la imagen.

El objetivo de calibrar la cámara es obtener la transformación geométrica que permita mapear puntos de la imagen a puntos en coordenadas del mundo real. Como ambos, el patrón de calibración y la imagen, son planos, el mapeado es una homografía que puede ser escrita como una matriz de transformación $3x3$ denominada H, la cual transforma el punto de coordenadas del mundo real $p = (x, y, z)^T$ en el punto de coordenadas de la imagen $p' = (x', y', z')^T$ de la siguiente manera:

$$p' = Hp = \begin{pmatrix} h_{00} & h_{01} & h_{02} \\ h_{10} & h_{11} & h_{12} \\ h_{20} & h_{21} & h_{22} \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix}. \tag{1.8}$$

Como H es invariante en escala, hay ocho grados de libertad a determinar. Para ello, hay que utilizar puntos conocidos y sus correspondencias en la imagen.

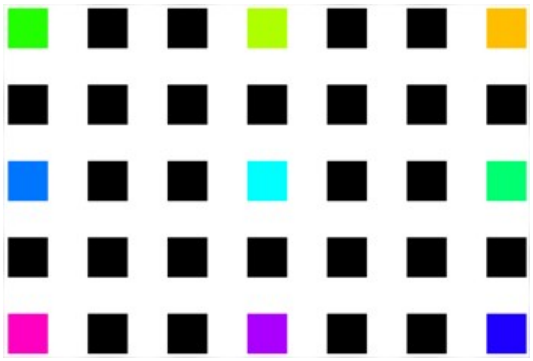

Figura 1.2: Patrón de calibración.

La calibración de una cámara usando un patrón de calibración se puede plantear en dos fases principales de cálculo, la de los parámetros intrínsecos y la de los extrínsecos. A continuación se describe el método de Zhang [88] para la calibración de cámaras usando un patrón de calibración: en primer lugar se calculan los parámetros intrínsecos de la cámara, definidos en la Ecuación (1.4). El proceso general para la obtención de estos parámetros en el caso de la calibración usando el patrón de la Figura 1.2, se divide en las siguientes etapas:

1. Se realizan varias fotografías del patrón desde diferentes posiciones.

2. Detección de los bordes de los cuadrados en cada imagen.

3. Identificación, a partir de los cuadrados de colores de la posición de los mismos.

4. Calcular la homografía entre el patrón de referencia y su imagen a partir de los bordes de los cuadrados detectados.

5. Obtención de los parámetros intrínsecos a partir de las homografías.

La fase de obtención de los parámetros intrínsecos se basa en los siguientes fundamentos matemáticos: para cada imagen I_m la homografía H_m del plano de referencia $z = 0$ al plano de la imagen I_m se expresa como:

$$H_m = sAR_m \begin{pmatrix} 1 & 0 & -c_x^m \\ 0 & 1 & -c_y^m \\ 0 & 0 & -c_z^m \end{pmatrix}, \tag{1.9}$$

de donde se obtiene:

$$H_m^T A^{-T} A^{-1} H_m = s^2 \begin{pmatrix} 1 & 0 & -c_z^m \\ 0 & 1 & -c_y^m \\ -c_z^m & -c_y^m & (c_y^m)^2 + 2(c_z^m)^2 \end{pmatrix}, \tag{1.10}$$

sustituyendo A, definida en la Ecuación (1.4), por su valor obtenemos:

$$H_m^T \begin{pmatrix} 1 & 0 & -x_c \\ 0 & \frac{1}{r^2} & -\frac{1}{r^2} y_c \\ -x_c & -\frac{1}{r^2} y_c & \frac{1}{r^2} y_c^2 + x_c^2 + f^2 \end{pmatrix}, H_m = s \begin{pmatrix} 1 & 0 & -c_z^m \\ 0 & 1 & -c_y^m \\ -c_z^m & -c_y^m & (c_y^m)^2 + 2(c_z^m)^2 \end{pmatrix}. \tag{1.11}$$

Sea $b = (b_1, b_2, b_3, b_4) = (\frac{1}{r^2}, x_c, -\frac{1}{r^2} y_c, -\frac{1}{r^2} y_c^2 + x_c^2 + f^2)$. De la igualdad anterior se obtienen dos ecuaciones en los coeficientes de b:

$$\begin{aligned} h_{21}^m(b_1 h_{21}^m + b_3 h_{31}^m) + h_{11}^m(b_2 h_{31}^m + h_{11}^m) + h_{31}^m(b_2 h_{11}^m + b_3 h_{21}^m + b_4 h_{31}^m) - h_{22}^m(b_1 h_{22}^m + b_3 h_{32}^m) \\ + h_{12}^m(b_2 h_{32}^m + h_{12}^m) + h_{32}^m(b_2 h_{12}^m + b_3 h_{22}^m + b_4 h_{32}^m) = 0 \\ h_{22}^m(b_1 h_{21}^m + b_3 h_{31}^m) + h_{12}^m(b_2 h_{31}^m + h_{11}^m) + h_{32}^m(b_2 h_{11}^m + b_3 h_{21}^m + b_4 h_{31}^m) = 0, \end{aligned} \tag{1.12}$$

coleccionando estas relaciones lineales se llega al sistema $Db = z$ donde D es una matriz $2m \times 4$. Minimizando $||Db - z||^2$ se consigue b como la solución de $D^T Db = D^T z$ y a partir de b se calcula A.

Una vez calculados los parámetros intrínsecos, el cálculo de los parámetros extrínsecos se realiza a partir de la homografía y los parámetros intrínsecos. Despejando de la Ecuación (1.9) se obtiene:

$$R = sA^{-1}H \begin{pmatrix} 1 & 0 & -c_x \\ 0 & 1 & -c_y \\ 0 & 0 & -c_z \end{pmatrix}^{-1} = sA^{-1}H \begin{pmatrix} 1 & 0 & -\frac{c_x}{c_z} \\ 0 & 1 & -\frac{c_y}{c_z} \\ 0 & 0 & -\frac{1}{c_z} \end{pmatrix}. \tag{1.13}$$

Igualando las dos primeras columnas de las matrices de la Ecuación (1.13) se consigue la matriz de rotación e igualando la tercera columna se obtiene el foco.

Un caso de aplicación real en el que se usa un patrón de calibración, es la calibración de cámaras en escenarios deportivos. Ya que se cuenta con un escenario conocido, el

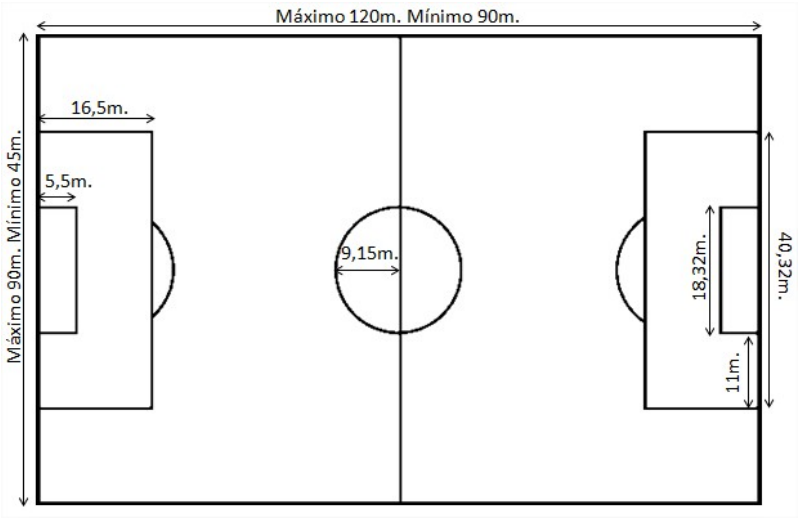

Figura 1.3: Medidas oficiales de un campo de fútbol según la FIFA.

cual puede usarse a modo de patrón de calibración. El patrón serían las líneas que delimitan el terreno de juego, dado que la forma y dimensión de las canchas siempre tienen que cumplir unas medidas reglamentarias. Por ejemplo, en el caso del fútbol, según la *FIFA* (Federación Internacional de Fútbol Asociación), los terrenos de juego deben cumplir ciertas restricciones de medidas que se reflejan en la Figura 1.3.

Centrando la atención en el caso de la calibración de cámaras en eventos deportivos, se observa que existen ciertas similitudes con el proceso de calibración usando el patrón sintético. Tales como que las líneas blancas de las canchas pueden proporcionar la misma información de la posición de la cámara que los bordes de los cuadrados del patrón de calibración. Por ello se usan estas primitivas para calibrar las cámaras en este tipo de escenarios reales. Por ejemplo, en [52] se necesitan las líneas blancas para obtener la homografía de la cámara. También se hace uso de las primitivas en [20], en el cual se necesitan cuatro puntos de intersecciones de líneas o dos puntos de intersección del círculo con la línea central y con la línea perpendicular a ella que cruza el centro del campo (esta línea no se pinta en el terreno de juego). Por otra parte, los autores de [59] usan para el cálculo de los parámetros los puntos del círculo central, de la línea de mediocampo y de las líneas de banda. Esta técnica con modelos de canchas y correspondencias con los puntos de la imagen real, se puede aplicar a varios escenarios deportivos como se plantea en [36]. Dicho artículo presenta un algoritmo de calibración en tiempo real que puede aplicarse a todas las canchas deportivas simplemente cambiando el modelo de la cancha. El algoritmo está basado en un detector de líneas especializado, una estimación de los parámetros de las líneas basado en *RANSAC*, una optimización

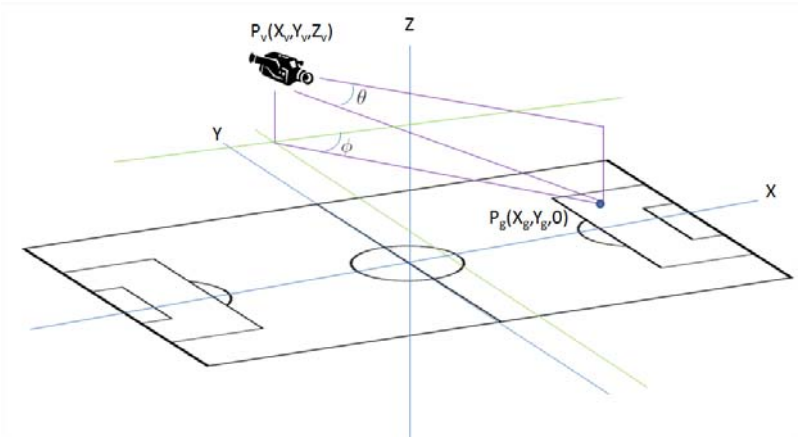

Figura 1.4: Modelo del que se obtienen las expresiones 1.14 y 1.15.

combinatoria para localizar la cancha en el conjunto de líneas y por último un seguimiento iterativo de modelos de canchas. Concretamente para vídeos de fútbol, en [19], se detalla un método para recuperar los parámetros de la cámara usando un modelo del campo de fútbol como el de la Figura 1.3. Utilizando el modelo matemático de la Figura 1.4, van realizando una búsqueda en el rango de los parámetros x_v, y_v, z_v, x_g, y_g para encontrar una configuración que al aplicarla al modelo coincida con la perspectiva del campo en la imagen. Variar dichos parámetros equivale a mover la cámara en sus ángulos de rotación según las siguientes expresiones:

$$\theta = \tan^{-1}\left(\frac{y_v - y_g}{x_v - x_g}\right) \tag{1.14}$$

$$\phi = \tan^{-1}\left(\frac{z_v - z_g}{\sqrt{(x_v - x_g)^2 + (y_v - y_g)^2}}\right) \tag{1.15}$$

En dicho artículo, se predefinen unos rangos de búsqueda para los parámetros, ya que normalmente la cámara está situada en un lateral del centro del campo. También usan como punto de partida las coordenadas del fotograma anterior, dado que los movimientos de la cámara son suaves y normalmente hacia la misma dirección.

Otra forma de utilizar las líneas de la cancha es la que explican los autores de [18], que usan los puntos de fuga de las líneas para calcular los parámetros de la cámara.

En este tipo de retransmisión de eventos deportivos, las cámaras suelen estar montadas en un trípode. Estas cámaras tienen una localización fija, pero pueden rotar libremente y cambiar sus parámetros intrínsecos al hacer *zoom*. En los trabajos [2, 68] se describen algunos métodos de calibración de cámaras con rotación y *zoom*. El trabajo [61] introduce optimizaciones de calibración para las cámaras *PTZ (Pan-Tilt-Zoom)*. Para calibrar las cámaras montadas sobre un trípode, se deben tener en cuenta los cambios en el modelo de cámara, como se explica en [45].

1.2. Extracción de primitivas

En la sección anterior, se ha visto que para calibrar las cámaras en ciertos escenarios, es necesario detectar primitivas en las imágenes. Como es el caso de la calibración usando patrones en escenarios reales, en concreto los eventos deportivos. Para disponer de la información de perspectiva que proporcionan las primitivas que se encuentran en la imagen, primero hay que detectarlas con cierta precisión e incluso puede ser necesario detectar los centros, si se trata de imágenes de alta definición cuyas primitivas pueden llegar a tener varios píxeles de grosor. La extracción de primitivas y sus centros es un problema importante en visión por computador, porque se trata de un procedimiento necesario para el desarrollo de otros procesos, como la calibración de cámaras, eliminar distorsión de las imágenes, control guiado de robots, etc.

Para detectar primitivas en la imagen, la mayoría de los métodos llevan a cabo una segmentación de la imagen y una detección de bordes. Por ejemplo cuando se realiza la calibración del patrón hay que detectar los bordes de los cuadrados y después asociar esos puntos a las ecuaciones de las primitivas. Otra opción en escenarios más complejos, es la segmentación para encontrar las primitivas, como por ejemplo las líneas de los campos de fútbol o de una carretera. El método descrito por [20], consiste en cuatro pasos: detectar la región del terreno de juego, analizar la región, detectar y reparar bordes, y localización de las líneas. Para localizar la región del terreno de juego, se utiliza un modelo de mezcla gaussiano. El análisis de la región está enfocado a detectar qué tipo de vista se está tratando, puede ser vista media (del centro del campo) o vistas de las áreas. Después se extraen los bordes de la imagen usando un detector Gaussiano-Laplaciano. Luego se reparan los bordes aplicando un crecimiento local de los píxeles vecinos, consiguiendo así que parte de los bordes de las líneas sean totalmente conexos. Finalmente se utiliza la transformada de Hough para extraer las líneas de la imagen resultante.

1.2.1. Detección de bordes

Como se explica en [51], los bordes de una imagen se pueden definir como transiciones entre dos niveles de gris significativamente distintos. Éstos suministran información sobre las fronteras de los objetos, que puede ser utilizada para la segmentación de la imagen, reconocimiento de objetos, visión estéreo, etc. La mayoría de las técnicas para detectar bordes emplean operadores locales basados en distintas aproximaciones discretas de la primera y segunda derivada de los niveles de grises de la imagen. Algunos de los métodos más usados basados en la primera derivada son los operadores de Roberts, Prewitt y Sobel. También son bastante usuales los que están basados en la segunda derivada como el Laplaciano, Marr y Hildreth. En la Figura 1.5, se puede observar el resultado de una detección de bordes en la imagen del patrón de calibración. Por otra parte, y a diferencia de los métodos anteriores, se encuentra el operador de Canny, que está basado en un desarrollo analítico de optimización partiendo de un modelo continuo unidimensional de un escalón. Considérese una función escalón con amplitud h_E afectada por un ruido blanco gaussiano de desviación típica σ_n. Supóngase que la detección de bordes se va a llevar a cabo mediante la convolución de una función continua unidimensional $f(x)$ con una función respuesta impulsional antisimétrica $h(x)$, que tiene amplitud cero fuera del intervalo $[-W, W]$. El borde buscado de la función $f(x)$ se marcará donde aparezca el máximo local del gradiente, obtenido tras la convolución de $f(x)$ con $h(x)$. Para determinar la función $h(x)$ buscada se exige que ésta satisfaga los siguientes criterios:

- **Buena Detección.** Se maximiza la amplitud de la relación señal-ruido (snr) del gradiente para obtener una alta probabilidad de marcarlo donde no lo hay. La relación señal-ruido para el modelo considerado es:

$$snr = \frac{h_E}{\sigma_n} S(h), \tag{1.16}$$

  con

$$S(h) = \frac{\int_{-W}^{0} h(x)dx}{\int_{-W}^{W} [h(x)]^2 \, dx}, \tag{1.17}$$

- **Buena localización.** Los puntos del borde marcados por el operador deben estar tan cerca del centro del borde como sea posible. El factor de localización se define como:

$$LOC = \frac{h_E}{\sigma_n} L(h), \tag{1.18}$$

  con

$$L(h) = \frac{h'(0)}{\int_{-W}^{W} [h(x)]^2 \, dx}, \tag{1.19}$$

donde $h'(0)$ es la derivada de $h(x)$. De acuerdo con estos dos criterios, el detector óptimo de bordes de tipo escalón es un escalón truncado (diferencia de escalones). Este operador, sin embargo, tiende a generar muchos máximos locales en su respuesta a bordes ruidosos de tipo escalón. Aunque estos bordes se deberían considerar erróneos de acuerdo con el primer criterio, sin embargo, no se ha considerado la interacción entre las respuestas en varios puntos próximos. Si se examina la salida del operador de diferencia de escalones, se ve que la respuesta a un escalón con ruido es un pico triangular con varios máximos en la vecindad del borde. Por ello es necesario incorporar el siguiente criterio que corrija este problema.

- **Respuesta única.** Debe haber una única respuesta para cada borde. La distancia entre picos del gradiente cuando sólo el ruido está presente, denotada por x_m, se establece como una fracción de k del ancho del operador, es decir:

$$x_m = kW. \tag{1.20}$$

  Canny combina los tres criterios minimizando el producto $S(h)L(h)$ sujeto a la restricción dada por la Ecuación (1.20). Debido a la complejidad de esta formulación no existe solución analítica de este problema.

Todo el desarrollo expuesto hasta ahora se refiere a una señal unidimensional continua. En el caso de imágenes digitales (bidimensionales y discreto), el operador propuesto por Canny se aproxima mediante la derivada de la Gaussiana en la dirección perpendicular al borde. En la práctica, aunque existen diferentes implementaciones, los pasos a seguir serían:

1. Calcular el módulo y la dirección del gradiente de la imagen suavizada aplicando el operador DroG (Derivada de la Gaussiana).
2. En la dirección del gradiente, eliminar puntos que no sean máximos locales del módulo (equivalente a encontrar el paso por cero en el operador LoG). Desechando los píxeles que no son máximos locales se mejora la localización y se evitan detecciones falsas.

El proceso de eliminación de no-máximos locales se suele implementar siguiendo el borde en la dirección perpendicular al gradiente, considerando los 8-vecinos. Normalmente se utiliza una función de histéresis tal que el primer píxel de un segmento del borde debe tener un módulo del gradiente que supere un valor *gradiente_alto*. Entonces se comienza a añadir píxeles vecinos en la dirección del borde (perpendicular al gradiente)

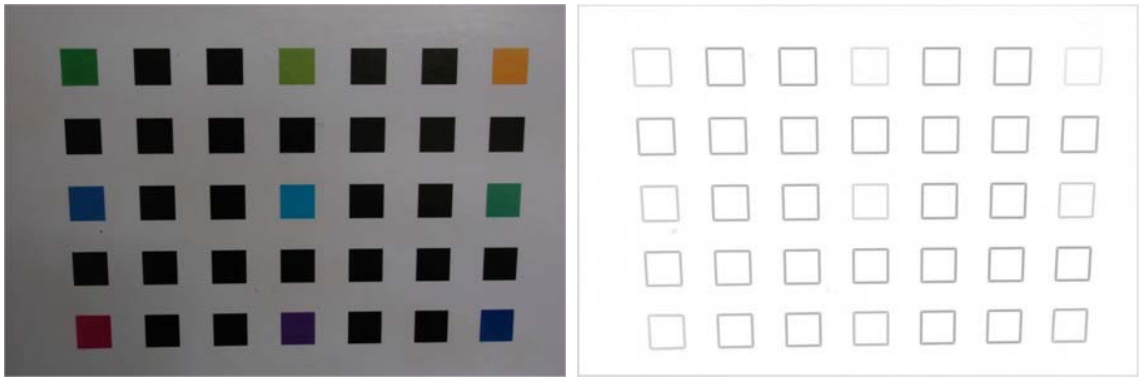

Figura 1.5: Detección de bordes con los operadores de Sobel (derecha) de la imagen del patrón de calibración (izquierda)

mientras que el valor del módulo de éste no caiga por debajo de un valor *gradiente_bajo* (menor que *gradiente_alto*) [51]. El uso de este operador está bastante extendido en el procesado de imágenes. Para el caso en el que se usa como parte del proceso de calibración de cámaras o procesado de vídeo, se encuentran ejemplos de uso como [50] y [89].

1.2.2. Segmentación para detectar primitivas

Existen escenarios donde las primitivas tienen un determinado contraste con el fondo o siempre son de un color similar, como por ejemplo las líneas de las carreteras o las de los campos de fútbol. En este tipo de escenarios, conociendo dicha información, se puede aplicar un proceso de segmentación para localizar las primitivas en la imagen antes de detectar bordes o incluso sustituir la fase de detección de bordes. En la mayoría de los procedimientos de segmentación para extraer primitivas, lo más normal es realizar una segmentación de la imagen detectando el color dominante, véase por ejemplo [63]. Dicho color suele corresponder con el color de fondo, por ejemplo el asfalto o el césped. En el trabajo [84], se extrae el color dominante del césped en secuencias de partidos de fútbol. El método utilizado se aplica en el espacio de color *RGB*. Se calculan los histogramas de los canales R,G y B, los valores de los picos de cada uno de ellos representan el color dominante. En la siguiente expresión se muestra como extraer el

color dominante:

$$O(x,y) = \begin{cases} 0, & si \begin{cases} I_G(x,y) > I_R(x,y) \\ I_G(x,y) > I_B(x,y) \\ |I_R(x,y) - R_{pico}| < R_t \\ |I_G(x,y) - G_{pico}| < G_t \\ |I_B(x,y) - B_{pico}| < B_t \\ G(x,y) < G_{th} \end{cases} \\ 255, & \text{en otro caso} \end{cases} \tag{1.21}$$

donde $O(x,y)$ es la salida binarizada, siendo el valor 0 el asignado al color que se etiqueta como dominante. I_R, I_G y I_B indican el valor de los canales RGB en un píxel. R_t, G_t y B_t son los valores de los umbrales para cada canal. Los autores configuran estos umbrales con los valores 10,15,10. Estos valores prefijados para el umbral pueden variar por las condiciones de iluminación o la varianza del color. Por lo tanto, se controlan los valores del umbral de acuerdo a la desviación de los picos de los histogramas. Si los valores de desviación son mayores que los prefijados, se pueden cambiar los umbrales por valores mayores. $G(x,y)$ representa el nivel de gris y G_{th} es el valor del umbral para $G(x,y)$. Y sirve para distinguir las primitivas del fondo, ya que normalmente las líneas blancas tienen un valor en el nivel de gris mayor. Siguiendo la estrategia de estudiar los histogramas, se encuentran [28] y [78], en los se que introduce el estudio en el espacio *HSV*, para mejorar el comportamiento en situaciones donde hay cambios de iluminación. Se describe el color dominante como el valor medio de cada componente del espacio *HSV*, el cual se computa alrededor de los picos de cada histograma. El cómputo implica determinar el índice de cada pico, i_{pico}, para cada histograma, el cual puede obtenerse de uno o más fotogramas. Después, se define un intervalo $[i_{min}, i_{max}]$ donde se encuentra el pico, satisfaciendo las siguientes condiciones, donde H es el histograma de color:

$$H[i_{max}+1] < K * H[i_{pico}] \leq H\left[i_{max}\right], \tag{1.22}$$

$$H[i_{\text{mín}} - 1] < K * H[i_{pico}] \leq H[i_{min}], \tag{1.23}$$

$$i_{min} \leq i_{pico} \leq i_{max}. \tag{1.24}$$

El pico debe tener un número mínimo de píxeles. En su implementación han fijado el número en el 20 % del conteo del pico, es decir $K = 0{,}2$. Finalmente calculan el color medio en cada componente dentro del intervalo detectado con la siguiente expresión:

$$Cm = Q_{size} * \sum_{i=i_{min}}^{i_{\text{máx}}} H[i] * i / \sum_{i=i_{min}}^{i_{\text{máx}}} H[i]. \tag{1.25}$$

donde Q_{size} es el tamaño de cuantización y se usa para convertir un índice en un valor de color. Se asumen valores diferentes de H,S y V. Luego en cada fotograma, los píxeles del fondo se detectan calculando la distancia entre el color de cada píxel y el color medio con una métrica cilíndrica robusta:

$$d_{cilindrica}(j) = \sqrt{(d_I(j))^2 + (d_C(j))^2}, \tag{1.26}$$

$$d_I(j) = |I_j - \bar{I}|, \tag{1.27}$$

$$d_C(j) = \sqrt{(S_j)^2 + \left(\bar{S}\right)^2 - 2S_j\bar{S}\cos(\theta(j))}, \tag{1.28}$$

$$\theta(j) = \begin{cases} \triangle(j) & if \triangle(j) \leq 180^{\text{o}} \\ 360^{\text{o}} & \text{en otro caso} \end{cases}, \tag{1.29}$$

$$\triangle(j) = |H - Hj|. \tag{1.30}$$

donde H,S y V se refieren a los valores del píxel j, los valores $\bar{A}$ indican el valor del color dominante. Luego se establece que los píxeles que estén por debajo de un cierto umbral en su distancia cilíndrica, serán los seleccionados como fondo, y el resto podrá estudiarse para ver si son primitivas u otros objetos de interés. Otro método [54], propone una combinación de los procedimientos explicados anteriormente. Se aplica también sobre el espacio HSV. Expone que los valores de los umbrales propuestos por [84] deben variar de una secuencia a otra, dependiendo del tiempo, la iluminación, etc. Se plantea una mejora en el algoritmo, de tal manera que se extraiga el color de fondo automáticamente, independientemente de las condiciones de la secuencia. Se plantea que la distribución del histograma del color de fondo para cada componente de color no es simétrico con respecto a los valores de los picos. Por lo tanto, la selección del valor del pico se hace de la siguiente manera:

$$A'_{pico} = \frac{\Sigma_{H(i)\geq\alpha H(A_{pico})} iH(i)}{\Sigma_{H(i)\geq\alpha H(A_{pico})} H(i)}, \tag{1.31}$$

donde A_{pico} se refiere al valor de pico del canal, $H(i)$ es el valor del índice i-ésimo del histograma, a indica qué índices deben ser seleccionados con respecto al conteo del pico. Se proponen las siguientes ecuaciones para seleccionar los valores de los umbrales:

$$A_t = std(I_A(x,y)) \text{ para } (x,y) \in H_A(I_A(x,y)) \geq \alpha A_{pico}, \tag{1.32}$$

$$GL_t = GL_{pico} + \beta * std(GL(x, y)). \tag{1.33}$$

donde β es una coeficiente predefinido y std() indica la desviación estándar de A_{pico}. Se asume que α y β son 0,1 y 0,75 respectivamente. Por otra parte, existen métodos de extracción de líneas que comienzan segmentando la imagen con un modelo de mezcla Gaussiano. Por ejemplo, en [80], donde se proyectan al espacio LUV todos los píxeles de los fotogramas seleccionados para la segmentación. Cada dimensión se discretiza en 64 valores y luego se aplica un modelo de mezcla Gaussiano consistente en dos Gaussianas $G(\mu_1, S_1), G(\mu_2, S_2)$. Estas funciones se usan para estimar la distribución del color con el algoritmo EM (Esperanza-Maximización). Para facilitar la convergencia de dicho algoritmo, se inicializan las medias de las Gaussianas μ_1, μ_2 correspondiendo a dos picos de la distribución del color. La media μ_1 se inicializa al color c_0, el pico principal del histograma, y μ_2 se inicializa al segundo pico, c, del histograma, que maximiza Dc x Fc. Donde Dc es la distancia cartesiana de c a c_0, y Fc es la frecuencia del color c en los fotogramas. La dependencia de la distancia permite encontrar dos picos distintos y evitar que las Gaussianas modelen el mismo color. Esencialmente, el segundo pico es el color más frecuente que no esté tan cerca del primer pico. Una vez se tiene el modelo estimado, se determina si la superficie del color de fondo está formada por un color simple o por dos. Si los picos de las dos Gaussianas estimadas están cerca uno del otro ($||\mu_1 - \mu_2||$ <umbral), o una de ellas es significativamente más débil que la otra ($F\mu_1 << F\mu_2$), se considera que existe un solo color dominante, y se modela con una Gaussiana simple. Para cada modelo Gaussiano, se puede expresar la probabilidad condicional p de un píxel dado x como:

$$p(x|\mu_k, \Sigma_k) = \frac{\exp\left[-(x-\mu_k)^T \Sigma_k^{-1}(x-\mu_k)/2\right]}{2\pi|\Sigma|^{1/2}}, \tag{1.34}$$

para $k = 1, 2$. Un píxel x se considera del color dominante cuando $p(x|\mu_1, S_1) > t$ o $p(x|\mu_2, S_2) > t$, siendo t el valor del umbral. Para refinar el modelo del color dominante, los datos de los fotogramas se filtran usando el modelo actual, y se usan los datos filtrados para re-estimar el modelo Gaussiano.

En [58] se usa un modelo de mezcla gaussiano adaptativo ($AGMM$) y un umbral para encontrar la zona correspondiente al fondo. La ventaja de usar un $AGMM$ es que los parámetros del modelo pueden actualizarse durante el proceso mediante maximizado de expectación incremental (IEM). Se observa que usando los histogramas del espacio de color C_bC_r en las secuencias que tienen un color dominante, existen pocas regiones del histograma distintas de cero y en general hay algunos picos. Normalmente esos picos corresponden al color dominante, pero pueden aparecer excepciones. Para determinar la región principal del histograma, la cual corresponde al color dominante, se procede de la siguiente forma:

1. Determinar el pico principal P_1.
2. Encontrar la región conexa alrededor de P_1. Solamente los índices que tengan valores superiores a T*Valor(P_1). Sumar todos los índices conectados (Sum_1) y luego restar la región conexa. Donde T es un porcentaje que los autores han puesto a 0,05.
3. Buscar el pico P_2 en los valores restantes del histograma, y realizar la suma de la región conexa en Sum_2.
4. Devolver la región más grande de entre Sum_1 y Sum_2.

Nótese que solo la región con una suma mayor de índices se considera como color principal, esto evita que se considere un color en un índice aislado. Después de la detección de regiones, se usa el *GMM* para modelar el color principal, como se describe a continuación:

$$G = \Sigma_{i=1}^{k} \pi_i G_i G_i\left(X; \theta_i\right) = \frac{1}{(2\pi)^{d/2}|\Sigma_i|^{1/2}} \exp(-\frac{1}{2}(X-\mu_i)^T(\Sigma_i)^{-1}\left(X-\mu_i\right)\Sigma_{i=1}^{k}\pi_i) = 1, \quad (1.35)$$

cada componente G_i es una función Gaussiana, parametrizada por θ_i, que consiste en el vector de media μ_i y la matriz de covarianza Σ_i. La dimensión de los datos de ejemplo X es d. El conjunto π_i, θ_i de todos los parámetros desconocidos pertenece a algún espacio de parámetros. Generalmente, esos parámetros se estiman con el algoritmo EM, con el que se actualizan los parámetros del modelo durante la ejecución. En esta propuesta, se incorporan tres mezclas al modelo, siendo dos de ellas usadas para modelar el color principal y otra modela el ruido.

1.2.3. Extracción de la ecuación de las primitivas

Una vez que se han detectado los puntos de los bordes de los objetos de interés, hay que agruparlos en primitivas. Para poder obtener sus ecuaciones y usarlas para cálculos subsiguientes. Debido a ciertas imperfecciones que se pueden encontrar en los datos de la imagen o en los proporcionados por el detector de bordes, puede producirse una pérdida de puntos pertenecientes a las primitivas, o aparecer ruido que dificulte la identificación de las mismas. Por estas razones, no es siempre una tarea fácil agrupar

los puntos extraídos por el detector de bordes en un conjunto apropiado de primitivas. Para resolver este problema, existen varios métodos en la bibliografía clásica de visión por computador. En general, las primitivas que se suelen extraer son líneas rectas y círculos, por ello se explicarán por separado los métodos usados para cada tipo de primitiva.

Extracción de líneas

Uno de los principales métodos utilizados para este fin, es la transformada de Hough, cuyo propósito es abordar este problema haciendo posible el agrupamiento de puntos en primitivas candidatas realizando un procedimiento de votación sobre un conjunto de primitivas parametrizadas. El caso más simple de la transformada de Hough es la transformación lineal para detectar líneas rectas. En el espacio de la imagen, una línea recta se describe como $y = mx + b$. La idea principal de esta transformada es considerar las características de la recta no como puntos de la imagen sino en términos de sus parámetros. Por ejemplo, la pendiente m y la intersección b, basándose en el hecho de que una línea puede representarse también como un punto (b, m) en el espacio paramétrico. Sin embargo, aparece el problema de que las líneas verticales pueden tener valores infinitos de m. Por razones computacionales, es mejor usar otra parametrización basada en el ángulo de orientación de la recta y su distancia al origen. El parámetro r representa la distancia entre la línea y el origen, mientras que θ es la orientación de la recta. Usando esta parametrización, la ecuación de la recta se puede escribir como:

$$r = x\cos\theta + y\sin\theta \tag{1.36}$$

Por lo tanto es posible asociar a cada línea de la imagen un par (r, θ) que es único si $\theta \in [0, \pi)$ y $r \in \mathbb{R}$ o si $\theta \in [0, 2\pi)$ y $r \geq 0$. El plano (r, θ) es llamado a veces el espacio Hough para el conjunto de rectas en dos dimensiones. Esta representación hace que la transformada de Hough esté conceptualmente muy cerca de la transformada Radon, también usada para detectar líneas (véase [90]). Para un punto cualquiera en la imagen, cuyas coordenadas sean por ejemplo (x_0, y_0), las líneas que pasan por dicho punto son las definidas por los pares (r, θ) con $r(\theta) = x_0\cos\theta + y_0\sin\theta$ donde r es la distancia entre la línea y el origen determinada por θ. Esto corresponde a una curva senoidal en el plano (r, θ), que es única para ese punto. Si la curva correspondiente a dos puntos se superimponen, la localización (en el espacio Hough) donde se cruzan corresponde a una línea (en el espacio original de la imagen) que pasa por ambos puntos. De manera más general, un conjunto de puntos que forman una línea recta producen senos que se cruzan en los parámetros para esa línea. Entonces, el problema de detectar puntos colineales se convierte en encontrar curvas concurrentes. El algoritmo de la transformada de Hough usa un vector, llamado acumulador, para detectar la existencia

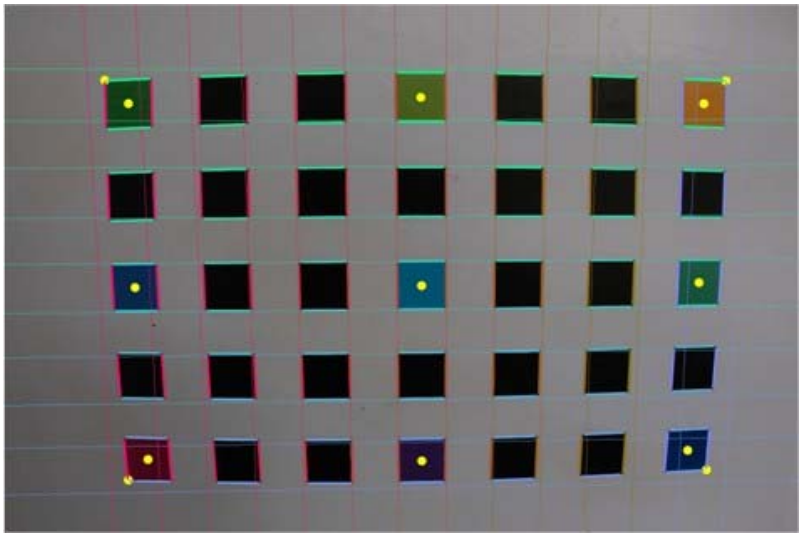

Figura 1.6: Líneas extraídas con la transformada de Hough, a partir del resultado de la detección de bordes.

de la línea $y = mx + b$. La dimensión del acumulador es igual al número de parámetros desconocidos del problema de la transformada de Hough. Por ejemplo, el problema de la transformada de Hough para las rectas tiene dos parámetros desconocidos (r, θ). Las dos dimensiones del vector acumulador corresponden a los valores cuantizados para (r, θ). Para cada píxel de la vecindad que pertenezca al conjunto de bordes detectados, se calculan los parámetros de cada línea, y se mira en qué valores de parámetros cae y se incrementa el acumulador de esa recta. Al final, encontrando los parámetros más votados, que se obtienen buscando los máximos del acumulador, se extraen las rectas más probables (ver [70]). En la Figura 1.6 se muestran las líneas que se han extraído de la imagen del patrón de calibración.

Se pueden encontrar varios trabajos que usan la transformada de Hough para extraer líneas en escenarios reales, como por ejemplo en los escenarios deportivos. En este tipo de escenarios el uso de la transformada de Hough está muy extendido para la extracción de líneas blancas de las canchas deportivas, entre ellas las canchas de tenis (ver [86]) o los campos de fútbol (véase [18], [19], [20], [84]).

Otra opción utilizada para extraer primitivas de un conjunto de puntos es el *RANSAC* [31]. El nombre del método es la abreviatura de *RANdom SAmple Consensus*. Es un procedimiento iterativo para estimar parámetros de modelos matemáticos a partir de un conjunto de datos. Es un algoritmo no determinista, en el sentido de que produce resultados razonables solamente con una cierta probabilidad. La suposición básica es

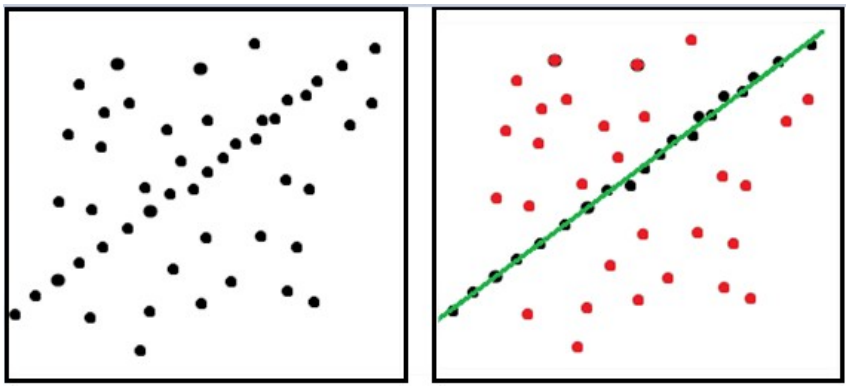

Figura 1.7: A la izquierda, datos observados. A la derecha modelo matemático de una línea que encaja con los *inliers*.

que el conjunto de datos está formado por *inliers*, datos cuya distribución se puede explicar con un conjunto de parámetros de un modelo, y *outliers*, los cuales no encajarían en el modelo, véase Figura 1.7.

La entrada del algoritmo de *RANSAC* es un conjunto de valores observados, un modelo parametrizado que puede explicar o encajar con las observaciones (por ejemplo líneas) y algunos parámetros de confianza. *RANSAC* consigue su objetivo seleccionando iterativamente un conjunto aleatorio de los datos originales. Estos datos son hipotéticamente *inliers* y la hipótesis se prueba de la siguiente manera:

- El modelo se encaja con los hipotéticos puntos *inliers*, por ejemplo todos los parámetros libres del modelo se construyen a partir de esos puntos.
- El resto de los datos se prueba con el modelo y si un punto encaja bien con el modelo estimado se considera también un *inlier*.
- El modelo estimado es razonablemente bueno si se han clasificado suficientes puntos como hipotéticos *inliers*.
- Este procedimiento se repite un número fijo de veces. Cada vez produce tanto modelos rechazados por tener pocos puntos *inliers*, como modelos hipotéticamente correctos y con una medida de error. Se considerará finalmente el modelo con el menor error.

Este procedimiento aplicado concretamente a la detección de líneas, se usa en escenarios reales como los eventos deportivos para extraer las líneas rectas, ver por ejemplo [37] y [59].

Extracción de círculos

Existen muchos métodos para extraer círculos o elipses de las imágenes, dos de ellos son generalizaciones de dos de las técnicas explicadas anteriormente para extraer líneas, la transformada de Hough y *RANSAC*. En la Sección 1.2.3 se explicó la transformada de Hough para encontrar líneas rectas, y dicha transformada puede usarse para encontrar cualquier forma que pueda ser representada por un conjunto de parámetros. Un círculo, por ejemplo, puede expresarse en un conjunto de tres parámetros, convirtiendo el espacio de Hough en tridimensional. Un círculo con radio R y centro (a, b) puede escribirse con ecuaciones paramétricas como:

$$\begin{aligned} x &= a + R\cos\theta \\ y &= b + R\sin\theta \end{aligned} \tag{1.37}$$

Cuando el ángulo θ completa el rango de 360 grados, los puntos (x, y) trazan el perímetro de un círculo. Si una imagen contiene muchos puntos, algunos de ellos coinciden en perímetros de círculos, entonces lo que hay que hacer es buscar los parámetros (a, b, R) que describen cada círculo. El hecho de que el espacio de Hough sea tridimensional hace que la implementación del método sea más compleja en términos de requerimientos de memoria y tiempo de procesamiento.

Hay que tener en cuenta, que en muchos escenarios reales, los círculos no aparecen en las imágenes como círculos perfectos debido a que se estén enfocando de forma oblicua y sufran distorsión por la perspectiva, convirtiéndose así en elipses. La detección de elipses también puede realizarse con la transformada de Hough, como se detalla en [85]. En este caso, una elipse se define por 5 parámetros, que tienen un amplio rango de valores y necesitan precisión alta, haciendo que la computación del problema sea muy costosa. Los autores de [85] proponen la detección de las elipses usando la información proporcionada por la orientación de los bordes y descomponiendo el problema en dos etapas de transformada de Hough que se ejecutan secuencialmente. La primera consiste en encontrar el centro de la elipse y la segunda en determinar los tres parámetros restantes.

Por otro lado, para aplicar otra de las técnicas descritas en la Sección 1.2.3, *RANSAC*, en la extracción de elipses, se comienza por seleccionar 5 puntos de los observados. Luego hay que computar la curva general de segundo orden que pasa por ellos, y determinar si el resultado se aproxima lo suficiente a una elipse, si no, hay que probar con mas selecciones de 5 puntos. Este procedimiento puede llegar a ser también bastante costoso.

Como se ha visto, la complejidad de la detección de elipses en las imágenes es alta, por esta razón hay muchas técnicas que intentan mejorar la solución a este problema. Como por ejemplo, en el trabajo [19] utilizan un método basado en las tangentes de la elipse. Como se observa en la Figura 1.8, la propiedad que se usa en este método se expresa como sigue: Sean P_1 y P_2 dos puntos de la elipse, y T_1 y T_2 dos líneas tangentes a la elipse en los puntos P_1 y P_2. Sea J la intersección de las tangentes T_1 y T_2 y I es el punto medio entre P_1 y P_2. Entonces la línea JI incluye el centro de la elipse. Se usa esta propiedad para detectar los arcos de la elipse de la siguiente manera:

1. Caracterización de parejas relevantes punto-tangente: Se analiza una vecindad de cada punto para decidir si el punto pertenece a una línea recta, en tal caso, se considera que ese punto pertenece a una curva y la línea recta es su tangente. Este primer paso produce un conjunto de parejas punto-tangente que se usarán después para detectar el centro de la elipse.

2. Detección del centro de la elipse: Se consideran todas las parejas de parejas $(punto-tangente)_i$ y $(punto-tangente)_j$. Se calculan las líneas correspondientes $D_{ij} = X_{ij}Y_{ij}$ donde X_{ij} es el punto medio del segmento P_iP_j y Y_{ij} la intersección entre T_i y T_j. Se calculan todas las intersecciones entre D_{ij} y $D_{i'j'}$ y se estudia la distribución de esos puntos en el plano: las áreas de concentración corresponden al centro de las elipses.

3. Evaluación de a, b y θ: La elipse centrada en el origen, sólo depende de tres parámetros que son el ángulo θ entre el eje principal y O_x, y los parámetros a y b de su expresión normalizada. Cuando se ha encontrado el centro O, se puede expresar la ecuación de la elipse en referencia a O, siendo: $Ax^2 + Bxy + Cy^2$.

4. Detección de los arcos de la elipse: Para la elipse detectada, hay que determinar los puntos asociados y analizar su distribución para extraer los arcos correspondientes. Dependiendo del ángulo de la cámara, o de algún objeto en la imagen, generalmente, se puede encontrar visible solamente una parte de la elipse.

Otro método para detectar los arcos de las elipses se encuentra en el trabajo [81], donde se plantea un algoritmo basado en el método *LSF* (*Least Square Fitting*), el cual denominan algoritmo *ALSF* (*Advanced Least Square Fitting*). Después de realizar una detección de bordes, usan un algoritmo de crecimiento de semilla para detectar las áreas conexas, las partes con menor número de puntos que un cierto umbral serán eliminadas. Como los arcos son transformaciones de círculos parciales, deben formar parte de elipses. Como resultado, se puede usar el algoritmo específico de ajuste de elipses (*LSF*), pero no en el caso de que el centro no esté cerca del origen o el eje mayor sea muy largo, ya que las ecuaciones se vuelven inestables. Para evitar esto, proponen

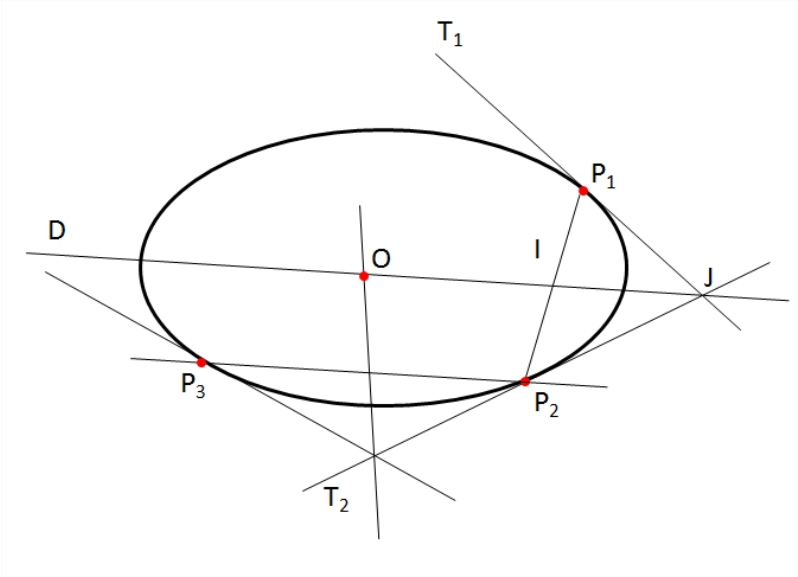

Figura 1.8: Propiedad de las tangentes de la elipse.

encontrar los valores mínimos y máximos de las coordenadas x e y de cada área conexa y luego transformar los puntos al intervalo $[0, 2]$. Después de la transformación, se puede representar el área como una cónica general mediante un polinomio de segundo orden. Se puede aproximar una cónica minimizando la suma de las distancias algebraicas al cuadrado. Después de resolver el problema del ajuste, se debe realizar un filtrado para eliminar posibles arcos detectados en objetos que no sean de interés.

1.3. Corrección de la distorsión de la lente

En la Sección 1.1, se puede observar que para calibrar una cámara hace falta corregir la distorsión de la imagen, porque en el modelo de cámara *pinhole* se asume que las líneas rectas del mundo real son proyectadas en la imagen como líneas rectas también. Pero en la realidad esto no siempre pasa así, debido a que las lentes pueden introducir una cierta distorsión, las líneas rectas pueden verse curvas en la imagen. Por ello, para poder utilizar líneas en los patrones de calibración, hay que corregir dicha distorsión de la lente. En el caso de trabajar con lentes que presenten una distorsión significativa resulta necesario tener en cuenta un modelo de distorsión radial en el modelo matemático de cámara utilizado. La distorsión radial suele ser la más común (ver [43]) y puede aparecer en dos variantes, la distorsión de barril (Figura 1.9 izquierda)

Figura 1.9: Distorsión de barril (izquierda), distorsión de cojín (derecha).

o de cojín (Figura 1.9 derecha). El modelo básico de distorsión de lentes que se usa en visión por computador para realizar la corrección de la distorsión (ver por ejemplo [25, 35, 73]) viene dado por la siguiente expresión:

$$\hat{\mathbf{x}} \equiv \tilde{L}(\mathbf{x}) = \mathbf{x}_c + L(r)(\mathbf{x} - \mathbf{x}_c) \tag{1.38}$$

donde $\mathbf{x} = (x, y)$ es el punto original de la imagen (distorsionado), $\hat{\mathbf{x}} = (\hat{x}, \hat{y})$ es el punto corregido, $\mathbf{x}_c = (x_c, y_c)$ es el centro del modelo de distorsión de la cámara, generalmente cerca del centro de la imagen, $r = \sqrt{(x - x_c)^2 + (y - y_c)^2}$ y $L(r)$ es la función que define la forma del modelo de distorsión. Usualmente, $L(r)$ se aproxima por el polinomio

$$L(r) = 1 + k_1 r^2 + k_2 r^4 + k_3 r^6 + \cdots , \tag{1.39}$$

donde el vector $\mathbf{k} = (k_1, k_2, \ldots, k_{N_k})^T$ representa los parámetros de distorsión radial. La complejidad del modelo viene dada por el grado del polinomio que se usa para aproximar $L(r)$ (por ejemplo la dimensión de $\mathbf{k}$).

La mayor limitación del modelo de distorsión anterior es que en la realidad, la distorsión de la lente depende de la distancia de los puntos de la escena a la cámara. Por tanto, el modelo habitual es una simplificación de la realidad y por ello la corrección de la distorsión no es perfecta en general, sobre todo cuando se trata de vistas en perspectiva, donde la distancia de los puntos de interés a la cámara varía mucho.

Una manera de medir la calidad de la lente es calcular la distorsión máxima en píxeles provocada por la misma. Cuanto menor es este valor, se tiene una lente de mayor calidad.

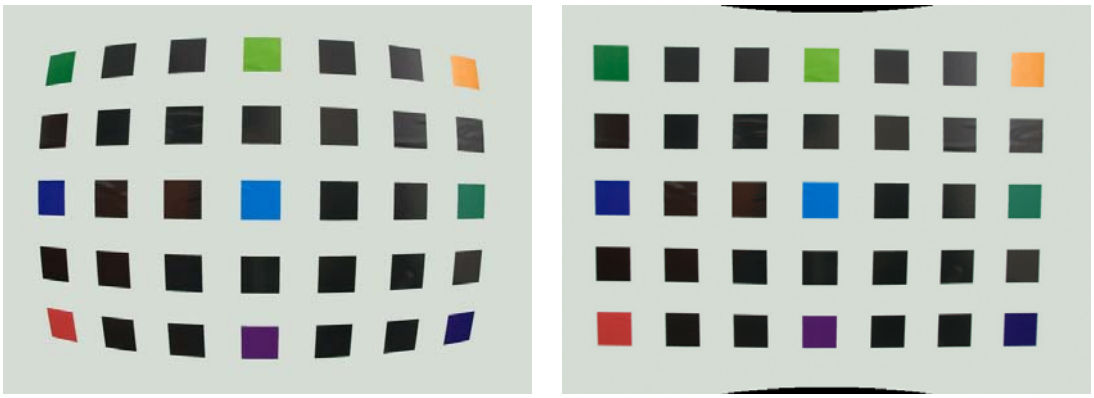

Figura 1.10: Imagen del patrón de calibración con distorsión de lente (izquierda). Imagen donde se ha eliminado la distorsión de la lente (derecha).

A continuación se muestra un ejemplo de como se elimina la distorsión de una imagen usando las líneas rectas distorsionadas detectadas en la misma y usando el modelo descrito en [11]. En el ejemplo se utiliza el patrón que aparece en la Figura 1.2, que es una imagen de dimensiones 1502×1000 píxeles, y el centro de distorsión utilizado es el centro de la imagen $(x_d, y_d) = (752, 500)$. En la Figura 1.10 se puede ver una fotografía del patrón donde se percibe la distorsión de la lente (izquierda), y también se muestra la misma imagen pero con la distorsión corregida (derecha).

Tradicionalmente, en la calibración de cámaras dependientes del *zoom* se modela la variación de los parámetros de la cámara (matriz de parámetros intrínsecos y matrices de rotación y traslación) con mínimos y máximos de *zoom* predefinidos. Ver [30], que usa un modelo de zoom para tener en cuenta solamente la variación de los intrínsecos, o [77] que presenta un modelo considerando las variaciones de los parámetros intrínsecos y extrínsecos.

Para calibrar una cámara sin un rango de *zoom* dado, se guardan en una tabla *look-up* un cierto número de ajustes de la lente y datos relacionados con la calibración (ver [74]). Así, para cada ajuste de la lente, se tiene que hacer un número considerable de medidas, necesitando una gran cantidad de tiempo. Entonces, los datos recolectados se procesan usando el método de mínimos cuadrados (Levenberg-Marquardt u otro método de optimización) y, aplicando un buen modelo, el resultado es la matriz de parámetros intrínsecos expresada como función del ajuste del *zoom* [3, 30, 57]. Debido a que la distorsión radial varía con los cambios de la distancia focal, el efecto de la distorsión de la lente normalmente se incluye como parte de los parámetros intrínsecos y se estima durante el procedimiento de calibración por medio de una corrección iterativa de las imágenes generadas por la cámara, como por ejemplo en [3, 16]. Para un análisis

más detallado del efecto de la distorsión radial de la lente en las cámaras ver [38], de donde se puede concluir que,

- la variación de la distorsión radial es no lineal con el *zoom*,
- la distorsión radial alcanza el máximo en la distancia focal más corta.

Además los autores de [38] muestran que la variación del primer coeficiente de la distorsión radial con el campo de *zoom* puede ser modelada como:

$$k_1^{c_i} = d_0 + d_1 c_i^{d_2}, \tag{1.40}$$

donde c_i es la distancia principal, d_i son coeficientes empíricos y, para las cámaras analizadas en [38], d_2 rangos desde -0.2 a -4.1. Estos resultados son para un ajuste del enfoque de 5 a 21 milímetros. En [55] los autores expusieron un método para corregir automáticamente la distorsión radial de la lente en una cámara con *zoom*. El método usa modelos de distorsión de un parámetro (por ejemplo solo se considera k_1) y dos modelos locales diferentes para tener en cuenta la distorsión de barril y la de cojín. Después de seleccionar algunas imágenes (fotogramas de vídeo) con diferentes distancias focales, los autores usan un sistema de hardware *POVIS* para estimar la distancia focal y k_1 para cada fotograma, después se aplica el método de mínimos cuadrados para ajustar un polinomio cuadrático para el primer coeficiente de distorsión radial k_1, teniendo como variable la inversa de la distancia focal f,

$$k_1(1/f) = c_0 + c_1(1/f) + c_2(1/f)^{-2}, \tag{1.41}$$

donde $\{c_i\}$ representa los coeficientes polinomiales. Para construir un modelo de distorsión de lente dependiente del *zoom* para un conjunto de m imágenes, se requiere estimar fotograma a fotograma el modelo de distorsión minimizando cualquier función apropiada de energía teniendo en cuenta la desviación entre los puntos distorsionados y los corregidos.

1.4. Inserción de gráficos en escenas reales

La inserción de gráficos en escenas reales es importante para la producción televisiva y cinematográfica, pero también tiene otras muchas aplicaciones en campos como el mantenimiento industrial, medicina, educación, entretenimiento y juegos (ver [22]).

La idea principal es añadir objetos virtuales dentro de una escena real. Dependiendo de la aplicación, los objetos insertados pueden ser instrucciones para reparar el motor del coche, la reconstrucción de un monumento arqueológico o publicidad. Para que estos efectos resulten creíbles, los objetos tienen que aparecer fijados al mundo real, lo cual requiere una medida precisa de la posición de la cámara. En la mayoría de las situaciones, existen algunos problemas prácticos para aplicar las técnicas de inserción de gráficos. Por ejemplo, normalmente es necesario colocar marcadores especiales en la escena, que interfieren en la apariencia o usar sistemas de seguimiento magnéticos o ultrasónicos, que limitan el movimiento y requieren un equipamiento especial. En producción televisiva se pueden usar cámaras equipadas con sensores, pero para emisiones en directo de eventos exteriores estos sistemas son difíciles de instalar además de costosos económicamente. Para los programas de televisión que incluyen un decorado virtual u objetos sintéticos, es necesario medir la pocisión de cada cámara del estudio para poder renderizar los objetos desde el punto de vista correcto. Las producciones en estudio que utilizan gráficos en tiempo real, generalmente usan un estudio equipado específicamente para tal fin. Pudiendo ser una solución basada en marcadores como por ejemplo el desarrollado por la BBC en [75]. Pero por otra parte, para producir los eventos deportivos, normalmente se usan cámaras con sensores de rotación, que pueden llegar a ser muy costosas y limitan el número de cámaras que se usan para cubrir dicho evento. Por todo lo mencionado, es interesante realizar la calibración de cámaras con aproximaciones basadas en la visión por computador como se explica en la Sección 1.1. Una vez se han obtenido los parámetros de la cámara mediante la calibración, ya se pueden añadir los gráficos a las escenas. Estos elementos sintéticos o gráficos se pueden dividir en dos tipos básicos: texto e ilustrativos. Los gráficos de texto se hacen con generadores de caracteres 2D o 3D, y los gráficos ilustrativos pueden generarse con simples PC o con complejos sistemas informáticos ejecutando programas de animación 3D en tiempo real. Aparte de los sofisticados efectos especiales utilizados actualmente en el cine, la inserción de gráficos es también importante en la retransmisión de eventos deportivos, ya que permiten ofrecer más información al espectador, por ejemplo:

- Gráficos de información virtual. Son gráficos virtuales introducidos en retransmisiones deportivas, justo en el terreno de juego, que dan información adicional. Como podría ser la inserción de una línea para marcar el fuera de juego en un partido de fútbol, o el FoxTrax, un sistema que incluye unos transmisores en el disco de hockey sobre hielo, y permite en el proceso de producción añadir gráficos de la trayectoria del mismo durante el juego para que el espectador pueda seguirlo con facilidad.

- Publicidad virtual, es la inserción de elementos publicitarios en las imágenes, pudiendo incorporarlos en cualquier lugar de la pantalla mediante pantallas verdes, o simulando estar sobre el terreno de juego o en la acción, previamente obteniendo

información de la colocación de las cámaras.

La inserción de gráficos virtuales puede dividirse en dos pasos principales: sincronización de las cámaras y renderizado. La sincronización de cámaras consiste en detectar la posición de una cámara en un vídeo real, y colocar en un mundo virtual una cámara en la misma posición y con la misma configuración de parámetros para obtener vistas de ese mundo con la misma perspectiva que la cámara real. Así, al renderizar, los objetos del mundo virtual podrán mezclarse con el vídeo del mundo real de una forma que parezca que forman parte de la escena real y no molesten la visión del espectador. Existen varias técnicas y librerías *software* para llevar a cabo estas dos etapas. Por ejemplo en los trabajos [48] y [62] se usa la matriz de proyección, obtenida en la fase de calibración de la cámara, para proyectar el contenido virtual en la imagen real, y renderizar el objeto virtual píxel a píxel. Para darle una apariencia más realista a la mezcla de imágenes virtuales y reales, en [23] se utiliza un técnica de mezclado consistente en utilizar colores que estén en armonía con la imagen real. Para mejorar la eficiencia del renderizado, se pueden usar librerías de gráficos, como se propone en [82]. Un problema común en todos los trabajos relacionados con la inserción de gráficos, es la segmentación de la imagen para detectar los píxeles que pueden ser reemplazados por objetos virtuales, por ejemplo el césped, y cuáles no pueden ser reemplazados como los jugadores. Para ello se utilizan técnicas de detección del color dominante, las cuales se explicaron en la Sección 1.2.2.

Capítulo 2

Localización de primitivas en la escena

2.1. Introducción

La localización de primitivas es una tarea importante en el procesado de secuencias de vídeo, porque suele ser el primer paso necesario para abordar otros problemas. Por ejemplo, guiar un coche sin conductor por dentro de la carretera o dirigir los movimientos de un robot. La detección de primitivas también es necesaria en el procesado de vídeos de eventos deportivos, porque forma parte de varios procesos, tales como la calibración de cámaras, *mosaicing*, detección de acciones destacadas (goles, faltas) o resúmenes automáticos. En la mayoría de los métodos de detección de líneas que se usan en las imágenes de eventos deportivos, se asume que los colores de las líneas y del terreno de juego son constantes en la región de interés. Como primer paso del procedimiento para detectar primitivas, lo más normal en los métodos es realizar una segmentación de la imagen detectando el color dominante en el espacio RGB [54, 78, 84], o el espacio HSV [28, 63, 83]. En estos métodos se suelen usar herramientas como los histogramas acumulativos [63], de los cuales se extraen los picos y se obtiene el color dominante, que normalmente coincide con el color del terreno de juego (que se denominará fondo de aquí en adelante), véase [54, 78, 84]. Por otra parte, existen métodos de extracción de líneas que comienzan segmentando la imagen con un modelo de mezcla Gaussiano [58, 80]. La gran mayoría de los métodos descritos anteriormente no ofrecen resultados muy precisos para imágenes entrelazadas, imágenes de alta definición (donde las líneas pueden ser de ocho píxeles de grosor o incluso más), o en escenarios con variaciones significativas del contraste entre fondo y primitivas.

Figura 2.1: Fotograma de un vídeo de alta definición de un partido de fútbol (imagen proporcionada por Mediapro).

En varios de los casos mencionados, es necesario además encontrar cuál es el centro de las líneas ya que los procesos posteriores necesitan precisión en la posición de las líneas. En este capítulo se analiza el problema de la detección de los centros de primitivas (líneas y círculos) en vídeos de escenarios reales, como por ejemplo vídeos de partidos de fútbol, para que posteriormente puedan ser utilizadas como información útil en los procesos de calibración de cámaras o de detección de acciones importantes en el partido. En esta clase de escenarios reales hay que enfrentarse a dificultades adicionales, como la distorsión en las líneas producida por el vídeo entrelazado o las zonas sombreadas en la escena que dificultan la segmentación (ver Figuras 2.1 y 2.2). El método propuesto en este capítulo usa como herramienta principal los operadores de morfología matemática, los cuales son bastante efectivos para extraer información geométrica de las formas en las imágenes. Para abordar el problema en el escenario real comentado anteriormente, la principal suposición que se hace es que, en la imagen, todas las líneas de interés son más claras (o más oscuras) que el fondo. Además, se puede asumir que el fondo mantiene un color uniforme. Por ejemplo, en las Figuras 2.1 y 2.2, se puede apreciar que el fondo es verde (el color del césped del campo de fútbol). Esta información es útil para evitar detectar líneas fuera de la región de interés, en este caso el terreno de juego. En esta situación, se consideran primitivas todas las líneas de un campo de fútbol, tanto las líneas rectas como los círculos, esto incluye las líneas de banda, líneas de fondo, línea de medio campo, áreas pequeñas y grandes, el círculo central y los semicírculos de las áreas.

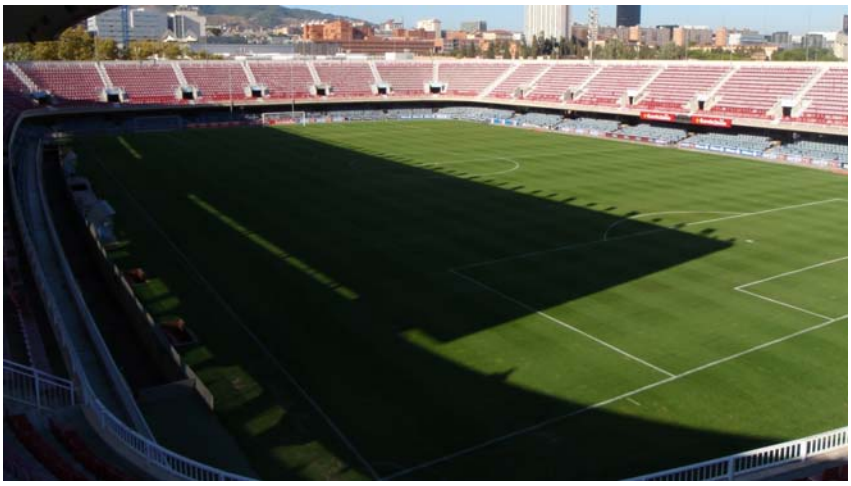

Figura 2.2: Imagen real de un estadio de fútbol con una zona sombreada (imagen proporcionada por Mediapro).

2.1.1. Contribución de este capítulo

Detección morfológica del centro de líneas:
Se ha desarrollado un método morfológico para detectar los centros de líneas en las imágenes, teniendo en cuenta que pueden estar en partes de la imagen con distinta iluminación y contraste. Utilizando el método propuesto, es posible detectar los centros de líneas de distinto grosor. El método aporta gran precisión en la detección de centros de líneas presentes en las imágenes de alta definición de escenarios reales, como son los vídeos de partidos de fútbol.

2.2. Morfología matemática

La morfología matemática es una teoría y técnica para el análisis y procesamiento de estructuras geométricas [69]. Puede ser usada de manera continua o discreta y en imágenes binarias o de escala de grises. Los operadores morfológicos básicos que se usan en este trabajo para localizar líneas y sus centros son los siguientes:

Operadores morfológicos de disco: Dado un disco $D_s(x)$ con centro x y radio s, se define:

Dilatación disco:	$I \oplus D_s(x) = \sup_{y \in D_s(x)} I(y)$,
Erosión disco:	$I \ominus D_s(x) = \text{ínf}_{y \in D_s(x)} I(y)$,
Apertura disco:	$I \circ D_s(x) = (I \ominus D_s) \oplus D_s(x)$,
Cierre disco:	$I \bullet D_s(x) = (I \oplus D_s) \ominus D_s(x)$.

Los operadores morfológicos de disco se usan para extraer las líneas de la imagen. Por ejemplo, se observa que, si el ancho de línea máximo en la imagen es s, y si las líneas en la imagen son más claras que el fondo, entonces la operación morfológica $I \circ D_s$ elimina las líneas.

Operadores morfológicos de línea: Dado un conjunto de orientaciones de ángulos Θ, $\theta \in \Theta$ y un segmento $L_{s,\theta}(x)$ de centro x,radio s y orientación θ, se define:

Apertura de línea:	$I \circ L_s(x) = \sup_{\theta \in \Theta}(\text{ínf}_{y \in L_{s,\theta}(x)} I(y))$,
Cierre de línea:	$I \bullet L_s(x) = \text{ínf}_{\theta \in \Theta}(\sup_{y \in L_{s,\theta}(x)} I(y))$.

Los operadores morfológicos de línea se usan en el método propuesto para filtrar el ruido de la imagen y limpiar los bordes de las líneas.

Esqueleto morfológico: Dado un conjunto X, el esqueleto morfológico está definido por:

Esqueleto morfológico: $S = \cup_{s>0} \left(\cap_{\mu>0} \left(X \ominus D_s \backslash (X \ominus D_s) \circ D_\mu \right) \right)$,
donde D_s es un disco de radio s centrado en 0.

El esqueleto representa, para una forma dada X, los centros de los discos más grandes incluidos en X. El esqueleto morfológico se usa en este trabajo para encontrar el centro de las líneas.

2.3. Procedimiento de desentrelazado usando filtro morfológico de líneas

La tecnología de vídeo entrelazado puede introducir fuertes perturbaciones en las líneas, especialmente cuando la cámara se mueve rápido y se está trabajando con vídeo de alta definición (fotogramas de 1920×1080 píxeles).

En la Figura 2.3, se observa este fenómeno en un fotograma de un vídeo real de alta

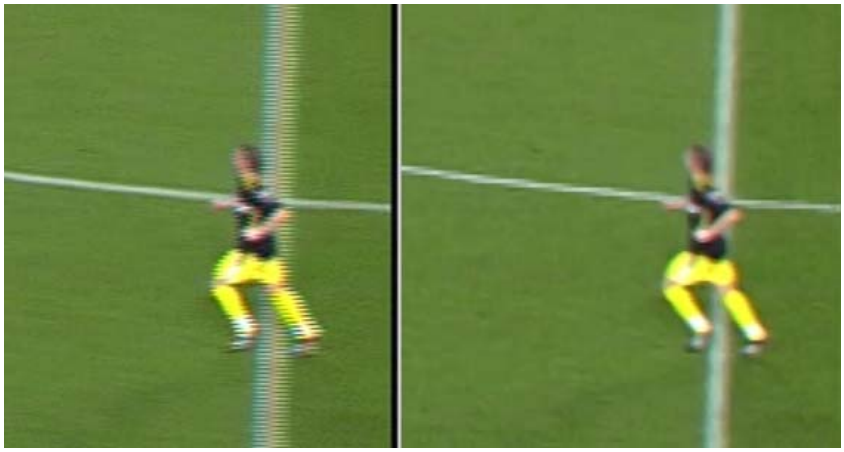

Figura 2.3: Imagen original entrelazada (izquierda) e imagen desentrelazada usando el algoritmo propuesto (derecha).

definición obtenido en un evento deportivo. Para encontrar correctamente las líneas en la imagen, es necesario realizar un preprocesado para eliminar el ruido. El desentrelazado de vídeo es un problema importante en visión por computador, pero el desarrollo de técnicas sofisticadas de desentralazado no es uno de los objetivos de este trabajo. De hecho, en este escenario, sólo es necesario un procedimiento de desentrelazado simple y rápido que elimine el ruido de las líneas de la imagen. Para ello se propone el siguiente método: reemplazar las líneas pares de la imagen por las impares y luego aplicar un operador morfológico lineal para limpiar las líneas. Esta operación se realiza independientemente en cada uno de los canales RGB de la imagen.

2.4. Detección del centro de las líneas en escenarios sin regiones sombreadas

Se comienza abordando el caso más simple, donde ninguna de las líneas de interés está localizada en una región sombreada. Primero, se realiza una apertura morfológica con un operador disco $I \circ D_s$ para encontrar las líneas. Se observa que si las líneas de la imagen son más brillantes que el fondo y la anchura máxima de línea es s, entonces, para cada uno de los canales RGB, la operación de apertura elimina las líneas de la imagen.

Por lo tanto, una primera aproximación a la región de líneas A, puede expresarse como:

$$A = \left\{ x : \left\{ \begin{array}{l} (R(x) - R \circ D_s(x)) > t_R \\ (G(x) - G \circ D_s(x)) > t_G \\ (B(x) - B \circ D_s(x)) > t_B \end{array} \right\} \right. . \tag{2.1}$$

donde t_R, t_G, t_B son umbrales para cada canal de la imagen. Se aprecia que el método propuesto es robusto frente a cambios en la iluminación, siempre y cuando el contraste entre líneas y fondo se mantenga lo suficientemente alto.

En el caso de que el color de fondo no cambie significativamente en la imagen, como en un campo de fútbol, donde el fondo es verde, se puede seleccionar la región de interés "a priori" en la imagen, de acuerdo con el color del fondo. Para manejar la información del color, es más cómodo trabajar en el espacio de color *HSV*, donde *Hue* (H) es el componente principal referente a la información del color. Se denota por $(H_s(x), S_s(x), V_s(x))$ los canales *HSV* de la imagen $(R \circ D_s, G \circ D_s, B \circ D_s)$. Entonces, el área de fondo puede expresarse como:

$$C = \{x : \ t_{H_1} \leq H_s(x) \leq t_{H_2}\} . \tag{2.2}$$

Esto significa, poner un umbral al valor de *Hue* H_s. Se observa que, como H_s se calcula después del proceso de apertura, las líneas de la imagen están incluidas en C. En otras palabras, el conjunto $A \cap C$ representa el conjunto final $\mathcal{B}$ de puntos de líneas que corresponde a las líneas localizadas en la región de interés. Los parámetros se eligen de la siguiente manera: s, el radio máximo de ancho de línea, se establece en 5 para estar seguro de que se incluyen todas las líneas de interés de la imagen. t_{H_1} y t_{H_2} se eligen analizando el pico del histograma del canal H_s usando una técnica de segmentación estándar de histogramas (véase [1] para más detalles). Como el área de las líneas es muy pequeña en comparación con el área del fondo, los parámetros t_R, t_G y t_B se escogen en términos de un porcentaje $0 < p < 1$ con respecto al histograma del canal correspondiente. Por ejemplo, t_R se elige para satisfacer:

$$p = \frac{|x \in C : (R(x) - R \circ D_s(x)) > t_R|}{|C|}$$

, donde $|.|$ representa el cardinal (tamaño) del conjunto. Para los experimentos se utilizó $p = 0{,}02$. La Figura 2.4 ilustra las líneas extraídas de una imagen de ejemplo sin grandes regiones sombreadas. Las Figuras 2.5 y 2.6 muestran ampliaciones de algunas zondas donde se detectaron líneas. Como se puede ver, se pueden localizar incluso líneas que ofrecen un bajo contraste con el fondo.

Figura 2.4: Líneas detectadas en la Figura 2.1.

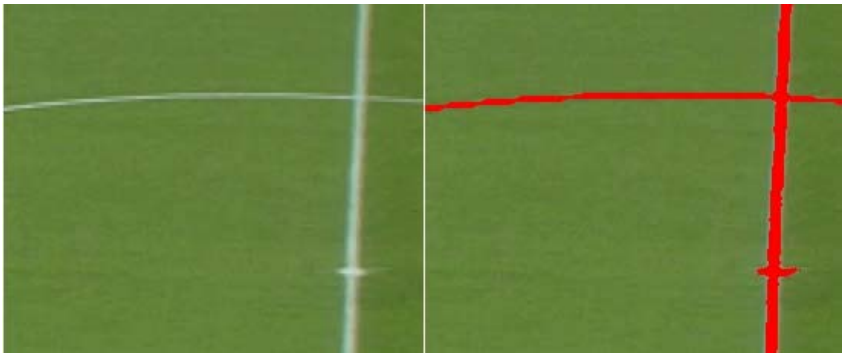

Figura 2.5: Detalle de las líneas extraídas de la Figura 2.1. Se aprecia que se detectan líneas de distinto grosor.

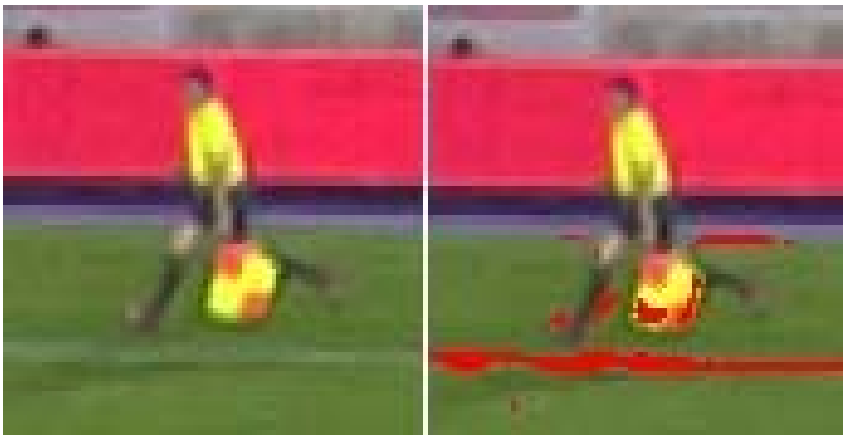

Figura 2.6: Detalle de las líneas detectadas en la Figura 2.1.

2.5. Detección de líneas en escenarios con regiones sombreadas

En situaciones con grandes regiones sombreadas, ver Figura 2.2, el método presentado en la sección anterior no funciona correctamente porque no se puede establecer una configuración de umbral simple con t_R, t_G y t_B que funcione simultáneamente en las regiones sombreadas e iluminadas. Por lo tanto, primero se necesita identificar que existen dos regiones de interés en la escena (las zonas sombreadas e iluminadas donde estén localizadas las primitivas) y seguidamente ajustar una configuración del umbral diferente para cada región.

Generalmente, el valor matiz o color (*Hue*) de la región de fondo no cambia significativamente de las zonas iluminadas a las sombreadas (por ejemplo, un campo de fútbol tiene un valor *Hue* similar, color verde, en las zonas iluminadas y sombreadas). Sin embargo el canal *Value* $V_s(x)$ en el espacio HSV varía significativamente de las áreas iluminadas a las sombreadas. Para identificar automáticamente si se trata de escenas con grandes áreas sombreadas se analiza el histograma del canal *Value* $V_s(x)$ pero en la región de interés definida por el canal *Hue*. Sea $h(w)$ el histograma de los valores de $V_s(x)$ en la región de interés, por ejemplo $H_s(x) \in [t_{H_1}, t_{H_2}]$. Si se trata solamente con una región, entonces $h(w)$ tiene un pico simple. Si por el contrario se trata de dos regiones, $h(w)$ tendrá dos picos. Usando una técnica estándar de segmentación de histogramas (ver [1]) se puede identificar automáticamente el número de picos significativos en $h(w)$. Una vez que se hayan separado las regiones iluminadas y sombreadas, se aplica el mismo procedimiento propuesto en la sección anterior a cada región y se obtiene la región de líneas para la imagen completa. La Figura 2.7 ilustra las primitivas detectadas en una imagen de ejemplo con grandes zonas sombreadas. La Figura 2.8 muestra que las líneas son extraídas correctamente en las regiones iluminadas y sombreadas. En la Figura 2.9 se ve que el círculo central se ha localizado de manera aceptable a pesar de ocupar zonas sombreadas e iluminadas al mismo tiempo y con un contraste muy pequeño en el área sombreada.

2.6. Detección del centro de primitivas usando un esqueleto morfológico

En el caso de distribuciones discretas, el esqueleto morfológico puede ser fijado de la siguiente manera: si se denota por D_n el disco de radio n centrado en 0, entonces,

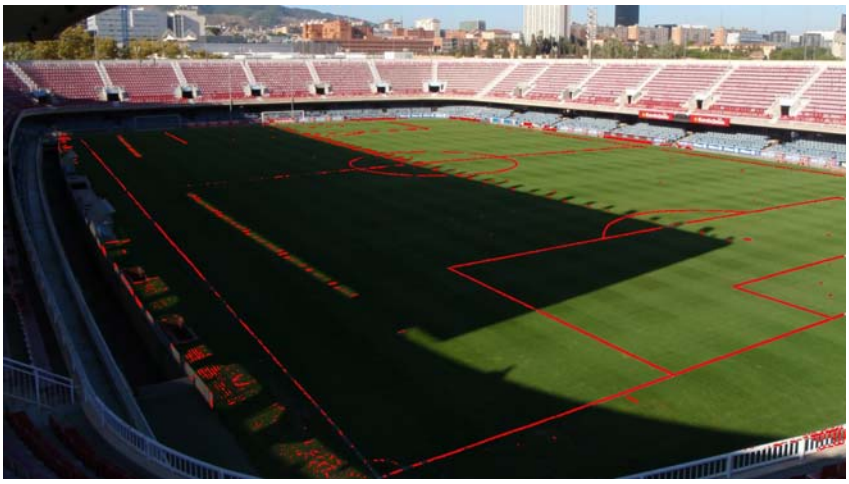

Figura 2.7: Primitivas detectadas en la Figura 2.2.

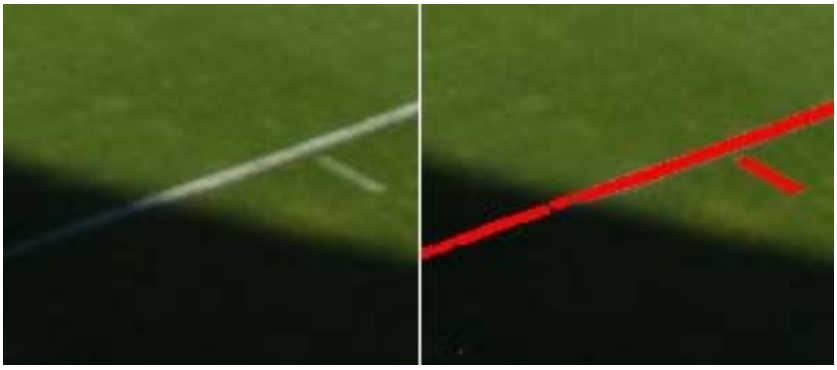

Figura 2.8: Detalle de las primitivas extraídas de la Figura 2.2.

Figura 2.9: Detalle de las primitivas detectadas en la Figura 2.2.

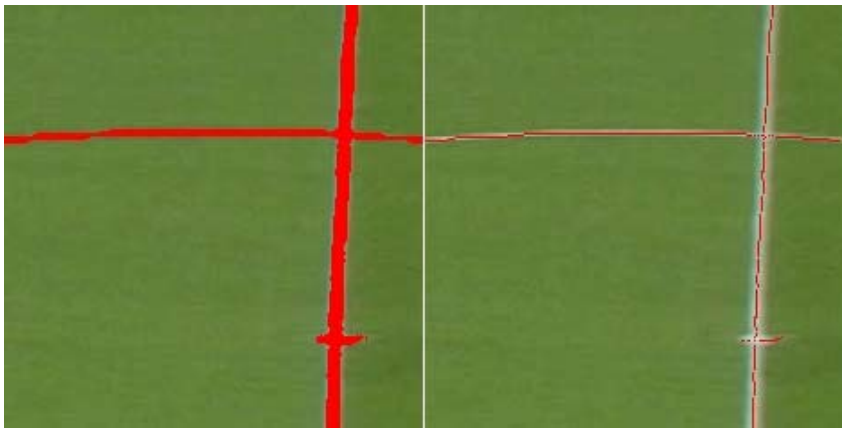

Figura 2.10: Detalle de los centros de líneas extraídos de la Figura 2.1.

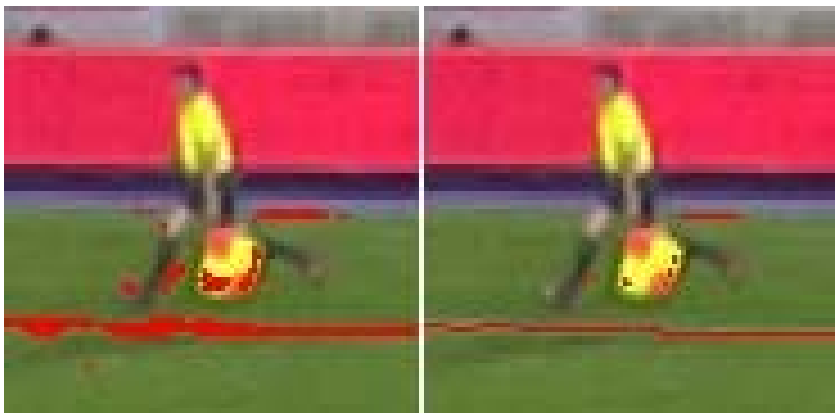

Figura 2.11: Detalle de los centros de las primitivas extraídos de la Figura 2.1.

los puntos centrales de la línea de ancho n pueden ser obtenidos como el conjunto:

$$S_n = (\mathcal{B} \ominus D_n) \setminus ((\mathcal{B} \ominus D_n) \circ D_1))$$

, donde $\mathcal{B}$ es la región de líneas detectada. Por consiguiente, el cálculo del esqueleto proporciona, automáticamente, el ancho de línea. Las Figuras 2.10, 2.11, 2.12 y 2.13 ilustran algunos ejemplos de la detección del centro de las primitivas en diferentes situaciones.

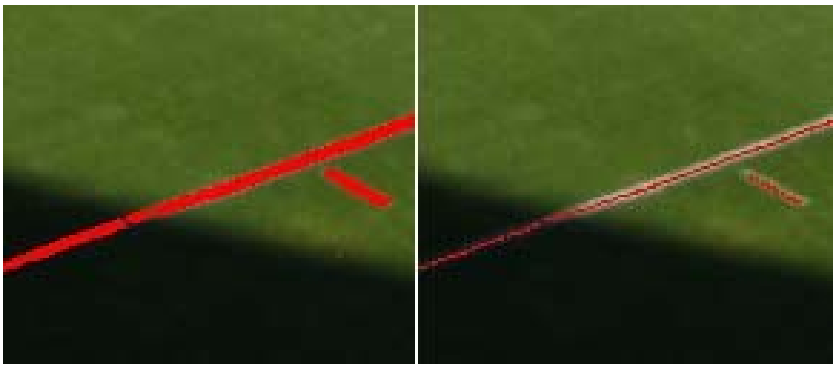

Figura 2.12: Detalle de los centros de las primitivas extraídos de la Figura 2.2.

Figura 2.13: Detalle de los centros de las primitivas extraídos de la Figura 2.2.

2.7. Conclusiones

En este capítulo se ha presentado una nueva técnica para la localización de primitivas y de su centro basada en operadores morfológicos. El método propuesto ha sido probado en situaciones reales y funciona correctamente incluso en escenarios complejos donde hay que tratar con imágenes entrelazadas o con grandes zonas sombreadas. En los experimentos se ha observado que se detectan la mayoría de las primitivas de interés y que la cantidad de líneas falsas detectadas es muy pequeña y aislada. Además, esas detecciones erróneas pueden eliminarse fácilmente en una fase de postprocesado donde se buscan líneas rectas y elipses en la imagen, basándose en el conjunto de centros de primitivas extraído.

Capítulo 3

Calibración de cámaras aisladas a partir de primitivas

3.1. Introducción

La calibración de cámaras es el proceso por el cual se recuperan los parámetros de la cámara con la que ha sido tomada una fotografía o con la que se ha grabado una secuencia de vídeo. Estos parámetros son la posición, rotación y parámetros intrínsecos. En este capítulo se aborda el problema de calibración de cámaras en escenarios planos como los eventos deportivos donde el terreno de juego es una superficie plana con características conocidas. Por ejemplo, las canchas de los diferentes deportes siempre tienen dimensiones conocidas dentro del reglamento y además cuentan con un número fijo de líneas o círculos (generalmente blancos) dividiendo las diferentes partes de la cancha. Ésta es una situación bastante común en los escenarios deportivos, por ejemplo: tenis, baloncesto o fútbol. Las primitivas de la cancha (líneas y círculos que dividen la cancha en diferentes partes) se usan como patrón de calibración para recuperar la posición de la cámara.

Cuando se toma una fotografía de un escenario plano, las posiciones de las primitivas en la imagen están dadas por una transformación de perspectiva (homografía) de su posición real. En otras palabras, si se considera un patrón plano compuesto por todas las primitivas de la cancha con sus dimensiones reales, entonces existe una transformación de la perspectiva que hace corresponder el patrón real de las primitivas con su proyección en la imagen. En la Figura 3.1 se ilustra esta transformación de perspectiva. En este trabajo se muestra que si un número mínimo de primitivas de la cancha es

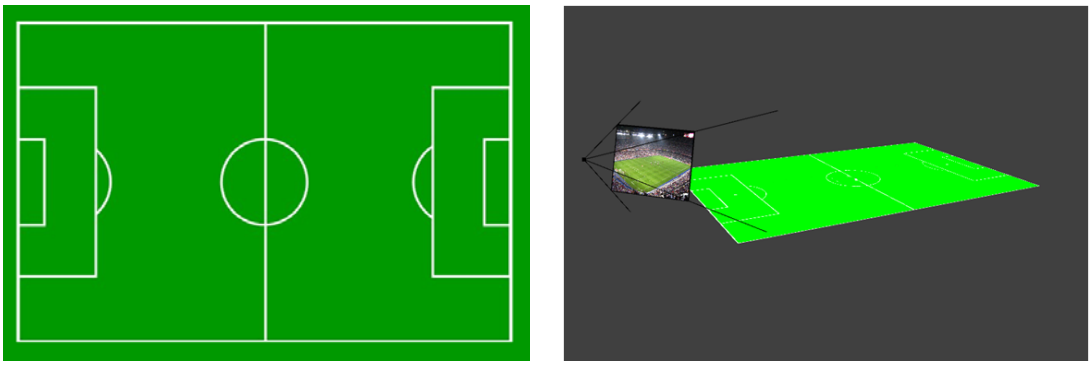

Figura 3.1: A la izquierda, imagen de un campo de fútbol real. A la derecha la posición de la cámara.

visible en la escena, entonces se puede recuperar la homografía que transforma el plano de la escena al plano de referencia real. Después, con esta homografía (y asumiendo que se conocen algunos parámetros internos de la cámara) se puede calibrar la cámara, es decir, recuperar la posición y orientación de la misma. Por tanto, el método que se propone para calibrar automáticamente la cámara puede dividirse en los siguientes pasos:

1. Extracción de primitivas planas (líneas y círculos) de la imagen.
2. Reconocimiento de la cancha. Se estima automáticamente la homografía que transforma la cancha real de referencia en su proyección en la imagen. Para ello, se propone un nueva estrategia para hacer coincidir las primitivas de la imagen con el modelo de referencia, usando homografías candidatas y una función de error.
3. Minimizar la función de error, usando las primitivas planas, la cual permite evaluar la calidad de la correspondencia entre el plano de la imagen y el real para una transformación de perspectiva específica (homografía).
4. Recuperar la posición de la cámara desde la homografía estimada.

En la primera fase de extracción de primitivas, se utiliza el método descrito en el Capítulo 2. Para llevar a cabo el siguiente paso, es necesario estimar el modelo de distorsión para poder obtener una evaluación precisa de la calidad de la correspondencia entre las primitivas de referencia y las de la imagen. Para tratar la distorsión de la lente, se usan los métodos detallados en [5, 10]. En la Figura 3.2 se observan las diferentes etapas del proceso de calibración propuesto.

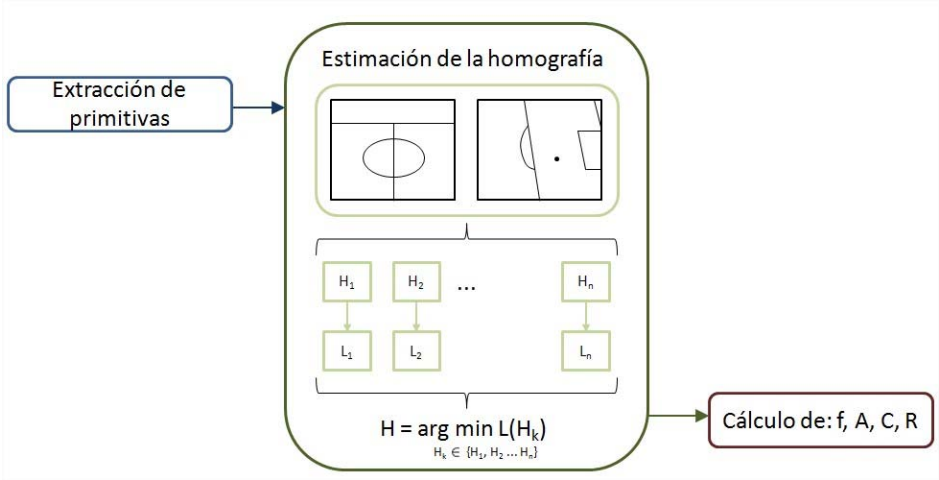

Figura 3.2: Etapas del proceso de calibración.

3.1.1. Contribución de este capítulo

Reconocimiento automático de la posición de la cámara en escenarios planos:
En este capítulo, se ha propuesto un método cuyo objetivo principal es reconocer automáticamente la posición de la cámara a partir de una imagen simple de un escenario plano. Se ha aplicado esta técnica a escenarios de eventos deportivos usando como información la líneas y círculos que dividen las diferentes partes del terreno de juego. Usando estas primitivas de la cancha, se define una función de error que se minimiza para obtener la mejor transformación de perspectiva (homografía), haciendo coincidir una cancha real con su proyección en la imagen. De dicha homografía se recupera la posición y orientación de la cámara en el espacio 3D.

3.2. Modelo de cámara

El modelo de cámara que se usa en este trabajo es el denominado modelo *pinhole*, donde se ha incluido un modelo de distorsión radial de lente. Usando este modelo de cámara, un punto 3D (X, Y, Z) se proyecta en la imagen en el punto 2D (x, y) en

coordenadas proyectivas de la siguiente manera:

$$\begin{pmatrix} d_x(x,y) \\ d_y(x,y) \\ 1 \end{pmatrix} = sAR \begin{pmatrix} 1 & 0 & 0 & -c_x \\ 0 & 1 & 0 & -c_y \\ 0 & 0 & 1 & -c_z \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}, \tag{3.1}$$

donde R es la matriz de rotación y $C = (c_x, c_y, c_z)$ es el foco. A es la matriz 3×3 de parámetros intrínsecos, $(d_x(x,y), d_y(x,y))$ es el modelo de corrección de la distorsión de la lente y s es un factor de escala que se puede asumir igual a 1.

3.3. Extracción de primitivas de la cancha.

La obtención de los puntos centrales de las primitivas de la cancha se realiza con el algoritmo descrito en el Capítulo 2. El conjunto de puntos de partida en esta fase es el que viene dado por:

$$P = \{p_i = (x_i, y_i) \ \in \ \text{Primitivas de la cancha en la imagen}\}. \tag{3.2}$$

Para extraer líneas de ese conjunto de puntos inicial P, se aplica la transformada de Hough (explicada en la Sección 1.2.3). Las Figuras 3.4, 3.5 y 3.6 muestran la colección de líneas detectadas en varias imágenes usando la transformada de Hough a partir del conjunto de puntos P. En ellas también se observa que algunas de las líneas corresponden a las tangentes de los círculos de la imagen. Aunque, inicialmente, esas tangentes pueden ser consideradas como fallos en la estimación de líneas de la transformada de Hough, la información de dichas tangentes puede usarse para extraer la ecuación de la elipse asociada. Para construir la ecuación de una elipse a partir de las tangentes se usa el trabajo clásico [72] (publicado en 1885). En el que se muestran diferentes métodos de construcción de la ecuación de la elipse utilizando distintos tipos de información. Por supuesto, esta forma es mucho más rápida y robusta que buscar elipses en la imagen usando técnicas estándar basadas solamente en la información de los puntos de las primitivas como la transformada de Hough o RANSAC (véase Sección 1.2.3).

3.4. Función de error: definición y minimización

Se denota por T la colección de primitivas reales. Dada una homografía H (matriz de 3×3) y un punto 2D p, se define $H(p)$ como la transformación de perspectiva inducida por H en el punto p. La función de error $L(H)$ queda de la siguiente forma:

$$L(H) = \frac{1}{\mid P \mid} \sum_{p_i \in P} distancia(H(p_i), T)^2. \tag{3.3}$$

Encontrar el mínimo de la función de error es un problema complicado. El método que se propone para minimizar la función, se basa en construir homografías candidatas usando la colección de líneas que se han extraído de la imagen. Se observa (ver Sección 3.6) que el número de líneas que se maneja es bastante pequeño, por ello el número de configuraciones potenciales que hay que manejar es también pequeño. Dependiendo de la configuración de la escena, se separa el análisis en dos casos:

Caso 1: Hay al menos cuatro líneas visibles en la imagen. En este caso se construyen las homografías candidatas poniendo en correspondencia las cuatro líneas visibles en la escena con cuatro líneas del modelo de la cancha real. Para cada par de líneas puestas en correspondencia entre la cancha de la imagen y la cancha real, se obtiene la siguiente relación:

$$\begin{pmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{pmatrix}^T \begin{pmatrix} a'_i \\ b'_i \\ c'_i \end{pmatrix} = S \begin{pmatrix} a_i \\ b_i \\ c_i \end{pmatrix}. \tag{3.4}$$

De la relación anterior se deducen dos ecuaciones lineales en los coeficientes de H

$$(a'_i h_1 + b'_i h_4 + c'_i h_7)\, b_i = (a'_i h_2 + b'_i h_5 + c'_i h_8)\, a_i, \tag{3.5}$$

$$(a'_i h_1 + b'_i h_4 + c'_i h_7)\, c_i = (a'_i h_3 + b'_i h_6 + c'_i h_9)\, a_i. \tag{3.6}$$

Juntando las ecuaciones lineales anteriores para las cuatro líneas en correspondencia se obtiene el sistema de ecuaciones $Bh = 0$ donde B es una matriz de $8x9$. La solución del sistema se obtiene minimizando $\|Bh\|^2$, y se deduce que la homografía H viene dada por el autovector mínimo de la matriz $B^T B$.

Caso 2: El círculo central y la línea de medio campo son visibles en la imagen. Adicionalmente, se supondrá que también es visible una de las líneas de banda o el punto del centro del campo. Este caso es más complejo de analizar y su solución está basada en el trabajo hecho por Luis Alvarez y Vicent Caselles en [7], donde explican un método para recuperar la homografía a partir de este escenario (ver [7] para mas detalles).

Cuando se empieza el proceso de calibración de la cámara, no se posee información "a priori" sobre la configuración del escenario que se está tratando. Por esto, se prueban ambas configuraciones (caso 1 y caso 2) y para cada configuración se construyen distintas homografías considerando combinaciones potenciales de primitivas. Finalmente, se mantiene la homografía con el valor más bajo de la función de error.

3.5. Reconocimiento de la posición de la cámara.

Se comienza fijando un sistema de coordenadas 3D donde se incluyen las primitivas reales en el plano $z = 0$. Para recuperar la posición de la cámara, se necesita manejar la matriz de parámetros intrínsecos A y la homografía estimada H que satisfacen:

$$A = \begin{pmatrix} f & 0 & x_c \\ 0 & f \cdot r & y_c \\ 0 & 0 & 1 \end{pmatrix}, \qquad H = sAR \begin{pmatrix} 1 & 0 & -c_x \\ 0 & 1 & -c_y \\ 0 & 0 & -c_z \end{pmatrix}. \tag{3.7}$$

Se asume que los parámetros intrínsecos de la cámara son conocidos, excepto la distancia focal f que varía dependiendo de la configuración del parámetro de *zoom*. Para recuperar la distancia focal a partir de H, usando las ecuaciones anteriores se consigue:

$$H^T A^{-T} A^{-1} H = s^2 \begin{pmatrix} 1 & 0 & -c_z \\ 0 & 1 & -c_y \\ -c_z & -c_y & (c_y)^2 + 2\,(c_z)^2 \end{pmatrix}. \tag{3.8}$$

Reemplazando A por su valor, se obtiene:

$$H^T \begin{pmatrix} 1 & 0 & -x_c \\ 0 & \frac{1}{r^2} & -\frac{1}{r^2}y_c \\ -x_c & -\frac{1}{r^2}y_c & \frac{1}{r^2}y_c^2 + x_c^2 + f^2 \end{pmatrix} H = s \begin{pmatrix} 1 & 0 & -c_z \\ 0 & 1 & -c_y \\ -c_z & -c_y & (c_y)^2 + 2\,(c_z)^2 \end{pmatrix}. \quad (3.9)$$

Con $b = (b_1, b_2, b_3, b_4) = (\frac{1}{r^2}, x_c, -\frac{1}{r^2}y_c, -\frac{1}{r^2}y_c^2 + x_c^2 + f^2)$. De la ecuación anterior se obtienen dos ecuaciones con los coeficientes de b:

$$h_{21}\,(b_1h_{21} + b_3h_{31}) + h_{11}\,(b_2h_{31} + h_{11}) + h_{31}\,(b_2h_{11} + b_3h_{21} + b_4h_{31}) - h_{22}\,(b_1h_{22} + b_3h_{32}) + h_{12}\,(b_2h_{32} + h_{12}) + h_{32}\,(b_2h_{12} + b_3h_{22} + b_4h_{32}) = 0, \quad (3.10)$$

$$h_{22}\,(b_1h_{21} + b_3h_{31}) + h_{12}\,(b_2h_{31} + h_{11}) + h_{32}\,(b_2h_{11} + b_3h_{21} + b_4h_{31}) = 0. \quad (3.11)$$

Como b_1, b_2 y b_3 son conocidos, el único desconocido es b_4, por lo tanto, a partir de una sola cámara es posible calcular b_4 y posteriormente la distancia focal f.

Seguidamente, para calcular los parámetros extrínsecos a partir de la homografía y los parámetros intrínsecos, hay que tener en cuenta que de la Ecuación (3.7) se puede llegar a:

$$R = sA^{-1}H \begin{pmatrix} 1 & 0 & -c_x \\ 0 & 1 & -c_y \\ 0 & 0 & -c_z \end{pmatrix} = sA^{-1}H \begin{pmatrix} 1 & 0 & -\frac{c_x}{c_z} \\ 0 & 1 & -\frac{c_y}{c_z} \\ 0 & 0 & -\frac{1}{c_z} \end{pmatrix}. \quad (3.12)$$

Igualando las dos primeras columnas de las matrices de la Ecuación (3.12) se tiene la matriz de rotación (teniendo en cuenta que R es una matriz ortonormal).

3.6. Resultados experimentales

Se han llevado a cabo varios experimentos en donde se ha probado el método propuesto en distintas imágenes provenientes de dos tipos de secuencias, modelo a escala

Imagenes	Vista	Puntos	Líneas	Función de error
Modelo a escala	Área	4146	9	5.592e-004
	Central	2705	9	8.000e-004
Partido real	General	3880	10	2.917e-002
	Central	3174	10	3.709e-002
Regiones sombreadas	Área	641	9	1.551e-002
	Área (esquina)	601	9	2.808e-002

Cuadro 3.1: Resultados cuantitativos para las imágenes en las Figuras 3.4, 3.5 y 3.6: Número de puntos y líneas de las primitivas usadas para la calibración y función de error medida en metros.

de un campo de fútbol (dimensiones de la imagen 1440×809) y escenas reales de partidos de fútbol (de dimensiones 1920×1080). Además, se probó el método en los dos casos posibles de configuración de la vista, cuando al menos hay cuatro líneas visibles y cuando son visibles el círculo central, la línea de medio campo y una de las líneas de banda. Los resultados obtenidos cuando se aplicó el método a una imagen de cada caso se pueden ver en el Cuadro 3.1. La segunda columna contiene el número de puntos que han sido etiquetados como primitiva con el detector morfológico de centros de líneas explicado en [6]. En la tercera columna aparece el número de líneas que han sido extraídas después de aplicar la transformada de Hough. Por último, se muestra el resultado de la función de error (Ecuación 3.3), medida en metros, y se observa que los resultados son muy precisos. En la Figura 3.3 se muestra un ejemplo de los resultados numéricos obtenidos para una imagen en particular.

Por otro lado, con la posición estimada de la cámara se generó una representación 3D de cada cámara. En las Figuras 3.4, 3.5 y 3.6 aparecen varios ejemplos: se presentan algunos pares de imágenes donde una de ellas (la superior) es la imagen real con las líneas que se detectaron (marcadas con diferentes colores), y la otra (la inferior) es una ilustración 3D con la posición estimada de la cámara representada por una imagen en miniatura y cuatro líneas determinando la vista. Con estas imágenes se comprueba que la posición y orientación de la cámara se han calculado correctamente.

3.7. Conclusiones

En este capítulo se ha estudiado el problema de calibración de cámaras en escenarios planos donde normalmente hay un número pequeño de primitivas visibles, las cuales

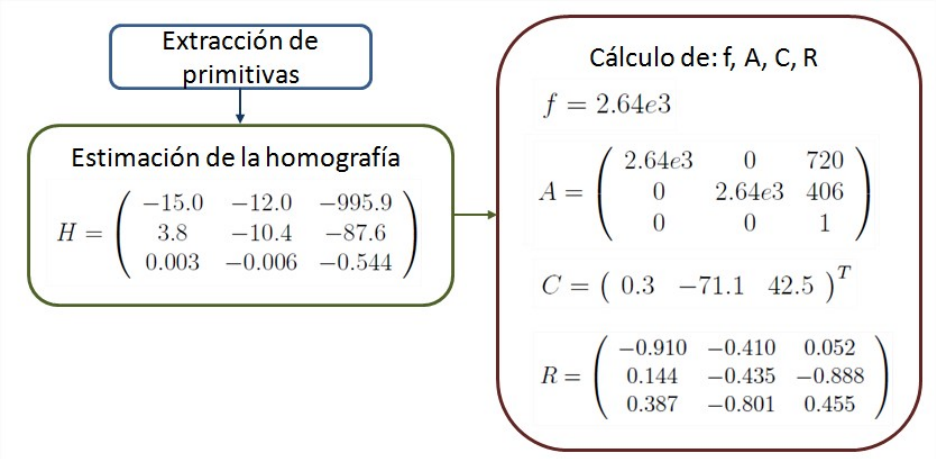

Figura 3.3: Resultados obtenidos en el proceso de calibración.(Imagen superior izquierda en la Figura 3.4)

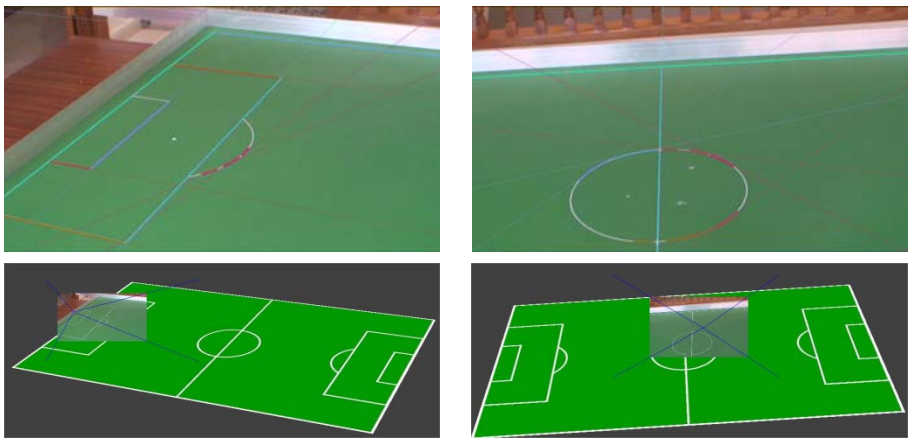

Figura 3.4: Modelo a escala: Área (izquierda) y vista central (derecha). Líneas detectadas en el campo (arriba), y reconstrucción de la posición estimada de la cámara, mostrando su posición y orientación con respecto al campo (abajo).

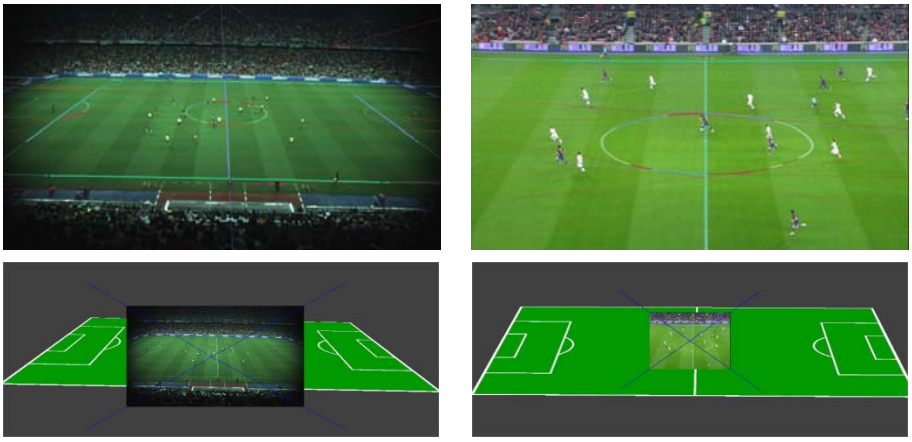

Figura 3.5: Partido real: Vista general (izquierda) y central (derecha). Líneas detectadas en el campo (arriba), y reconstrucción de la posición estimada de la cámara, mostrando su posición y orientación con respecto al campo (abajo).

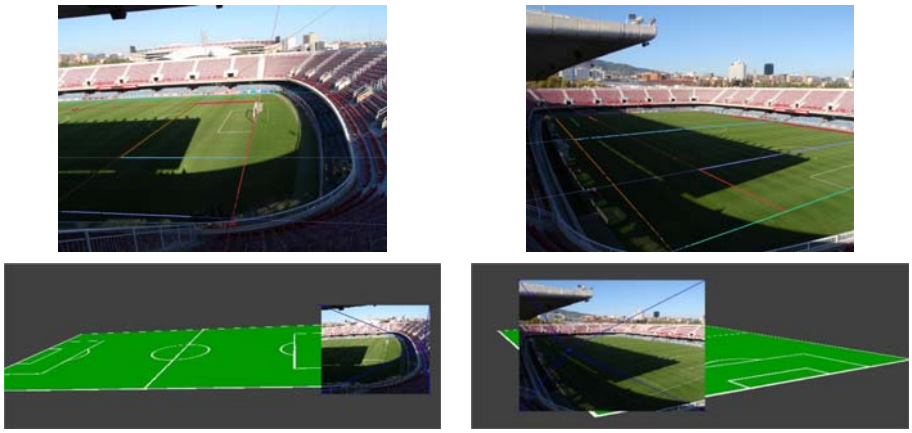

Figura 3.6: Regiones sombreadas: Área (izquierda) y área desde la esquina (derecha). Líneas detectadas en el campo (arriba), y y reconstrucción de la posición estimada de la cámara, mostrando su posición y orientación con respecto al campo (abajo).

pueden ser consideradas para llevar a cabo la estimación de la posición de la cámara. Se demostró que si un número mínimo de primitivas de la cancha son visibles en la escena, con el método propuesto, se puede recuperar la homografía que transforma el plano de la imagen al plano de referencia. El método está basado en la construcción de homografías candidatas usando la colección de líneas que se extrajeron de la imagen y en encontrar la homografía que minimice la función de error. Luego a partir de esta homografía se puede recuperar el *zoom* de la cámara además su posición y orientación. Con los resultados obtenidos en los experimentos, se ha mostrado la robustez del método bajo condiciones difíciles. Como se observó, la calibración estimada es coherente con la vista de las imágenes originales y se probó que la posición y orientación se calculan correctamente.

Capítulo 4

Calibración de cámaras montadas en un trípode

4.1. Introducción

Una práctica muy común en la producción de vídeo es trabajar con cámaras montadas en un trípode. Cuando se procesa vídeo proveniente de estas cámaras, hay que tener en cuenta los cambios en el modelo de cámara, como se explica en [45, 53]. En este capítulo, se centra la atención en escenarios deportivos, los cuales presentan un escenario plano, y en los cuales es muy común que se encuentre la cámara montada en un trípode. Estudiando este escenario se pueden encontrar trabajos que realizan calibración de cámaras en este tipo de situaciones [42, 47, 52, 59]. En todos ellos se sigue una estrategia similar, la cual consiste en unas ciertas tareas que se aplican en cada fotograma del vídeo: extracción de características (por ejemplo primitivas y fondo), estimación del movimiento de la cámara (basada en los fotogramas previos), seguimiento de primitivas, y mejora de la estimación de los parámetros de la cámara. Las cámaras montadas en un trípode tienen una localización fija, pero pueden rotar libremente y cambiar sus parámetros intrínsecos al hacer *zoom*. En los trabajos [2, 68] se describen algunos métodos de calibración de cámaras con rotación y *zoom*. El trabajo [61] introduce optimizaciones de calibración para las cámaras *PTZ (Pan-Tilt-Zoom)*. En [42] se propone un método basado en una versión simplificada del modelo del trípode, donde se asume que el punto principal de la cámara y el centro de rotación del trípode son el mismo punto. En este capítulo, se analiza el problema en escenarios reales, donde hay que lidiar con algunas dificultades adicionales, como el reducido número de primitivas visibles (necesarias para calibrar), o el gran tamaño de los vídeos de alta definición

(provoca que el procesado sea lento). En el método propuesto en este capítulo, la etapa de inicialización consiste en tres pasos, los cuales se desarrollan en el primer fotograma:

1. Leer la información previamente calculada (parámetros geométricos del trípode e información necesaria para la detección de primitivas).
2. Detección de primitivas.
3. Calibración de la cámara para el primer fotograma.

La detección de primitivas en esta etapa de inicialización, se realiza con el método morfológico explicado en el Capítulo 2. El proceso de calibración del primer fotograma se consigue aplicando las técnicas descritas en el Capítulo 3. Para calibrar el resto de fotogramas de la secuencia, se empieza directamente en la fase de la estimación de la calibración y en ella se utiliza la información obtenida en los dos fotogramas anteriores. Se asume que como la cámara está colocada en un trípode, los movimientos de la misma son restringidos y por lo tanto los cambios de rotación y *zoom* entre fotogramas consecutivos no son muy grandes. Después, se continúa el proceso con la fase de seguimiento de primitivas, en la cual se buscan las primitivas en la imagen partiendo de la proyección de las primitivas de referencia. Estas primitivas son líneas y círculos de un modelo de un campo de fútbol con dimensiones reales, que son proyectadas en la imagen usando la homografía estimada a partir de los parámetros de la cámara calculados los dos fotogramas anteriores. Esa parte del proceso se explica en el Capítulo 5. Finalmente, con las primitivas detectadas y la geometría del trípode calculada, se mejora la calibración del fotograma actual. Esto consiste en determinar tres parámetros: *pan, tilt* y *zoom*. Para un fotograma n hay que computar el conjunto (p_n, t_n, z_n). El cálculo de estos parámetros es de la siguiente manera:

1. Inicializar (p_n, t_n, z_n). Se usa un método de interpolación estándar para estimar un valor inicial de (p_n, t_n, z_n) a partir de los fotogramas previos.
2. Seguimiento de las líneas blancas que delimitan el terreno de juego en el fotograma n (se explica en la Sección 5.3). Los puntos centrales de las líneas están asociados, aplicando una regla de proximidad, a las primitivas del campo de referencia en el fotograma n. La suposición que se hace es que la cámara se mueve suavemente y por eso los puntos centrales de las líneas blancas en el fotograma n estarán cerca de la estimación inicial (p_n, t_n, z_n), obtenida de los fotogramas previos.
3. Recalcular los parámetros (p_n, t_n, z_n) usando las nuevas primitivas que se han detectado en el fotograma n. Se minimiza el error *RMS* para las primitivas del campo y sus proyecciones recalculadas con los nuevos parámetros (véase Sección 4.3.2).

4.1.1. Contribución de este capítulo

Modelo matemático para la calibración de cámaras montadas en un trípode:
Se presenta un nuevo modelo matemático para la calibración de secuencias de vídeo cuando la cámara está montada en un trípode. Una de las novedades es que no se supone que el centro de rotación del trípode y el centro de proyección de la cámara sean el mismo punto. La calibración está basada en la geometría del trípode y en el seguimiento de primitivas.

4.2. Geometría y calibración de cámaras montadas en un trípode

Un trípode se define por un centro de rotación $\bar{X}_0 = (X_0, Y_0, Z_0)^T$ y dos ejes de rotación unitarios $\bar{e}^0 = \left(\bar{e}^0_x, \bar{e}^0_y, \bar{e}^0_z\right)^T$, $\bar{e}^1 = \left(\bar{e}^1_x, \bar{e}^1_y, \bar{e}^1_z\right)^T$. La matriz para rotar el ángulo θ_k sobre el eje $\bar{e}^k$ es $R\left(\bar{e}^k, \theta_k\right)$. Para rotar un punto 3D $\bar{X}$ sobre el eje $\bar{e}^k$ usando como centro de rotación $\bar{X}_0$, la transformación se convierte en la siguiente ecuación:

$$\bar{X}\left(\theta_k\right) = \bar{X}_0 + R\left(\bar{e}^k, \theta_k\right)\left(\bar{X} - \bar{X}_0\right). \tag{4.1}$$

El movimiento general de un trípode es la composición de dos rotaciones del tipo explicado anteriormente. Se asume que el centro de rotación $\bar{X}_0$ es el mismo para ambos ejes, lo que es equivalente a asumir que los dos ejes sobre los cuales rota el trípode se cruzan en un punto. Ésta es una situación común y los puntos son transformados de acuerdo con la ecuación general para el movimiento de un trípode:

$$\bar{X}\left(\theta_0, \theta_1\right) = \bar{X}_0 + R\left(\bar{e}^0, \theta_0\right) R\left(\bar{e}^1, \theta_1\right)\left(\bar{X} - \bar{X}_0\right). \tag{4.2}$$

A partir de ahora se usará la siguiente notación:

$$R\left(\theta_0, \theta_1\right) \equiv R\left(\bar{e}^0, \theta_0\right) R\left(\bar{e}^1, \theta_1\right), \tag{4.3}$$

$$\bar{t}\left(\theta_0, \theta_1\right) = \bar{X}_0 - R\left(\theta_0, \theta_1\right)\bar{X}_0. \tag{4.4}$$

Por tanto, la Ecuación (4.2) se puede escribir de la forma:

$$\bar{X}\left(\theta_0, \theta_1\right) = R\left(\theta_0, \theta_1\right)\bar{X} + \bar{t}\left(\theta_0, \theta_1\right). \tag{4.5}$$

La ecuación general para la proyección de un punto 3D $\bar{X} = (X, Y, Z)^T$ en el plano de la imagen es la que sigue:

$$s \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = A(f_0) R_0 \left\lfloor Id, -\bar{c}^0 \right\rfloor \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}, \tag{4.6}$$

donde R_0 es una matriz de rotación y:

$$A(f_0) = \begin{pmatrix} f_0 & 0 & x_c \\ 0 & rf_0 & y_c \\ 0 & 0 & 1 \end{pmatrix}, \qquad \left[Id, -\bar{c}^0\right] = \begin{pmatrix} 1 & 0 & 0 & -\bar{c}_x^0 \\ 0 & 1 & 0 & -\bar{c}_y^0 \\ 0 & 0 & 1 & -\bar{c}_z^0 \end{pmatrix}. \tag{4.7}$$

En la Ecuación (4.6), se asume que la posible distorsión de la lente ha sido corregida previamente. La matriz $P_0 \equiv A(f_0) R_0 \left\lfloor Id, -\bar{c}^0 \right\rfloor$ es denominada matriz de proyección. Para cada fotograma, los valores de $(f_i, \theta_0^i, \theta_1^i)$ determinan la matriz de proyección como sigue:

$$P\left(f_i, \theta_0^i, \theta_1^i\right) = A(f_i) R_0 \left[Id| - \bar{c}^0\right] \begin{pmatrix} R\left(\theta_0^i, \theta_1^i\right) & \bar{t}\left(\theta_0^i, \theta_1^i\right) \\ 0 & 1 \end{pmatrix}. \tag{4.8}$$

Por lo tanto, considerando la siguiente expresión:

$$P_i\left(f_i, \theta_0^i, \theta_1^i\right) = A(f_i) R_0 R\left(\theta_0^i, \theta_1^i\right) \left[Id| R^T\left(\theta_0^i, \theta_1^i\right) \left(\bar{t}\left(\theta_0^i, \theta_1^i\right) - \bar{c}^0\right)\right], \tag{4.9}$$

se puede deducir que la rotación y el foco de la cámara después del movimiento son:

$$R_i = R_0 R\left(\theta_0^i, \theta_1^i\right), \tag{4.10}$$

$$\bar{c}^i = -R^T\left(\theta_0^i, \theta_1^i\right) \left(\bar{t}\left(\theta_0^i, \theta_1^i\right) - \bar{c}^0\right). \tag{4.11}$$

Una de las principales novedades de esta propuesta es el hecho de que no se supone que el centro de rotación del trípode y el centro de proyección de la cámara sean el mismo punto (lo cual es una simplificación usual del modelo). Hay que tener en cuenta que cualquier vista adquirida con el trípode puede considerarse como referencia para moverlo, y cuando se cambian los parámetros iniciales de la cámara, también se modifican los ejes de rotación del trípode.

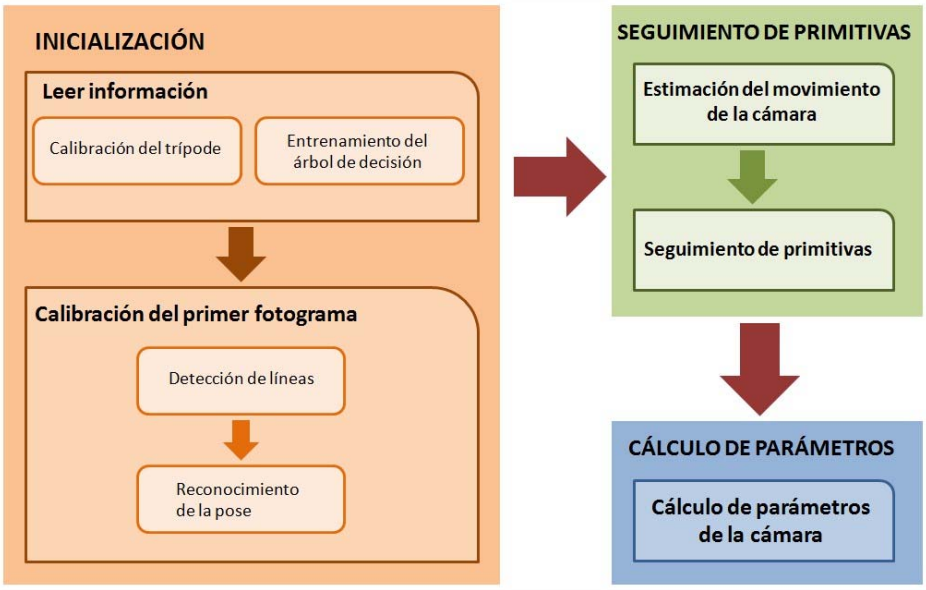

Figura 4.1: Etapas del proceso de calibración de cámaras montadas en un trípode

4.3. Calibración de una secuencia de vídeo tomada con una cámara montada en un trípode

Este procedimiento está dividido en tres etapas principales: inicialización, seguimiento de primitivas y cálculo de los parámetros que definen el movimiento de la cámara (véase Figura 4.1). En la primera fase, se obtienen datos de la secuencia de vídeo tales como la geometría del trípode (centro de rotación y la orientación inicial de los ejes de rotación), clasificación de píxeles (necesaria para el seguimiento de primitivas, Sección 5.2) y la calibración de la cámara en el primer fotograma de la secuencia (Sección 4.3.1). Después de la inicialización, se prosigue con la calibración de la cámara en la secuencia completa con un proceso incremental. Para llevar a cabo esta etapa, se sigue el movimiento de la cámara en cada fotograma, usando la información obtenida en la calibración de los fotogramas anteriores. Normalmente, el proceso comienza en el primer fotograma, también se puede elegir empezar desde el último fotograma y recorrer la secuencia de manera inversa, pero la mejor manera sería empezar desde el fotograma en el que se observen un mayor número de primitivas.

Figura 4.2: Geometría de la calibración del trípode para tres fotogramas de referencia. Para validar los resultados, las primitivas del campo de fútbol han sido proyectadas en las imágenes reales usando diferentes colores.

4.3.1. Calibración del trípode y del primer fotograma

En la práctica, para estimar la geometría del trípode, previamente se han calibrado algunos fotogramas aislados de la secuencia de vídeo. Estos fotogramas se calibran usando la técnica descrita en el Capítulo 3, y después se estima la geometría del trípode usando una técnica estándar de *bundle adjustment*. El último paso en la fase de inicialización es la calibración del primer fotograma, esto significa, obtener los parámetros intrínsecos y extrínsecos de la cámara. Este primer fotograma se calibra usando también la misma técnica usada para calibrar los fotogramas de la calibración del trípode. Dentro de este proceso, se incluye un paso en el cual se calcula el modelo de distorsión de lentes en el primer fotograma. Este modelo se usa en la secuencia completa, pero solamente se calcula en el primer fotograma con el método descrito en [10].

4.3.2. Cálculo de los parámetros del movimiento de la cámara

En esta fase es donde se calculan los parámetros *pan*, *tilt* y *zoom* para el fotograma actual. Primero, se utilizan los parámetros obtenidos en los fotogramas anteriores para inicializar los parámetros (p_n, t_n, z_n) por medio de un procedimiento básico de extrapolación. Seguidamente se optimiza (p_n, t_n, z_n) minimizando el error entre la proyección de los puntos centrales de las primitivas obtenidas y las primitivas de referencia, es decir, se minimiza la función de error:

$$L(H) = \frac{1}{\mid P \mid} \sum_{p_i \in P} distancia(H(p_i), T)^2, \tag{4.12}$$

donde T es la colección de las primitivas detectadas. H es la homografía (matriz de 3×3, obtenida de *pan,tilt* y *zoom*) y p es un punto 2D. Se define $H(p)$ como la

transformación perspectiva inducida por H en el punto p. La función *distancia* es la distancia Euclídea entre las primitivas y sus respectivas transformaciones. Encontrar el mínimo de la Ecuación 4.12 es un problema complicado. El método que se propone, está basado en la mejora de la homografía a partir de los parámetros *pan,tilt* y *zoom*) estimados. Finalmente se aplica el algoritmo de Levenber-Marquart. Como se parte de una aproximación de la homografía, se pueden obtener los parámetros de la cámara en situaciones donde los procedimientos de calibración estándar fallarían. Por ejemplo en imágenes sin suficientes primitivas visibles para realizar la correspondencia con las líneas y círculos de referencia.

4.4. Experimentos

4.4.1. Configuración de los experimentos

Para calibrar una secuencia de vídeo se necesita cierta información antes de comenzar, como se explica en las Secciones 4.3.1 y 5.2. Anteriormente se ha descrito que la calibración del trípode se calcula con algunos fotogramas que han sido extraídos de diferentes instantes de la secuencia. Dichos fotogramas se calibran usando la técnica descrita en el Capítulo 3, obteniendo así los parámetros intrínsecos y extrínsecos. Se pueden ver los resultados de la calibración de estos fotogramas en la Figura 4.2. Una vez finalizada la etapa de inicialización, se procede a calibrar la secuencia completa.

4.4.2. Resultados

Para mostrar el funcionamiento del método propuesto, se ha calibrado la cámara en dos tipos de secuencias diferentes. En las Figuras 4.3 y 4.4 se pueden observar varios resultados de la calibración en fotogramas extraídos de cada secuencia. Se muestran las proyecciones del campo de referencia en las imágenes usando las homografías calculadas en cada uno de los fotogramas.

Los experimentos se ejecutaron en un ordenador personal con un procesador Intel Core i7 2.00Ghz y 4GB de memoria RAM. Los tiempos de procesamiento obtenidos al calibrar secuencias de vídeo con el método propuesto se expresan en el Cuadro 4.1. La computación del seguimiento del movimiento de la cámara para un fotograma tarda 3 milisegundos en la secuencia del modelo a escala y 5 milisegundos en la secuencia del partido real (1920 x 1080). También se observa que el método es mas rápido que el

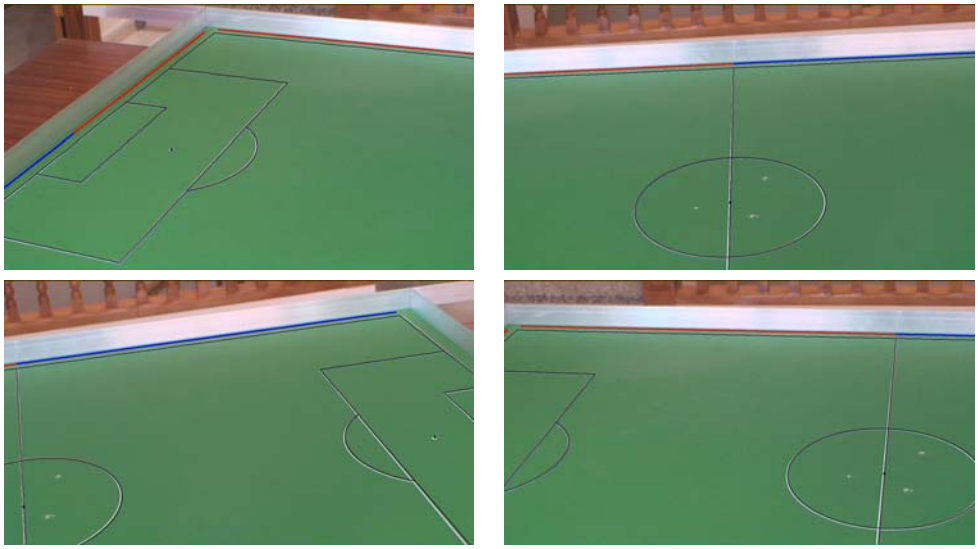

Figura 4.3: Resultados de la calibración de la cámara. Imágenes del modelo a escala. Se muestran las imágenes donde se ha proyectado el campo de fútbol de referencia usando los parámetros de la cámara calculados.

Figura 4.4: Resultados de la calibración de la cámara. Imágenes de partidos de fútbol reales: A la izquierda seguimiento de líneas. A la derecha se muestran las imágenes donde se ha proyectado el campo de fútbol de referencia usando los parámetros de la cámara calculados.

Secuencias	Modelo a escala	Fútbol real
seguimiento primitivas	1 ms	2 ms
cálculo parámetros cámara	2 ms	3 ms
seguimiento movimiento cámara (total)	3 ms	5 ms

Cuadro 4.1: Media de tiempos de procesamiento para ambas secuencias en milisegundos por fotograma. La secuencia del modelo a escala contiene 823 fotogramas de tamaño 1440×812 píxeles. La secuencia de fútbol real está compuesta por 385 imágenes de dimensiones 1920×1080.

Secuencia	Error
Modelo a escala	0.01610
Fútbol real	0.01468

Cuadro 4.2: Error cuadrático medio (en metros) entre la proyección de los puntos de las primitivas de referencia y de los puntos de las primitivas extraídas con el seguimiento de primitivas.

propuesto en [42], que requiere 5.8 milisegundos para procesar una imagen más pequeña (720 x 576). El tiempo total es el resultado de la suma de los tiempos de las etapas de seguimiento de primitivas (Sección 5.3) y cálculo de los parámetros de la cámara (Sección 4.3.2). Se demuestra que el método puede trabajar en tiempo real. En estos experimentos no se ha tenido en cuenta el tiempo de lectura de las imágenes desde el disco duro, ya que puede variar mucho dependiendo de la arquitectura del sistema.

En algunos fotogramas, se pueden encontrar imágenes con sólo unas pocas primitivas visibles. Por ejemplo en la secuencia del modelo a escala hay 51 fotogramas con sólo 3 primitivas visibles. En ese caso, las técnicas estándar para recuperar los parámetros de movimiento de la cámara fallan debido a que requieren información más robusta, por ejemplo cuatro líneas visibles o cuatro intersecciones, etc.

Como en este caso se trabaja con cámaras montadas en un trípode, las restricciones de la geometría del trípode simplifican enormemente el problema y permiten usar el seguimiento de primitivas para recuperar con precisión los parámetros del movimiento de la cámara en estos casos. Este último aspecto es otro punto importante, y los resultados de los experimentos demuestran que los parámetros de la cámara obtenidos por el método propuesto son precisos. El Cuadro 4.2 muestra que el error numérico de los resultados es bastante pequeño para este tipo de escenarios. Se ha medido la precisión con el error cuadrático medio entre la proyección de los puntos de las primitivas de referencia y de los puntos de las primitivas extraídas con el seguimiento de primitivas.

4.5. Conclusiones

Este capítulo muestra el estudio del problema de la calibración de cámaras de vídeo en escenarios donde, la cámara está montada en un trípode (lo cual es una situación común en la práctica) y en cada fotograma, hay normalmente pocas primitivas visibles para ser utilizadas en la calibración. Para resolver el problema, primero se estudia la geometría del trípode desde un punto de vista matemático. Esta suposición simplifica enormemente el problema de calibración y permite recuperar la calibración del fotograma en situaciones donde las técnicas generales de calibración fallan. En el modelo propuesto, una de las principales novedades es el hecho de que no se supone que el centro de rotación del trípode y el centro de proyección de la cámara sean el mismo punto. También se demuestra que el método puede trabajar en tiempo real y que el error numérico de los resultados es bastante pequeño en los escenarios reales.

Capítulo 5

Seguimiento de primitivas en una secuencia vídeo

5.1. Introducción

Este capítulo se centra en la etapa de seguimiento de primitivas en una secuencia de vídeo. Esta fase es una parte muy importante del proceso de calibración de secuencias de vídeo y además es un procedimiento que puede consumir gran cantidad del tiempo de ejecución. El método de calibración de cámaras propuesto en el Capítulo 4 es un proceso incremental donde la calibración de cada fotograma se calcula con la información obtenida de los dos fotogramas previos. Las primitivas son las líneas y círculos de un patrón con medidas reales, las cuales se proyectan usando la homografía estimada a partir de los dos fotogramas anteriores. Esto significa que en la etapa de seguimiento, se buscan las primitivas en la imagen usando como punto de partida la proyección estimada de las primitivas de referencia. Luego se usa un árbol de decisión para determinar la posición de las líneas y círculos en la imagen. Como siempre se empieza a buscar desde un punto cercano a la posición final y siempre dentro de la región de interés, solo es necesario diferenciar dos clases con el árbol de decisión, primitivas y fondo. Además, en vez de procesar todos los píxeles de la imagen como se hace en el Capítulo 2, solamente hay que clasificar los píxeles que se encuentran en una vecindad de la localización anterior de la primitiva. Actualmente, el uso árboles de decisión está ampliamente extendido en segmentación y clasificación de imágenes porque generan reglas fáciles de entender, y se entrenan y aplican rápidamente, véase por ejemplo [60] y [66]. Por otro lado, los árboles de decisión pueden manejar diferentes características para clasificar los píxeles. Esta clasificación se usa en imágenes de diferentes áreas, como

por ejemplo imágenes médicas [29], o imágenes de satélite [65, 91]. Normalmente se trabaja con los canales *RGB (Red-Green-Blue)* combinados con otros espacios de color, como *HSV(Hue-Saturation-Value)* o bandas diferentes en las imágenes de satélite, por ejemplo el infrarrojo. Para realizar el seguimiento de primitivas, se han propuesto previamente varias soluciones como en [52], un procedimiento de seguimiento de líneas guiado por los parámetros de la cámara. Luego se usa una regla de proximidad para hacer corresponder las líneas detectadas con las estimadas. Por otra parte, en [47] se aplica una técnica de correlación. Otro método se describe en [79], donde se definen unos rangos en donde estarán los parámetros de la cámara y se hace una búsqueda de la correspondencia óptima. La aproximación presentada en [37] usa el espacio de color *YCbCr* para detectar líneas blancas en la imagen. Después, los parámetros para cada línea detectada se refinan minimizando la distancia a la línea del campo más cercana. La técnica descrita en [42] integra el cálculo de la transformada de Hough. El método que se propone en este trabajo usa un *CART* (árbol de decisión) para clasificar los píxeles como primitiva o fondo basándose en un proceso de aprendizaje en los canales *RGB*. En el libro [15] se encuentra una introducción general a *CART*.

5.1.1. Contribución de este capítulo

Aplicación de *CART* al seguimiento de primitivas en secuencias de vídeo: En este trabajo se presenta un nuevo método para el seguimiento de primitivas en secuencias de vídeo basado en *CART (Classification and Regression Tree)*. El procedimiento de seguimiento usa líneas y círculos como primitivas. Se estiman los parámetros del *CART* usando un proceso de aprendizaje basado en los canales *RGB* de la imagen. La calidad del seguimiento de las primitivas con el árbol de decisión se valida por medio de los porcentajes de error obtenidos al clasificar imágenes y la comparación con otras técnicas. Se presenta también cómo se incluye este método en el proceso de calibración de cámaras y cómo acelera la ejecución del mismo.

5.2. Construcción y entrenamiento del árbol de decisión para clasificar primitivas

Para detectar las primitivas blancas en los fotogramas del vídeo, en este trabajo se usa un *CART (Classification And Regression Tree)*, descrito en [15]. Para discriminar las diferentes clases de píxeles, se considera un conjunto de características, pudiendo ser los valores de los píxeles en los canales *RGB* o *HSV*. En el proceso de entrenamiento del

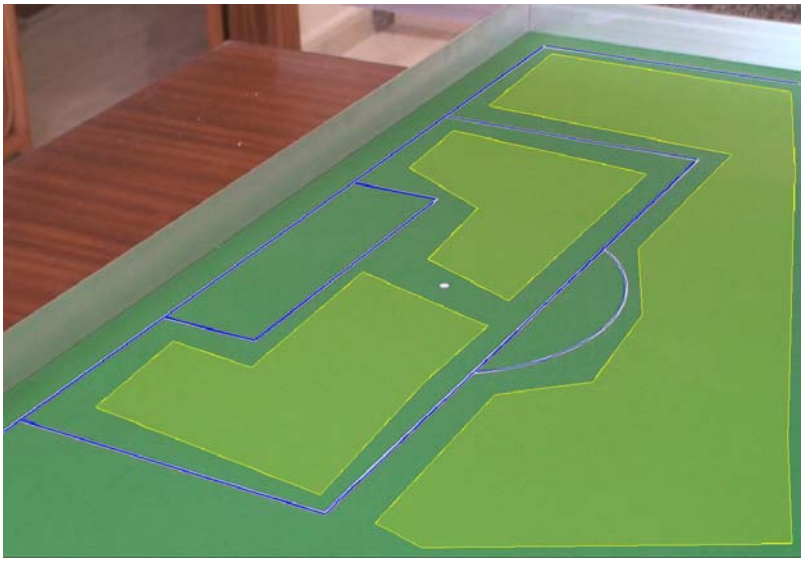

Figura 5.1: Imagen modelo a escala. Se usan dos clases diferentes en la segmentación: primitivas blancas y césped. El césped se segmenta usando polígonos, mientras que para las primitivas se usan segmentos.

árbol de decisión, es importante determinar la manera de seleccionar los canales y los umbrales para obtener un árbol simple. Para construir un árbol de decisión es necesario una etapa de aprendizaje basada en un conjunto de entrenamiento con información sobre las distintas clases, véase por ejemplo [26]. Por ello, para cada secuencia de vídeo se tiene un conjunto de datos de clasificación, que poseen información sobre dos clases, primitivas y fondo. En este trabajo, los experimentos se realizarán con secuencias de partidos de fútbol, en donde por regla general, las primitivas son blancas y el fondo es verde. En la práctica, el conjunto de datos de entrenamiento para el árbol de decisión, se obtiene segmentando algún fotograma de la secuencia, usando solamente dos clases diferentes: primitivas (líneas blancas y círculos) y césped. Se pueden ver ejemplos de segmentación en la Figuras 5.1 y 5.2.

En el conjunto de datos de entrenamiento, hay valores RGB obtenidos a través de la segmentación. Estas tripletas RGB se usan para construir un árbol de decisión de tres canales. Esto determina, en cada nodo, qué canal proporciona la mejor discriminación. Con este propósito, se usa una medida para estimar la impureza de los conjuntos basada en el índice de Gini. Se eligió esta medida por su facilidad de implementación, véase [44]:

$$\sum_{K \neq K'} P_K P_{K'} = \sum_{K=1}^{N} P_K \left(1 - P_K\right), \tag{5.1}$$

Figura 5.2: Imagen campo real. Se usan dos clases diferentes para segmentar: primitivas blancas y césped. El césped se segmenta usando polígonos, mientras que para las primitivas se usan segmentos.

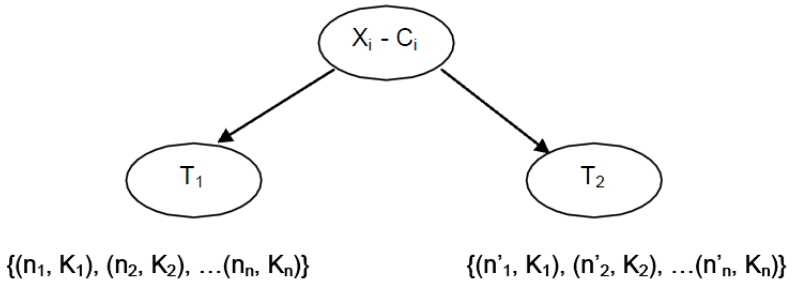

Figura 5.3: Árbol de decisión binario. Los puntos n_i en el nodo T_1 tienen un valor i en el canal X menor que el umbral C_i. Los puntos n_i' en el nodo T_2 tienen $X_i > C_i$.

donde P_K es la probabilidad de un punto de pertenecer a una clase. En el proceso de aprendizaje del árbol de decisión, para decidir el canal y el umbral que se seleccionan en cada nodo, se tiene que minimizar la medida de impureza resultante dividiéndola por el conjunto de puntos (ver imagen 5.3). El objetivo es encontrar los valores X_i y C_i que minimicen la energía compuesta en la Ecuación (5.2), la cual es la suma de las energías de los dos nodos hijo:

$$E_t = E_0\left(X_i, C_i\right) + E_1\left(X_i, C_i\right), \tag{5.2}$$

$$E\left(X_i, C_i\right) = \sum_{i=1}^{n} \frac{n_i}{n_1 + n_2 + ...n_n}\left(1 - \frac{n_i}{n_1 + n_2 + ...n_n}\right). \tag{5.3}$$

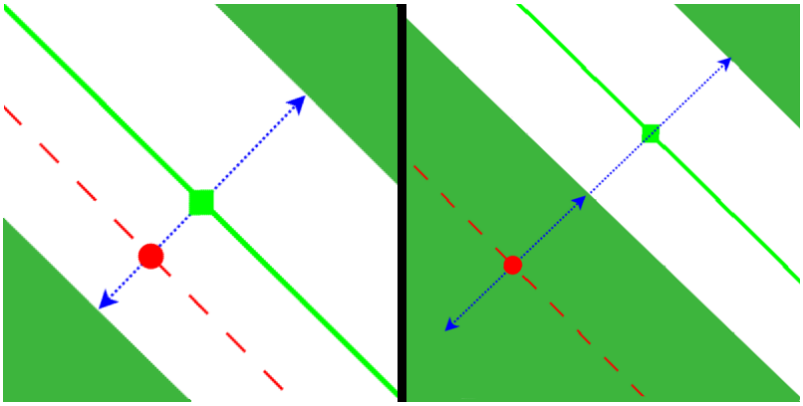

Figura 5.4: Búsqueda de los bordes de las primitivas en el procedimiento de seguimiento. El punto rojo es la inicialización basada en los fotogramas anteriores. La línea azul de puntos es la línea ortogonal que se recorre para buscar los bordes. El cuadrado verde es el centro de la primitiva.

5.3. Seguimiento de primitivas usando el árbol de decisión

Para cada fotograma, primero, se inicializa la situación de las primitivas. Esta inicialización es una estimación de los posibles cambios en los valores *pan*, *tilt* y *zoom*, teniendo en cuenta los dos fotogramas anteriores. Con los parámetros estimados, se calcula la homografía. Después, se proyectan todos los puntos pertenecientes a las primitivas de referencia en el fotograma usando la homografía calculada. Al proyectar los puntos, se realiza una evaluación inversa de la distorsión, aplicando el modelo de distorsión estimado en la fase inicial. Esto se hace para introducir distorsión a los puntos de referencia y que la proyección se ajuste mejor a las primitivas de la imagen. Después de proyectar los puntos de las primitivas, se pueden dar dos situaciones diferentes: que el punto haya sido proyectado dentro de una primitiva blanca o en el fondo. Si el árbol de decisión clasifica el punto proyectado dentro de la clase de primitivas, se deben buscar los bordes de la primitiva en ambas direcciones de la orientación perpendicular (ver Figura 5.4 izquierda). Una vez que se encuentren los bordes, se calcula el punto medio entre los dos, dicho punto será considerado como el centro de la primitiva. Por otra parte, si el punto proyectado se clasifica como perteneciente a la clase del fondo, también hay que buscar los bordes de la primitiva en las dos direcciones de la orientación perpendicular. Sin embargo, en este caso, cuando se encuentre un píxel de una primitiva, solamente se continuará buscando en esa dirección tratando de encontrar el otro borde. Cuando se alcanza el segundo borde, se calcula el punto medio (véase Figura 5.4 derecha).

Figura 5.5: Seguimiento de primitivas. Secuencia de fútbol real. Los puntos negros representan todos los píxeles procesados. Los puntos coloreados son los que se han seleccionado como centros de primitivas.

En ambas situaciones, para evitar considerar grandes zonas blancas y clasificarlas como primitivas, tales como anuncios o jugadores vestidos de blanco, hay que controlar el grosor de las primitivas con un umbral. Este umbral se obtiene dinámicamente, dado que las primitivas más alejadas, aparecen más delgadas en la imagen que las más cercanas a la cámara. El umbral se calcula como la distancia entre las proyecciones de un punto de referencia y otro punto obtenido al sumarle el grosor real de una primitiva a dicho punto de referencia. Si se han examinado mas píxeles que el ancho máximo, la detección de primitivas es desechada en ese punto. Haciendo uso de este umbral se evita considerar los píxeles blancos de la ropa de los jugadores, como se observa en la Figura 5.5.

En vez de procesar todos los píxeles de la imagen, como se hace en el método descrito en el Capítulo 2, con este método de seguimiento solamente hay que clasificar píxeles en una vecindad de la situación previa de las primitivas. Esta estrategia permite reducir la cantidad de píxeles que se procesan (véase Figuras 5.6, 5.7). En consecuencia, se mejora enormemente el tiempo de procesado en comparación con la alternativa de procesar todos los píxeles de la imagen. Esta característica, sumada a la rápida clasificación de los píxeles que proporciona el árbol de decisión, hacen posible alcanzar el procesado en tiempo real.

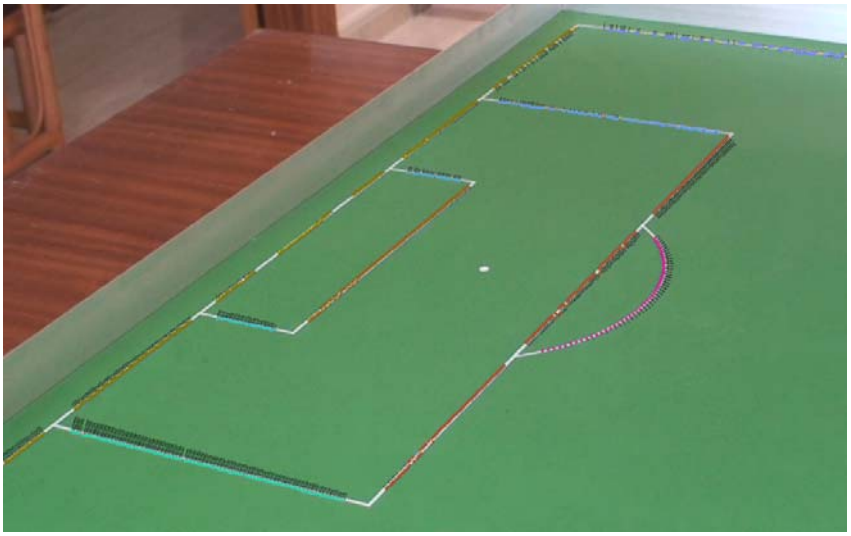

Figura 5.6: Seguimiento de primitivas. Secuencia del modelo a escala. Los puntos negros representan todos los píxeles procesados. Los puntos coloreados son los que se han seleccionado como centros de primitivas.

Figura 5.7: Seguimiento de primitivas. Secuencia de partido de fútbol real. Los puntos negros representan todos los píxeles procesados. Los puntos coloreados son los que se han seleccionado como centros de primitivas

5.4. Resultados experimentales: Clasificación con árbol de decisión

Para alcanzar los mejores resultados en la clasificación de los píxeles, se probaron diferentes configuraciones de *CART* variando el número de canales usados para discriminar las clases. Estas pruebas consisten en construir un árbol de decisión con un conjunto de datos de entrenamiento, y en cada experimento variar los datos que se usan, por ejemplo, usar sólo el canal rojo, todos los canales *RGB* o una combinación de los espacios *RGB* y *HSV*. Después del procedimiento de aprendizaje, en cada experimento, se clasifican con el árbol de decisión los píxeles de cuatro fotogramas aleatorios de cada vídeo y se cuentan los píxeles mal clasificados. Al final, se comparan las clasificaciones de los árboles de decisión con clasificaciones manuales de los mismos fotogramas. Los fotogramas se seleccionan para mostrar diferentes partes del campo de fútbol, por ejemplo, el área, el centro del campo o el campo casi completo. Los resultados están en los Cuadros 5.1 y 5.2 , donde la primera fila contiene los resultados de clasificación del conjunto de datos de entrenamiento y las otras filas son los resultados para los fotogramas escogidos aleatoriamente de las secuencias de vídeo. También se pueden ver estos resultados en forma de gráfica en las Figuras 5.8 y 5.9. Se observó que los mejores resultados los ofrece la configuración de tres canales *RGB*, ya que tiene el porcentaje de error más bajo.

La clasificación realizada con el árbol de decisión se muestra en las Figuras 5.10 y 5.11. Hay que destacar que la discriminación en césped/primitiva se centra solamente en una vecindad de la inicialización de la primitiva, y no importa el resultado que ofrece el árbol de decisión para otras áreas de la imagen (como las gradas). Se han comparado los resultados de clasificación obtenidos con el árbol de decisión *RGB* y los resultados de clasificación del método morfológico propuesto en el Capítulo 2 para detectar las primitivas. Se probaron ambos con cuatro imágenes segmentadas manualmente provenientes de dos vídeos diferentes, modelo a escala y partidos de fútbol reales. En el Cuadro 5.3 se muestran los porcentajes de error, y en la Figura 5.12 se ve la comparación entre los dos métodos. En http://www.ctim.es/demo105/ se ofrecen algunas secuencias completas usadas en los experimentos y clasificadas con el *CART*.

R	G	B	H	S	V	RGB	HSV
0.068	0.268	0.153	1.112	0.060	0.268	0.005	0.009
0.067	0.128	0.100	0.563	0.055	0.128	0.049	0.051
0.087	0.119	0.101	0.399	0.065	0.119	0.066	0.072
0.069	0.110	0.067	0.270	0.076	0.110	0.059	0.046
0.049	0.049	0.076	0.475	0.035	0.091	0.040	0.047

Cuadro 5.1: Porcentajes de error de las configuraciones del árbol de decisión, imágenes del modelo a escala.

R	G	B	H	S	V	RGB	HSV
0.200	0.344	0.194	0.398	0.322	0.344	0.067	0.070
0.174	0.246	0.157	0.297	0.230	0.246	0.121	0.124
0.231	0.358	0.201	0.414	0.322	0.358	0.102	0.105
0.215	0.268	0.208	0.396	0.315	0.268	0.163	0.166
0.146	0.292	0.154	0.263	0.206	0.292	0.124	0.134

Cuadro 5.2: Porcentajes de error de las configuraciones del árbol de decisión, imágenes reales.

RGBs	Ms	RGBr	Mr
0.049	0.072	0.121	0.198
0.066	0.253	0.102	0.312
0.059	0.259	0.163	0.328
0.040	0.084	0.124	0.185

Cuadro 5.3: Comparación de los porcentajes de error entre el método del árbol de decisión y el método morfológico. RGBs y RGBr son los resultados obtenidos con CART en imágenes del modelo a escala e imágenes reales. Ms y Mr representan los resultados obtenidos con el método morfológico.

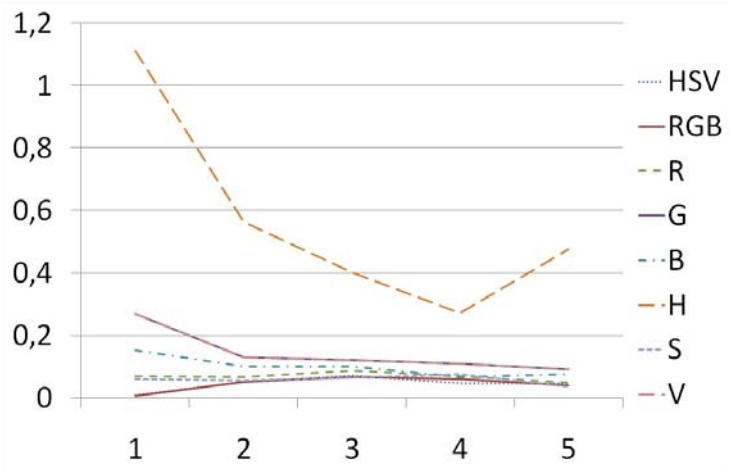

Figura 5.8: Porcentajes de error de las configuraciones del árbol de decisión, imágenes del modelo a escala. Porcentajes del Cuadro 5.1

5.5. Resultados experimentales: Seguimiento de primitivas

Para ilustrar la calidad de los resultados, se usaron las primitivas resultantes de aplicar este método para realizar la calibración de varias secuencias de vídeo. Algunos fotogramas de estas secuencias pueden observarse en las Figuras 5.13 y 5.14.

Un aspecto interesante del método propuesto es su tiempo de ejecución. El método morfológico utilizado en el Capítulo 2, emplea una media de 2725 milisegundos en procesar un fotograma de alta definición en un procesador de cuatro núcleos. El método propuesto en este trabajo usando un árbol de decisión emplea 7 milisegundos por fotograma de alta definición con el mismo procesador. Se pueden ver más resultados que demuestran la rapidez del método detectando los centros de las líneas, en el Cuadro 5.4. En términos de complejidad computacional, la característica principal, es que la computación del árbol de decisión es muy rápida y el método es local. Solamente se necesita procesar una vecindad de la primitiva, mientras que la operación morfológica tarda mucho más tiempo porque el procedimiento procesa toda la imagen.

En los resultados que se muestran en el Cuadro 5.4, los tiempos correspondientes a la implementación paralela, se obtuvieron con una implementación multihilo simple del método. La paralelización consiste en asignar el procesamiento de cada primitiva a un hilo diferente. Esta implementación se desarrolló de manera muy simple usando

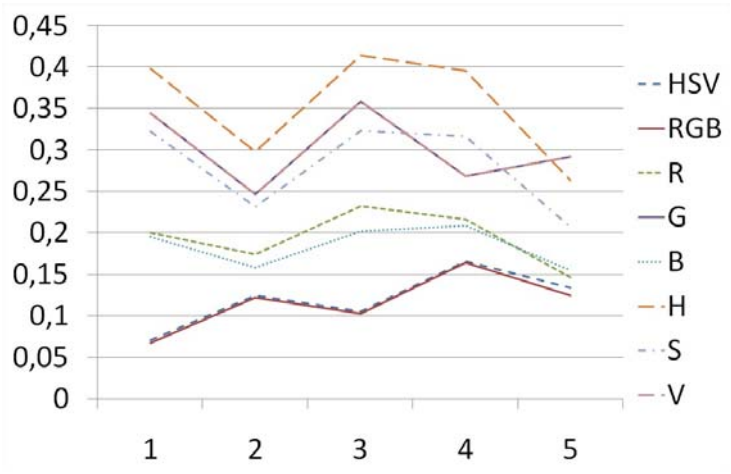

Figura 5.9: Porcentajes de error de las configuraciones del árbol de decisión, imágenes reales. Porcentajes del Cuadro 5.2

Figura 5.10: Imagen del modelo a escala segmentada con el método *CART*. Los puntos clasificados como primitiva se han destacado en amarillo. Los vídeos completos se pueden encontrar en http://www.ctim.es/demo105/

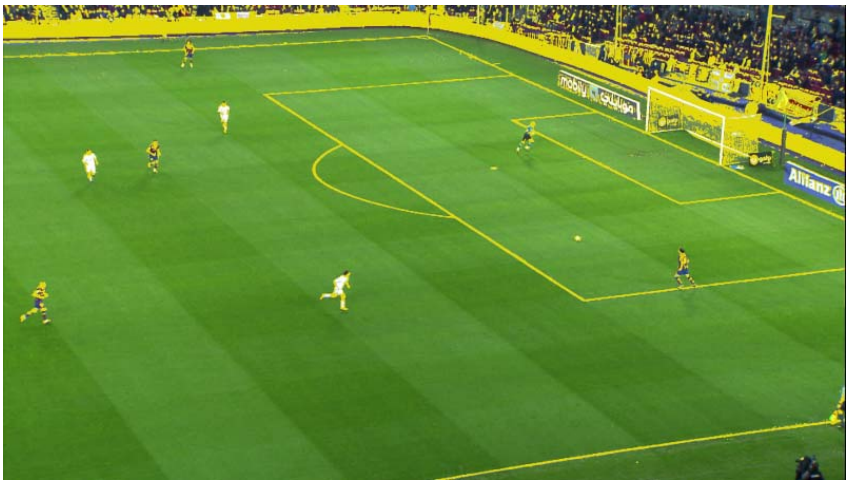

Figura 5.11: Imagen real segmentada con el método *CART*. Los puntos clasificados como primitiva se han destacado de amarillo. Los vídeos completos se pueden encontrar en http://www.ctim.es/demo105/

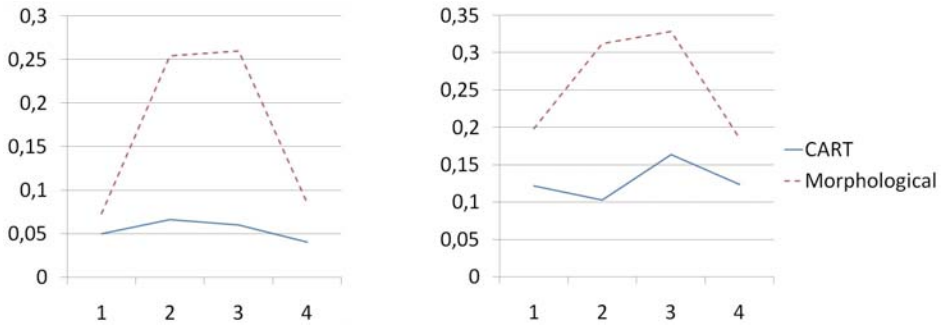

Figura 5.12: Comparación de los porcentajes de error entre el método del árbol de decisión y el morfológico en el modelo a escala(izquierda) e imágenes reales(derecha). Estos datos son del Cuadro 5.3

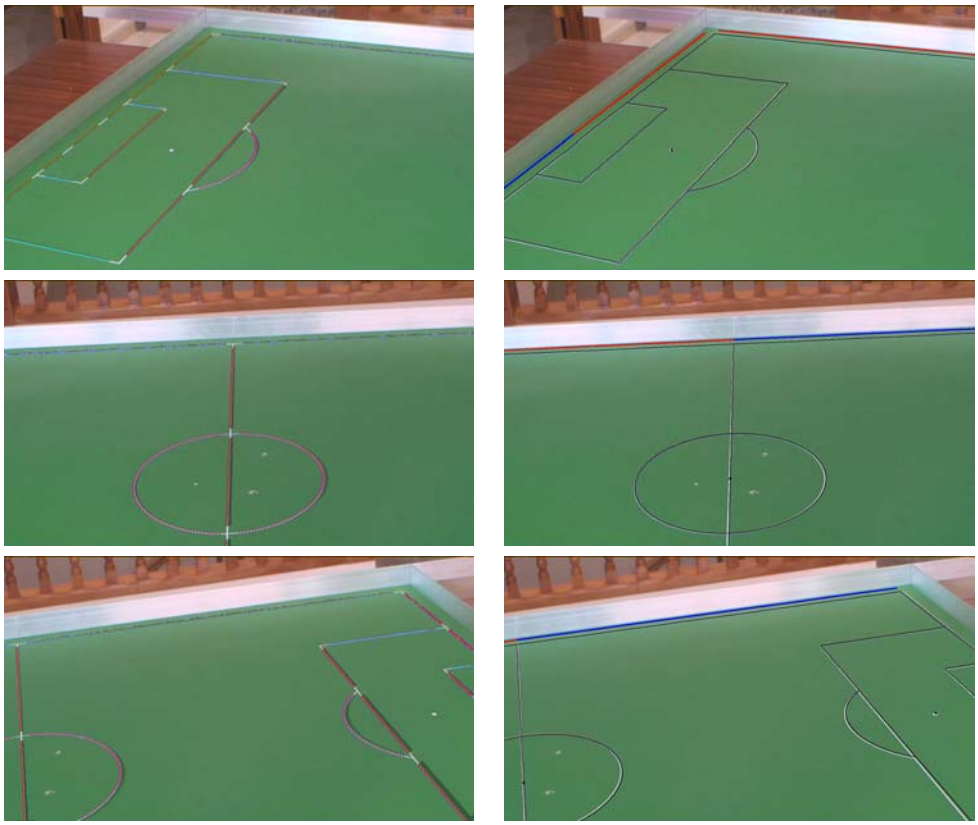

Figura 5.13: Resultados de la calibración de la cámara. Imágenes del modelo a escala: Seguimiento de primitivas (izquierda). Imágenes donde se ha proyectado el campo de fútbol de referencia usando los parámetros de la cámara calculados a partir de las primitivas (derecha).

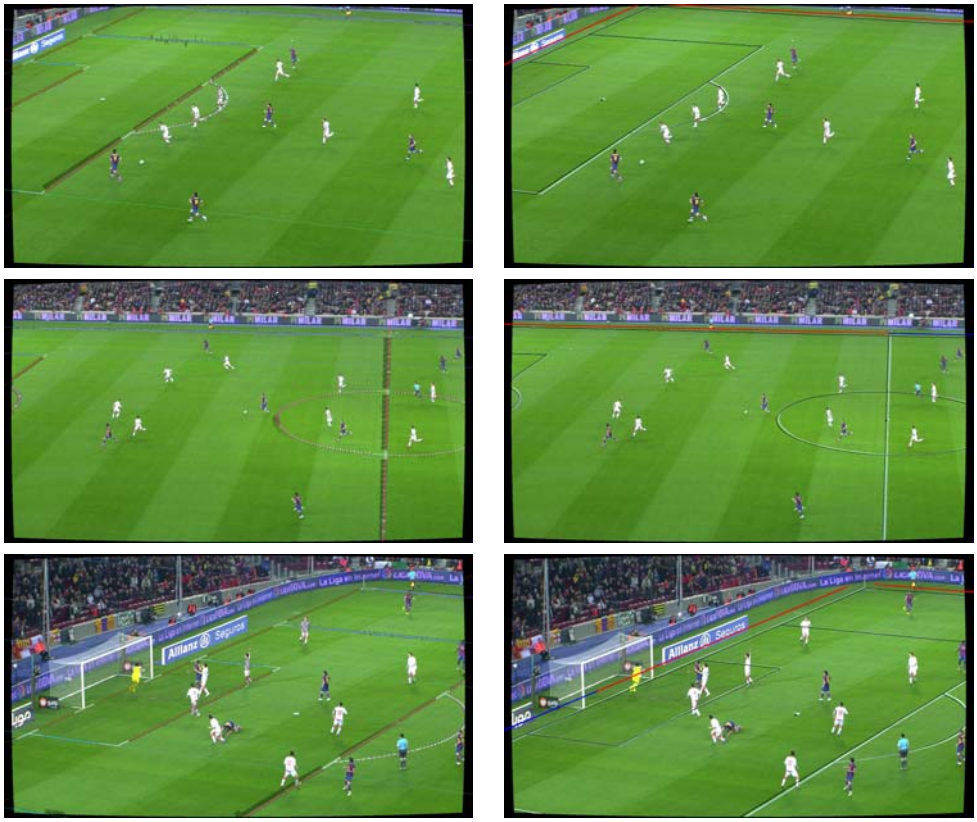

Figura 5.14: Resultados de la calibración de la cámara. Imágenes de partidos de fútbol reales: Seguimiento de primitivas (izquierda). Imágenes donde se ha proyectado el campo de fútbol de referencia usando los parámetros de la cámara calculados a partir de las primitivas (derecha).

Secuencias	Modelo a escala	Fútbol real
Implementación secuencial	2 ms	5 ms
Implementación paralela	1 ms	2 ms

Cuadro 5.4: Tiempos de ejecución para las diferentes implementaciones del seguimiento de primitivas en milisegundos por fotograma. Los tiempos son la media del tiempo de procesado de todos los fotogramas de cada secuencia. La secuencia del modelo a escala contiene 823 fotogramas de tamaño 1440 x 812 píxeles. La secuencia de fútbol real está compuesta por 385 imágenes de dimensiones 1920 x 1080.

OpenMP. Concretamente, sólo se usa la sentencia *parallel for*, la cual asigna a cada hilo una iteración del bucle, véase [21]. Sin embargo, como se observa en el Cuadro 5.4, la implementación secuencial del método es también capaz de procesar los fotogramas en tiempo real.

5.6. Conclusiones

En este capítulo se ha estudiado cómo mejorar el seguimiento de primitivas como parte de la calibración de cámaras en secuencias de vídeo. Estos vídeos son de escenarios donde, en cada fotograma, usualmente hay pocas primitivas visibles. Se propuso un nuevo método basado en un *CART (Classification and Regression Tree)*. El árbol de decisión se construye usando un proceso de aprendizaje basado en los canales *RGB* de la imagen y un conjunto de entrenamiento. Se han presentado varios experimentos usando vídeos de alta definición de eventos deportivos (concretamente partidos de fútbol), donde el procedimiento propuesto ha demostrado ser muy rápido y preciso. Usando una combinación de canales *RGB* el error de clasificación máximo es aproximadamente un 0,16 % de los píxeles (para imágenes que no están incluidas en el conjunto de entrenamiento). También se ha visto que con el método de seguimiento propuesto, se mejora el tiempo de procesado sin perder precisión. En términos de complejidad computacional, la principal novedad del método es que, además de que la computación de un árbol de decisión es muy rápida, el método es local. Es decir, que solamente se necesita procesar una vecindad de píxeles alrededor de la localización de las primitivas.

Capítulo 6

Variación del modelo de distorsión de la lente en una secuencia vídeo

6.1. Introducción

Se sabe que las lentes de las cámaras, debido a varios factores, introducen distorsión en las imágenes que capturan. La magnitud de esta distorsión depende de algunas características como la calidad de la lente o la magnitud del *zoom*. Una consecuencia importante de la distorsión de lentes es que al realizar la proyección $3D$ de líneas rectas en la imagen, éstas se convierten en curvas. Normalmente, los modelos de distorsión usados en visión por computador dependen de funciones radiales, y se pueden estimar utilizando solamente información de la imagen. El modelo básico de distorsión de lentes que se usa en visión por computador (ver por ejemplo [25, 35, 73]) viene dado por la siguiente expresión:

$$\hat{\mathbf{x}} \equiv \tilde{L}(\mathbf{x}) = \mathbf{x}_c + L(r)(\mathbf{x} - \mathbf{x}_c), \tag{6.1}$$

donde $\mathbf{x} = (x, y)$ es el punto original de la imagen (distorsionado), $\hat{\mathbf{x}} = (\hat{x}, \hat{y})$ es el punto corregido, $\mathbf{x}_c = (x_c, y_c)$ es el centro del modelo de distorsión de la cámara, generalmente cerca del centro de la imagen, $r = \sqrt{(x - x_c)^2 + (y - y_c)^2}$ y $L(r)$ es la función que define la forma del modelo de distorsión. Generalmente, $L(r)$ se aproxima por el polinomio:

$$L(r) = 1 + k_1 r^2 + k_2 r^4 + k_3 r^6 + \cdots, \tag{6.2}$$

donde el vector $\mathbf{k} = (k_1, k_2, \ldots, k_{N_k})^T$ representa los parámetros de distorsión radial. La complejidad del modelo viene dada por el grado del polinomio que se usa para aproximar $L(r)$ (por ejemplo la dimensión de $\mathbf{k}$). Se puede incluir en los modelos términos no radiales para tener en cuenta efectos tangenciales o de descentrado [33, 56, 73, 76], aunque para las lentes estándar normalmente se omiten. Los modelos de distorsión dados por la Ecuación (6.1) son bastante conocidos, simples y pueden estimarse usando solamente información de la imagen. En concreto, en el trabajo [5] se propone un método algebraico para calcular modelos de distorsión corrigiendo la distorsión de líneas introducida por la lente.

Aparte de lo descrito anteriormente, lo cual se aplica normalmente a cámaras monofocales, las características del *zoom* se deben tener en cuenta para calibrar correctamente una cámara en un escenario real, véase [17, 32, 38]. Modificando el foco y los valores del *zoom*, una cámara puede ajustarse a muchos campos de visión, profundidad de campos e incluso condiciones de iluminación. Las aplicaciones de las lentes con *zoom* son, por ejemplo: reconstrucción de la profundidad de escenas 3D [49], telepresencia [46], navegación de robots [40, 64],o seguimiento [34, 67]. En el rango de valores del *zoom*, la distorsión radial puede aparecer como distorsión de barril, normalmente afecta a distancias focales cortas, o distorsión de cojín, generalmente afecta a distancias focales largas. Por lo tanto, se requiere un modelo para tener en cuenta la variación de la distorsión dependiendo del rango de valores del *zoom*.

6.1.1. Contribución de este capítulo

Modelos matemáticos de distorsión de lentes dependientes del *zoom*:
Se proponen nuevos modelos matemáticos para estudiar la variación de los modelos de distorsión cuando se modifica el *zoom* de la lente. Los nuevos modelos están basados en una aproximación polinómica para tener en cuenta la variación de los parámetros de distorsión radial a lo largo del rango de *zoom* de la lente y minimizar un error global de la energía al medir la distancia entre secuencias de alineaciones de puntos distorsionados y líneas rectas después de la corrección de la distorsión. Para validar el rendimiento del método, se realizan experimentos con imágenes de un patrón de calibración y con vídeos de eventos deportivos.

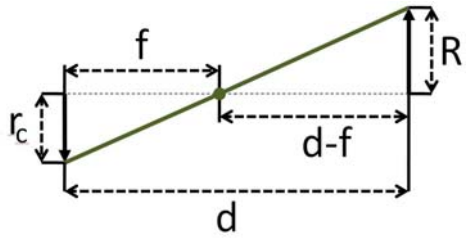

Figura 6.1: Modelo de proyección *pinhole*. f es la distancia focal efectiva.

6.2. Geometría de lentes con zoom

Se asume que, después de la corrección de la distorsión de la lente, la formación de la imagen sigue el modelo de proyección *pinhole*, el cual se usa ampliamente en visión por computador. En la Figura 6.1 se ilustra el modelo básico *pinhole* donde f es la distancia focal efectiva y d es la distancia entre un punto de la escena y el plano de proyección de la cámara. Usando relaciones trigonométricas se puede obtener:

$$\frac{r_c}{f} = \frac{R}{d-f}. \tag{6.3}$$

En el caso de una lente de *zoom* real, la f efectiva depende de dos parámetros de control ajustables de la lente:

1. Ajuste del *zoom*.
2. Parámetro de distancia de enfoque, que es la distancia entre el plano de proyección de la imagen y los puntos en la escena hacia donde enfoca la lente.

En la Figura 6.2 se ilustra el modelo básico de lente donde se puede apreciar la variación de la distancia focal efectiva con respecto a la distancia enfocada.

El ajuste del *zoom* es el parámetro más significativo en el valor de la distancia focal efectiva f. El máximo valor del intervalo del ajuste del *zoom* de la lente lo proporciona normalmente el fabricante. Por ejemplo, en los experimentos para este trabajo se usa una lente NIKKOR AF-S 18-200 con intervalo máximo de ajuste de *zoom* de la lente $(18, 200)$. Este intervalo se obtiene con la combinación adecuada de los ajustes del *zoom* y de la distancia de enfoque. En el caso de fijar el enfoque, el intervalo de la distancia

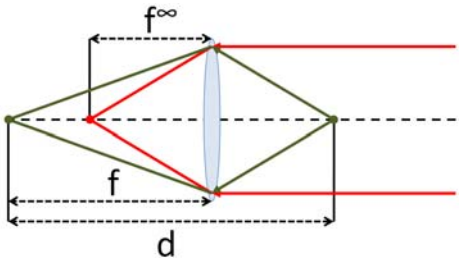

Figura 6.2: Modelo básico de lente. f^{∞} es la distancia focal de los puntos situados en el infinito. f es la distancia focal efectiva cuando la lente se enfoca a la distancia d.

focal efectiva es más pequeño. Como se verá en el apartado de experimentos, dicho intervalo es [20,56, 127,36] para una distancia de enfoque de 1185 milímetros.

6.3. Modelo de distorsión de lentes propuesto

Para estimar el modelo $L(r)$, se usa la aproximación general imponiendo el requisito de que después de la corrección de la distorsión, al proyectar líneas 3D en la imagen, éstas deben ser líneas rectas 2D. Esta aproximación se ha usado en [5, 25] para minimizar la siguiente función de error, la cual se expresa en términos de la distancia de los puntos de la primitiva a la línea asociada después de la corrección:

$$E\left(\{k_i\}\right) = \sum_{l}^{N_l} \sum_{p}^{N_p(l)} \frac{dist^2\left(\tilde{L}(\mathbf{x}_{l,p}), S_l\right)}{N_l \cdot N_p(l)}, \tag{6.4}$$

donde N_l es el número de líneas detectadas en la imagen, $N_p(l)$ es el número de puntos extraídos asociados a una línea, $\mathbf{x}_{l,p}$ es un punto asociado a la línea S_l, $\tilde{L}(.)$ es el modelo de distorsión de lentes dado por la Ecuación (6.1) y $E\left(\{k_i\}\right)$ es la media de la distancia al cuadrado del punto de la línea a la línea recta después de la corrección de la distorsión. Esta función de error se aplica ampliamente y el minimizado puede llevarse a cabo a través de cualquier método de optimización (basado en el gradiente). El objetivo principal de este capítulo es modelar la variación de los parámetros del modelo de distorsión de la lente con respecto a la distancia focal efectiva f. Primero

se observa que, usando la Ecuación (6.3) se obtiene:

$$R = d \cdot r_c \frac{1}{f} - r_c, \tag{6.5}$$

en concreto, la variación de R es lineal con respecto a $1/f$. Se espera que para el plano de enfoque, la magnitud de la distorsión de la lente dependa de R y por lo tanto, la elección natural para modelar la variación del parámetro k_i de la distorsión sea una función de $1/f$, esto es:

$$k_i(f) \equiv P_i(1/f), \tag{6.6}$$

donde $k_i(f)$ representa el parámetro de distorsión k_i para la distancia focal efectiva f. De hecho, en [55], los autores dividen el intervalo de distancia focal en dos regiones (área de cojín y de barril) y en cada área se usa una aproximación polinómica variable en $1/f$ para modelar la variación del *zoom*. Probablemente como ellos usan un modelo de distorsión de un solo parámetro, necesitan dividir el intervalo de distancia focal para mejorar la precisión. Pero en este trabajo se usan modelos de distorsión más complejos, y se puede tratar con el rango completo de distancias focales sin tener que separar el intervalo en varias regiones. En lo que sigue, se asumirá que $P_i(.)$ se aproxima por una función polinómica, que es:

$$P_i(x) \equiv a_0^i + a_1^i x + a_2^i x^2 + ... + a_N^i x^N, \tag{6.7}$$

por lo tanto el modelo de distorsión de la lente depende de $\{a_n^i\}$ y se denota por:

$$L_{\{a_n^i\}}(f, r) \equiv 1 + P_1(1/f)r^2 + P_2(1/f)r^4 + \cdots, \tag{6.8}$$

el modelo de distorsión de lentes radial para la distancia focal efectiva f y por:

$$\hat{\mathbf{x}} = \tilde{L}_{\{a_n^i\}}(f, \mathbf{x}), \tag{6.9}$$

siendo $\hat{\mathbf{x}}$ la corrección de la distorsión del punto $\mathbf{x}$ usando el modelo de distorsión descrito anteriormente. Se propone estimar los coeficientes polinómicos $\{a_n^i\}$ minimizando la función de error:

$$E_G\left(\{a_n^i\}\right) = \sum_m^M \sum_l^{N_l(m)} \sum_p^{N_p(l,m)} \frac{dist^2\left(\tilde{L}_{\{a_n^i\}}(f_m, \mathbf{x}_{m,l,p}), S_{m,l}\right)}{M \cdot N_l(m) \cdot N_p(l,m)}, \tag{6.10}$$

donde M es el número de imágenes, f_m es la distancia focal efectiva asociada a la imagen m, $N_p(l,m)$ es el número de puntos extraídos asociados a una primitiva en particular, $\mathbf{x}_{m,l,p}$ es un punto de una primitiva asociado a la línea $S_{m,l}$. Se observa que

$E_G(\{a_n^i\})$ es la media del error de distorsión en el fotograma dado por la Ecuación (6.4) cuando se estiman los coeficientes de distorsión usando los modelos polinómicos. Concerniente a la variación del centro de distorsión, no se asume ningún modelo porque no se espera una variación significativa. Como se verá en los experimentos, la influencia de la variación del centro de distorsión es despreciable. Por eso se asume que el centro de distorsión de la lente es el centro de la imagen.

En lo que sigue, se llamará modelo fotograma a fotograma al modelo de distorsión de lente estimado independientemente para cada fotograma sin el rango de interés del *zoom*. Para un polinomio de grado n, para el desarrollo de Taylor de la Ecuación (6.1), el modelo fotograma a fotograma para m imágenes, es el conjunto de coeficientes de distorsión radial proporcionados por el minimizado de la Ecuación (6.4), que se expresa como:

$$\mathbf{k} = \left\{ \left(k_1^p, k_2^p, \ldots k_n^p\right), p = 1, 2, \ldots, m \right\}. \tag{6.11}$$

6.4. Experimentos

6.4.1. Configuración de los experimentos

Para validar el modelo propuesto, se ha construido un patrón de calibración (ver Figura 6.3) compuesto por una colección de 31×23 franjas blancas. Las dimensiones del patrón de calibración son 1330×1010 milímetros. La cámara está fija en frente del patrón de calibración y se toman un cierto número de fotografías cambiando el ajuste del *zoom* de la lente cubriendo el intervalo de valores completo. Para cada imagen se estiman las líneas de los bordes de las franjas blancas (usando por ejemplo el método propuesto en [4]) que proporciona las líneas distorsionadas que se usan para validar el modelo. Además, para cada imagen, la distancia focal efectiva se estima usando la expresión:

$$f = \frac{d \cdot r_c}{R + r_c}, \tag{6.12}$$

obtenida de la Ecuación (6.3) donde d es la distancia desde el plano de proyección de la cámara al patrón de calibración, r_c es la distancia entre dos franjas consecutivas en la imagen y R es la distancia entre dos franjas consecutivas en el patrón de calibración. Nótese que los errores pequeños relacionados con el valor de distancia al objetivo, d en

la Ecuación (6.12), serán de alguna manera compensados durante el minimizado global y, por tanto, no afectarán a la eficiencia del modelo. Resumiendo, el procedimiento que se usa para validar la aproximación propuesta usando el patrón de calibración puede dividirse en los siguientes pasos:

1. Se toma una colección de fotografías del patrón de calibración para una distancia de enfoque fija, cubriendo el intervalo completo de valores de ajuste del *zoom*.

2. Extraer las líneas distorsionadas de la imagen.

3. Para cada imagen se calcula la distancia focal efectiva usando la Ecuación (6.12).

4. Se estima el modelo polinómico de distorsión de lentes con *zoom* minimizando la Ecuación (6.10).

5. Se analiza el error de distorsión obtenido usando: (i) El modelo de distorsión dependiente del *zoom* propuesto para el intervalo completo de valores de *zoom*, (ii) modelo de distorsión obtenido independientemente para cada imagen minimizando el error de la energía de la Ecuación (6.4) y (iii) el error original de la distorsión de la lente sin usar ninguna corrección de la distorsión.

La Ecuación (6.10), se minimiza con un método de gradiente simple aplicando una longitud de paso apropiada para considerar las diferencias en magnitud de las variables (nótese que pueden ser de 1.e-005 a 1.e-017). Se estima la solución inicial de la siguiente manera:

1. Se seleccionan algunas imágenes y se calculan los coeficientes de distorsión para el modelo fotograma a fotograma.

2. Se ajustan los polinomios cuadráticos de la Ecuación (6.8) a los coeficientes de distorsión del modelo fotograma a fotograma usando el error de mínimos cuadrados.

Los resultados que se muestran en las siguientes secciones fueron calculados usando solamente tres imágenes para el modelo fotograma a fotograma: la imagen correspondiente a la distancia focal máxima f_{max}, la correspondiente a la mínima f_{min}, y una imagen capturada con distancia focal igual a $(f_{min} + f_{max})/2$.

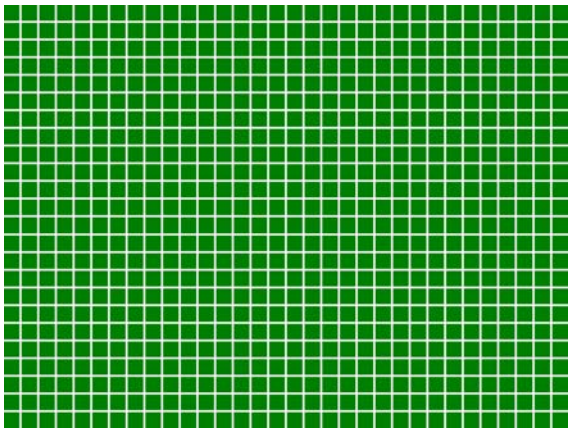

Figura 6.3: Patrón de calibración compuesto por una colección de 31×23 de franjas blancas.

6.4.2. Realización de experimentos

Se han realizado experimentos en dos escenarios diferentes. En el primero de ellos, se comprueba la precisión del modelo propuesto usando el patrón de calibración presentado anteriormente. En el otro caso, se aplica el modelo a un escenario real: una secuencia de vídeo de un partido de fútbol con una variación significativa del *zoom* de la lente. En la Figura 6.4 se muestran fotogramas de los vídeos de ambos escenarios.

En el experimento con el patrón de calibración, se usa una cámara Nikon D90 con una lente NIKKOR AF-S 18-200 mm., y un CCD de $23{,}7 \times 15{,}6$ mm. La resolución de la imagen capturada es de 4288×2848 píxeles. En la secuencia de vídeo del partido de fútbol, se ha usado una vídeo cámara profesional de alta definición con una resolución de 1920×1080 píxeles (el tipo de cámara de vídeo que se usa normalmente en la emisión de eventos deportivos). En este caso, el fabricante de la cámara y el rango de *zoom* de la lente son desconocidos. Para cada fotograma, se estima la distancia focal efectiva teniendo en cuenta las dimensiones reales de un campo de fútbol (las cuales se conocen "a priori") y el tamaño del campo de fútbol proyectado en la imagen. Este tamaño se puede calcular usando la homografía de la imagen del campo de fútbol al modelo real correspondiente. Dicha homografía puede estimarse usando la aproximación propuesta en el método clásico de calibración de Zhang [88]. Nótese que la Ecuación (6.12) no se usa en este caso. Hay que destacar las diferencias significativas existentes entre los escenarios seleccionados para los experimentos, las cuales ponen de manifiesto las amplias posibilidades de la metodología propuesta.

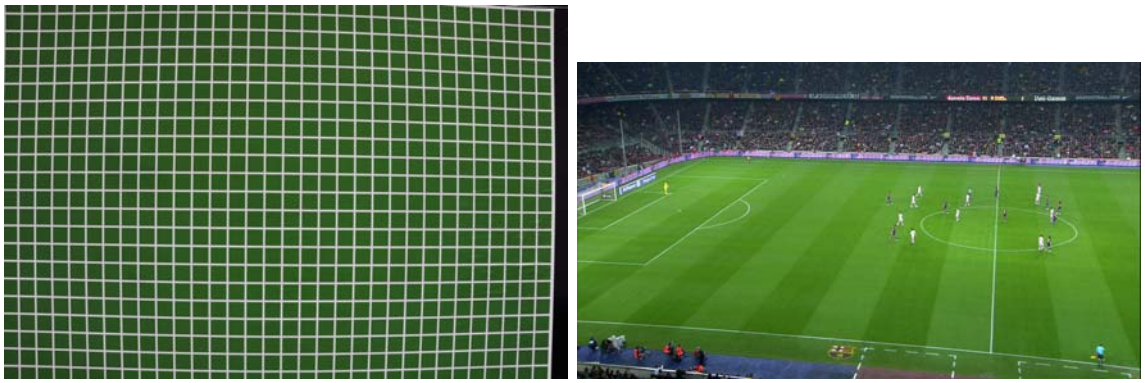

Figura 6.4: Ejemplo de fotogramas de las secuencias de vídeo usadas para los experimentos: patrón de calibración (izquierda), partido de fútbol (derecha)

Para el patrón de calibración, en la Figura 6.5 se presentan las primitivas de la imagen que se usan para calcular los modelos de distorsión que se incluyen en el modelo de *zoom* propuesto. Se pueden ver las primitivas seleccionadas para dos casos correspondientes al *zoom* mínimo (ajuste del *zoom* de la lente = 18 mm que corresponde a una distancia focal efectiva f=20.55 mm. medida a partir del proceso de calibración, como se indicó anteriormente) y al *zoom* máximo (ajuste del *zoom* de la lente = 200mm y una distancia focal efectiva f=127.35 mm., medida a partir de la calibración). Las primitivas de la imagen son un conjunto de puntos de bordes pertenecientes, respectivamente, a las franjas blancas horizontales y verticales del patrón. Para obtener este conjunto de puntos, se puede usar cualquier detector de bordes (ver por ejemplo [4] o [6] para más detalles). Por ejemplo, el número de primitivas que se extrajo fue, para el caso f = 20.55 mm., 303,623 puntos y 103 líneas y para el caso f = 127.35 mm., 52,433 puntos y 18 líneas. La cantidad total de puntos de primitivas extraídos en las 50 imágenes que se usaron en el experimento fueron 8,166,660.

Para el caso de la secuencia de vídeo de fútbol, se puede observar en la Figura 6.6 las primitivas que han sido seleccionadas para considerar el modelo de distorsión radial. En este caso, el número total de puntos de primitivas disponible es más pequeño que en el caso del patrón de calibración y corresponden a los centros de las líneas blancas que aparecen en el terreno de juego (línea de banda, línea de medio campo, área grande y área pequeña), como se aprecia en la imagen. Nótese que estas primitivas puede que no estén siempre visibles, así que, calibrar este tipo de imágenes es un problema difícil porque hay un número pequeño de primitivas visibles para realizar la calibración (véase Capítulos 3 y 4).

Se representan los casos para dos ajustes del *zoom*, f = 45.16 mm. y f = 156.55 mm.

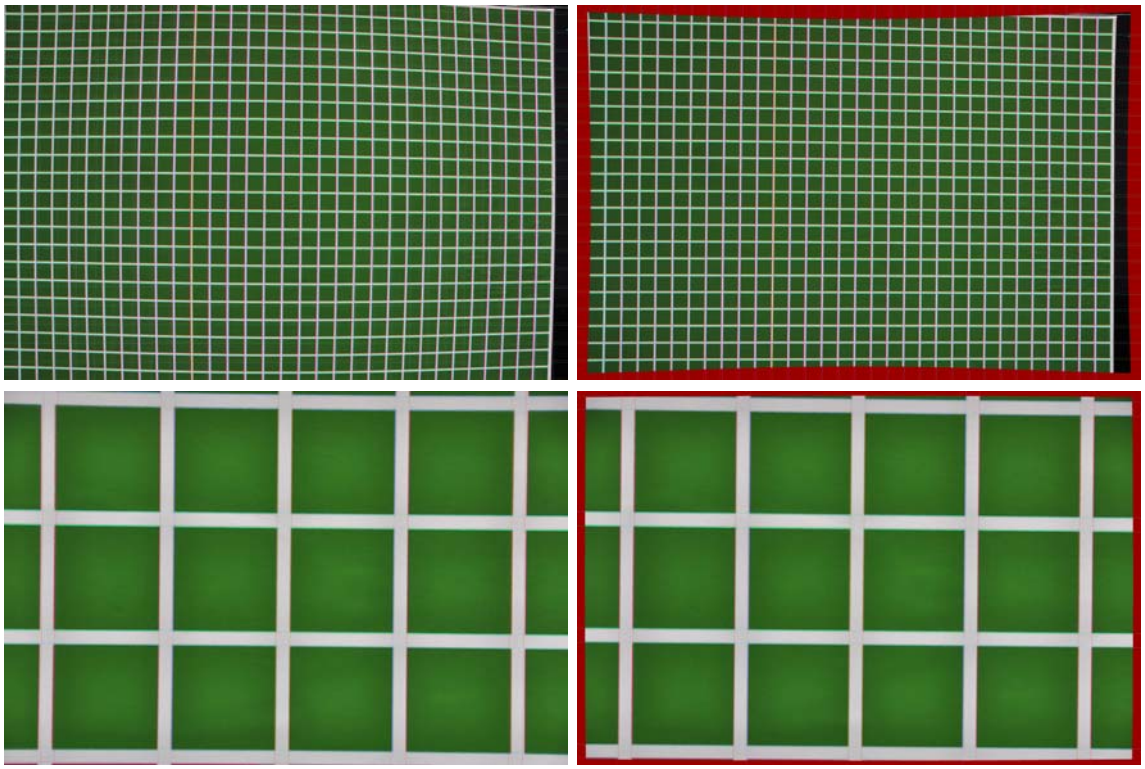

Figura 6.5: A la izquierda, las primitivas usadas en los experimentos del patrón geométrico: arriba (f = 20.55 mm.), y abajo (f = 127.35 mm.). A la derecha, imágenes y primitivas donde se ha eliminado la distorsión aplicando el modelo de *zoom* propuesto. Se aprecia la variación de los modelos de distorsión con respecto a f, mirando la curvatura del límite de la imagen: en el caso de f = 127.35 mm., la corrección de la distorsión de la lente es significativamente más pequeña que en el caso de f = 20.55 mm.

los cuales corresponden a los extremos de la distancia focal efectiva de la secuencia de vídeo. El número de primitivas extraído fue, para el caso 45.16 mm., 1060 puntos y 13 líneas, y 957 puntos seleccionados y 8 líneas, para el caso f = 156.55 mm. La cantidad total de puntos de primitivas extraídos de las 55 imágenes que se usaron en el experimento fueron 55,447.

Primero se evaluó el rendimiento del modelo de distorsión de lentes con *zoom* propuesto para el patrón de calibración geométrico y, después de una evaluación detallada, se aplicó el modelo a la secuencia de vídeo de fútbol.

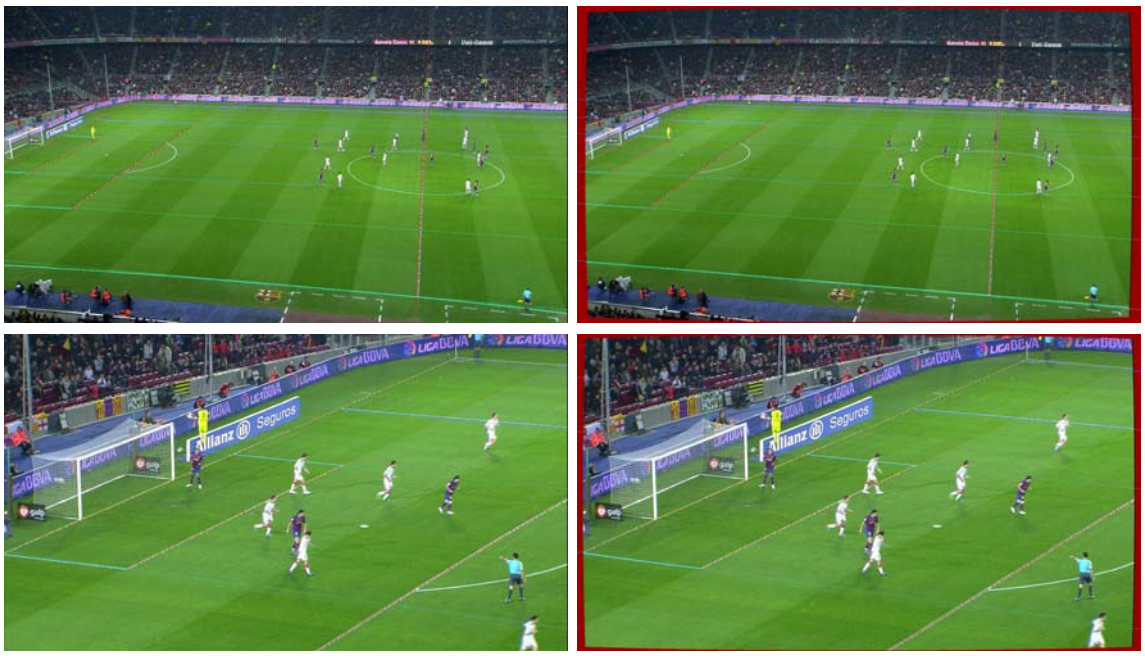

Figura 6.6: A la izquierda, las primitivas usadas en el experimento para la secuencia de vídeo de fútbol: arriba (f = 45.16 mm.), y abajo (f = 156.55 mm.). A la derecha, imágenes y primitivas con la distorsión eliminada aplicando el modelo del *zoom* propuesto. Se puede apreciar la variación de los modelos de distorsión con respecto a f, mirando la curvatura de los límites de la imagen: en el caso de f = 156.55 mm., la corrección de la distorsión de lente es significativamente más pequeña que en el caso de f = 45.16 mm.

Cuadro 6.1: Resumen de resultados para el patrón geométrico (valores RMS).

Comparación de modelo de distorsión	*Píxeles*	*Milímetros*
Residuo sin usar modelo de distorsión	4.2991	2.0421
Residuo del modelo fotograma a fotograma	1.8192	0.8482
Residuo del modelo de zoom polinómico propuesto	1.8241	0.8604

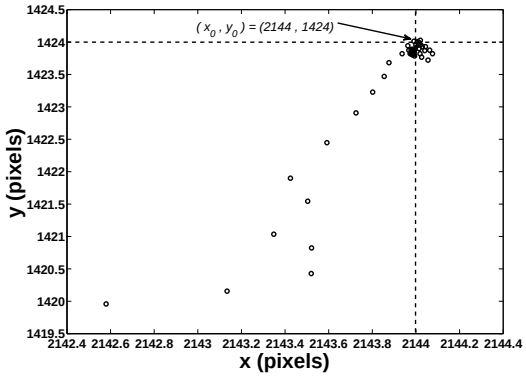

Figura 6.7: Desplazamiento del centro de distorsión para el patrón geométrico.

6.4.3. Resultados para el patrón de calibración

Hay que decir que el patrón de calibración puede ser visto como un experimento ideal de *zoom* con una distribución densa de líneas que permiten analizar con precisión el comportamiento del modelo de distorsión de la lente. Primero se evaluó la influencia de la variación de la distorsión de la lente en el rango de distancia focal (f = 20.55 mm. a f = 127.33 mm.). El centro de la distorsión radial se minimiza cuando se estiman los coeficientes de distorsión k_1 y k_2 para el modelo fotograma a fotograma. Se usa el método algebraico [5] para estimar dichos coeficientes y, por medio del *steepest descent algotihm*, se ha mejorado la solución calculando la función de distancia *RMS* como se explicó en [5].

Conforme a los resultados obtenidos, se puede concluir que la variación del centro de distorsión de la lente se puede despreciar por dos razones. Primero, como se muestra en la Figura 6.7, el desplazamiento del centro de distorsión para el modelo de distorsión de la Ecuación (6.11) es muy pequeño (con una norma máxima de 4 píxeles). Segundo, como se muestra en la Figura 6.8, la mejora relativa del porcentaje de error de la energía de la Ecuación (6.4), cuando se optimiza el centro de distorsión de la lente y cuando no se optimiza (el centro de distorsión es el centro de la imagen), es muy pequeño (con un porcentaje máximo de 1,5 %).

Por tanto, se puede concluir que la influencia de la variación del centro de distorsión de la lente es despreciable. Por ello, en lo que sigue se considerará que el centro de distorsión de la lente es el centro de la imagen.

La variación de los coeficientes de distorsión estimados a lo largo del campo de

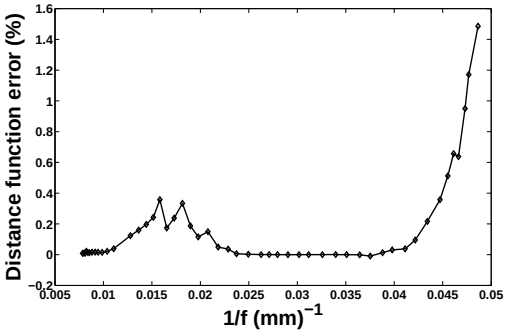

Figura 6.8: Error relativo del porcentaje de mejora optimizando el centro de distorsión.

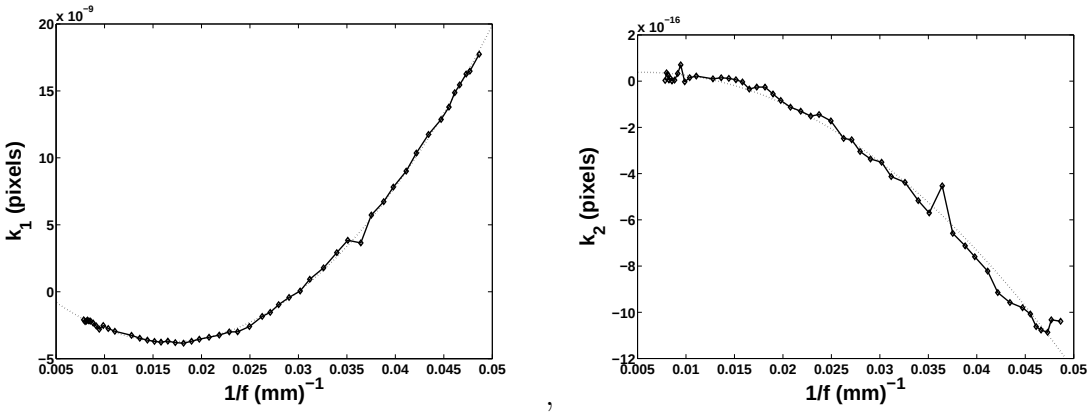

Figura 6.9: Variación de los coeficientes de distorsión estimados k_1 (izquierda) y k_2 (derecha) con la inversa de la distancia focal y la aproximación del polinomio de segundo orden estimado. Se observa que los polinomios encajan de manera precisa con la distribución de los parámetros de distorsión. (especialmente en el caso de k_1 el cual es el parámetro más importante). Además la variación de k_1 y k_2 con respecto a los polinomios se mueve en direcciones opuestas entonces se espera una compensación del movimiento en términos de la corrección del modelo de distorsión.

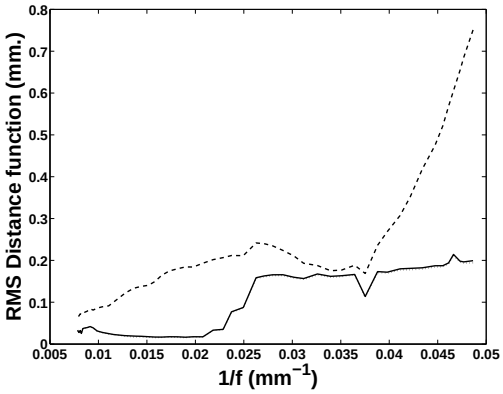

Figura 6.10: Función de distancia para el patrón geométrico estimada con los tres modelos. Línea discontinua: función de error original, línea continua: modelo cuadrático de *zoom* propuesto, línea de puntos: modelo fotograma a fotograma (Ecuación (6.11)).

zoom usando el modelo fotograma a fotograma se puede ver en la Figura 6.9 (representado como una función de la inversa de la distancia focal). También se representa la aproximación del polinomio de segundo orden estimado. Se observa que los polinomios encajan de manera precisa con la distribución de los parámetros de distorsión (especialmente en el caso de k_1 el cual es el parámetro más importante). Además, se puede observar que, en general, en las distancias focales donde k_1 varía con respecto a la aproximación polinómica, la variación de k_2 con respecto al polinomio se mueve en direcciones opuestas. Entonces se espera una compensación del movimiento en términos de la corrección del modelo de distorsión.

La Figura 6.10 muestra el rendimiento del modelo de *zoom* propuesto. En dicha figura se presenta por cada fotograma: (i) el error original de distorsión de la lente sin usar ninguna corrección de la distorsión de la lente. (ii) Modelo de distorsión de la lente obtenido independientemente para cada imagen minimizando el error de energía. (iii) Error de energía calculado usando el modelo polinómico de distorsión dependiente del *zoom*. Se observa que la calidad de la corrección de la distorsión obtenida usando el modelo propuesto es tan buena como la obtenida de la forma fotograma a fotograma.

En el Cuadro 6.1 se resumen los valores *RMS* para los 3 modelos presentados en la Figura 6.10. A partir de estos resultados, se puede apreciar que la diferencia relativa del valor *RMS* entre el modelo dependiente del *zoom* y el modelo estimado independientemente fotograma a fotograma es aproximadamente 1,43 % (para los resultados en mm.). Debido al hecho de que el patrón de calibración está colocado en una posición frontal paralela con respecto al plano de proyección de la cámara, el cambio de unidad

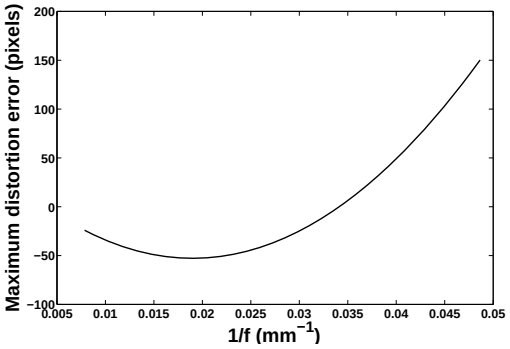

Figura 6.11: Error máximo de distorsión estimado usando el modelo propuesto para el patrón. El error se calcula como $\|\hat{\mathbf{x}} - \mathbf{x_c}\| - \|\mathbf{x} - \mathbf{x_c}\|$, donde, $\hat{\mathbf{x}}$ es el punto corregido, $\mathbf{x}$ es un punto localizado en una esquina de la imagen, y $\mathbf{x_c}$ es el centro de distorsión.

entre píxeles y mm. es trivial usando la Ecuación (6.3).

En la Figura 6.11 se muestra el error máximo de la distorsión. Este error se calculó para un píxel localizado en una esquina de la imagen (la esquina superior). Nótese que el error máximo abarca alrededor de 200 píxeles para corregir la distorsión radial debida al *zoom*.

En este experimento, los coeficientes del modelo optimizado de distorsión de lente con *zoom* vienen dados por los polinomios:

$$k_1(f) = 2{,}26 \times 10^{-9} - 7{,}19 \times 10^{-7}(1/f) + 2{,}14 \times 10^{-5}(1/f)^2,$$

$$k_2(f) = 1{,}74 \times 10^{-17} + 7{,}36 \times 10^{-15}(1/f) - 6{,}55 \times 10^{-13}(1/f)^2.$$

6.4.4. Resultados para la secuencia de fútbol

La secuencia de vídeo que se usa para los experimentos ha sido proporcionada por MEDIAPRODUCCION S.L. La secuencia de vídeo está en alta definición (1920×1080 píxeles) y dura 28 segundos (841 fotogramas). El rango de ajuste del *zoom* va desde 45.16 mm. a 156.55 mm. Para estimar el modelo polinómico propuesto dependiente del *zoom* se seleccionaron 55 fotogramas cubriendo el rango completo de distancia focal efectiva. Se obtuvieron los siguientes modelos polinómicos para los coeficientes de distorsión de la lente:

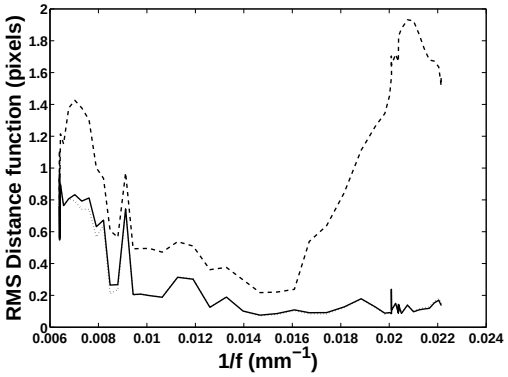

Figura 6.12: Función distancia para la secuencia de vídeo de fútbol estimada por los tres modelos. Línea discontinua: error de la función original, línea continua: modelo cuadrático de *zoom* propuesto, línea de puntos: modelo fotograma a fotograma (Ecuación (6.11)).

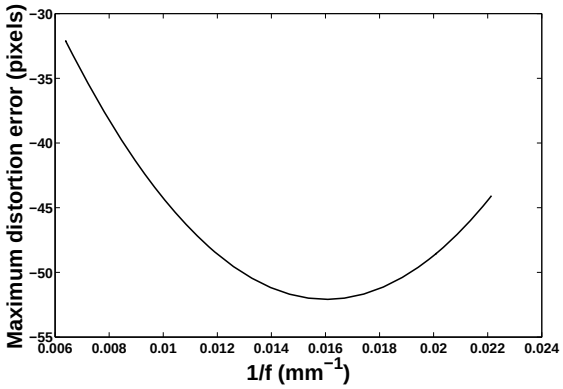

Figura 6.13: Error máximo de distorsión estimado para el fútbol usando el modelo propuesto. El error se calcula como $\|\hat{\mathbf{x}} - \mathbf{x_c}\| - \|\mathbf{x} - \mathbf{x_c}\|$, donde, $\hat{\mathbf{x}}$ es el punto corregido, $\mathbf{x}$ es un punto localizado en una esquina de la imagen, y $\mathbf{x_c}$ es el centro de distorsión.

$$k_1(f) = 2{,}65 \times 10^{-8} - 8{,}88 \times 10^{-6}(1/f) + 2{,}86 \times 10^{-4}(1/f)^2,$$
$$k_2(f) = 1{,}99 \times 10^{-14} + 3{,}07 \times 10^{-12}(1/f) - 1{,}04 \times 10^{-10}(1/f)^2.$$

En la Figura 6.12 se ilustra el rendimiento del modelo propuesto. Como el caso del experimento con el patrón de calibración, se presenta una comparación de las medidas de error de la distorsión de lentes entre (i) el error original de la distorsión de la lente (Ecuación (6.4)) sin usar ninguna corrección de la distorsión, (ii) modelo de distorsión de la lente obtenido independientemente para cada imagen minimizando el error de la energía (Ecuación (6.4)) y (iii) error de la energía (Ecuación (6.4)) calculado usando el modelo polinómico propuesto de distorsión de lente dependiente del *zoom*. Se observa que la calidad de la corrección de la distorsión obtenida usando el modelo de *zoom* propuesto es tan bueno como el obtenido de la manera independiente fotograma a fotograma.

En el Cuadro 6.2 se resumen los valores *RMS* para los 3 modelos presentados en la Figura 6.12. Se incluye también el modelo de *zoom* simple para compararlo con el resto. A partir de estos resultados, se puede apreciar que la diferencia relativa de los valores *RMS* entre el modelo propuesto y el modelo fotograma a fotograma es solamente de 1,33 %. Estos resultados se expresan en píxeles porque, como la cámara no está en posición frontal paralela con respecto a la vista, no se puede asociar una medida única y real (metros) al tamaño del píxel.

En la Figura 6.13 se muestra el error máximo de distorsión. Como se puede ver, varía sobre 20 píxeles con el *zoom* para corregir la distorsión radial.

Una ventaja muy importante del modelo propuesto es que usando los polinomios obtenidos se puede estimar el modelo de distorsión de la lente para cualquier distancia focal efectiva f. En concreto, se pueden obtener modelos de distorsión de la lente para la secuencia de vídeo completa (841 fotogramas) en vez de los 55 fotogramas que se han usado para estimar los polinomios.

Para ilustrar la aplicación de la propuesta a la secuencia de vídeo completa, se ha creado un vídeo donde se corrigió la distorsión en cada fotograma usando el modelo polinómico propuesto dependiente del *zoom* (puede verse el vídeo demostración en *http://www.ctim.es/demo101/*).

Cuadro 6.2: Resumen de los resultados para el conjunto de imágenes de la secuencia de vídeo del fútbol (valores RMS).

Comparación modelo de distorsión	*Píxeles*
Residuo sin usar el modelo de distorsión de lente	1.0601
Residuo del modelo fotograma a fotograma	0.5478
Residuo del modelo polinómico de zoom propuesto	0.5551

6.5. Conclusiones

En el capítulo se han tratado nuevos modelos matemáticos para estudiar la variación de los modelos de distorsión de lentes en cámaras con *zoom*. Tales modelos están basados en una aproximación polinómica para tener en cuenta la variación de los parámetros de distorsión radial a lo largo del rango del *zoom* y, el minimizado de la energía del error global midiendo la distancia entre secuencias de puntos alineados distorsionados y líneas rectas después de la corrección de la distorsión. Esto se ha obtenido usando una aproximación polinómica de segundo orden, la calidad de la corrección de la distorsión de la lente con el modelo propuesto, es tan buena como la aproximación fotograma a fotograma. Esto es destacable porque usando solamente 6 parámetros (3 para el polinomio asociado al primer coeficiente de distorsión k_1 y 3 parámetros para el segundo coeficiente k_2) se puede estimar el modelo de distorsión de lente para cualquier distancia focal efectiva de la lente con *zoom*. El modelo propuesto se ha aplicado a dos secuencias de vídeo diferentes para estimar el modelo de distorsión de lente dependiente del *zoom*, dichas secuencias son un vídeo de un patrón de calibración y un vídeo real de fútbol grabado con una vídeo cámara profesional. Los resultados para ambos casos muestran la potencialidad del nuevo modelo.

Capítulo 7

Suavizado del movimiento de una cámara en una secuencia vídeo

7.1. Introducción

En los capítulos anteriores, se han tratado varios aspectos de la calibración de cámaras de vídeo, y en este capítulo se introduce la condición de suavizado para dicha calibración. La atención se centra de nuevo en las cámaras montadas en un trípode. En este tipo de escenarios, para cada instante t de la secuencia de vídeo, la configuración de la calibración de la cámara viene dada por 3 parámetros: $P(t)$ *(Pan)* y $T(t)$ *(Tilt)* que representan la rotación en los ejes vertical y horizontal del trípode respectivamente, y $Z(t)$ *(Zoom)* es el ajuste de *zoom* de la lente de la cámara. De estas funciones dependientes del tiempo se puede deducir fácilmente (usando la información del trípode), para cada instante t, la calibración de la cámara. El sistema visual humano es muy sensible al movimiento, y pequeñas perturbaciones en los valores $P(t)$, $T(t)$, y $Z(t)$ durante el tiempo, producen pequeñas oscilaciones en el movimiento de la cámara molestando al observador. Eliminar esas perturbaciones es un punto crítico en aplicaciones que incluyen gráficos artificiales en vídeo real. Ya que si en los gráficos virtuales existe vibración, el espectador notará que los gráficos están insertados artificialmente y dificultará la comprensión de la escena. Para ilustrar este fenómeno, en http://www.ctim.es/demo102 se encuentra una secuencia de vídeo donde se han introducido varios objetos. En el vídeo se puede observar el problema con el que hay que tratar, porque las técnicas estándar de calibración que no tienen en cuenta la regularidad temporal de $\mathbf{u}(t) = (P(t), T(t), Z(t))$ pueden introducir un ruido significativo en la estimación del movimiento de la cámara durante el tiempo. En este trabajo se

Figura 7.1: Un fotograma de un escenario típico de aplicación. Las primitivas que se usan para calibrar son las líneas blancas y los círculos detectados en la imagen.

propone incluir una restricción de suavizado en la estimación de $\mathbf{u}(t)$ minimizando la siguiente energía:

$$I[\mathbf{u}] = \int_{t_0}^{t_1} \left(P'(t)^2 + T'(t)^2 + Z'(t)^2 + \alpha F(\mathbf{u}(t), t) \right) dt, \tag{7.1}$$

donde $[t_0, t_1]$ es el intervalo de tiempo, $\alpha \geq 0$ es un peso para balancear los distintos componentes de la energía y, $F(x_1, x_2, x_3, t) \geq 0$ es una función de calibración estándar que fuerza, en cada instante t, que la proyección de los puntos 3D esté cerca de las primitivas detectadas en la imagen. De hecho, cuando no se usa regularización temporal, $\mathbf{u}(t)$ se estima generalmente minimizando $F(\mathbf{u}(t), t)$ independientemente para cada instante t. Usando el modelo variacional propuesto (Ecuación (7.1)), se introduce una condición de regularidad temporal en el procedimiento de calibración de la cámara de vídeo.

Las ecuaciones de Euler-Lagrange asociadas a la energía, Ecuación (7.1), producen el siguiente sistema de ecuaciones diferenciales no lineales:

$$\begin{cases} -P''(t) + \alpha \frac{\partial F}{\partial x_1}(P(t), T(t), Z(t), t) = 0 \quad en \quad (t_0, t_1) \\ -T''(t) + \alpha \frac{\partial F}{\partial x_2}(P(t), T(t), Z(t), t) = 0 \quad en \quad (t_0, t_1) \\ -Z''(t) + \alpha \frac{\partial F}{\partial x_3}(P(t), T(t), Z(t), t) = 0 \quad en \quad (t_0, t_1), \end{cases}$$

con las condiciones adecuadas para los límites.

7.1.1. Contribución de este capítulo

Aproximación variacional al suavizado del movimiento de cámaras: En este trabajo se estudia el problema variacional que se deriva de la calibración de vídeo con restricción de suavizado. En este caso, se trabaja con cámaras montadas en un trípode, y para cada fotograma capturado en el instante t, la calibración queda definida por 3 parámetros: *Pan,Tilt* y *Zoom*. La función de calibración $t \to \mathbf{u}(t) = (P(t), T(t), Z(t))$ se obtiene como el mínimo de una función de energía $I[\mathbf{u}]$. Por ello, en esta aportación se estudia la existencia de un mínimo de dicha función de energía a la vez que las soluciones de las ecuaciones de Euler-Lagrange asociadas. El estudio matemático del modelo propuesto fue realizado por el director de tesis D. Luis Álvarez León, siendo mi contribución personal el desarrollo de la parte experimental.

7.2. Geometría y calibración de las cámaras montadas en un trípode

Un trípode se define por dos ejes de rotación unitarios $\bar{e}^0 = \left(\bar{e}^0_x, \bar{e}^0_y, \bar{e}^0_z\right)^T$ y $\bar{e}^1 = \left(\bar{e}^1_x, \bar{e}^1_y, \bar{e}^1_z\right)^T$ y un centro de rotación $\bar{X}_0 \in R^3$. El parámetro *Pan* $P(t)$ determina el ángulo de rotación del trípode con respecto a $\bar{e}^0$. Se define $R\left(\bar{e}^0, P(t)\right)$ la matriz de rotación asociada. Y de manera similar, $R\left(\bar{e}^1, T(t)\right)$. La matriz de rotación general producida por el movimiento del trípode es una composición de la matriz definida anteriormente. Dado un punto 3D $\bar{X}$ la transformación inducida por el movimiento del trípode puede expresarse como:

$$\bar{X}\left(P(t), T(t)\right) = \bar{X}_0 + R\left(\bar{e}^0, P(t)\right) R\left(\bar{e}^1, T(t)\right)\left(\bar{X} - \bar{X}_0\right). \tag{7.2}$$

Se usa el modelo básico *pinhole* para modelar la forma en que la escena 3D se

proyecta en el plano de proyección de la imagen 2D. Dicha proyección se expresa en coordenadas proyectivas como una matriz 4×3 de proyección $\mathcal{P}(\mathbf{u}(t))$ definida por:

$$\mathcal{P}(\mathbf{u}(t)) \equiv A\left(Z(t)\right) R_0 \left[Id, -c^0\right] \begin{pmatrix} R\left(P(t), T(t)\right) & \bar{t}\left(P(t), T(t)\right) \\ 0 & 1 \end{pmatrix}, \tag{7.3}$$

donde :

$$R\left(P(t), T(t)\right) \equiv R\left(\bar{e}^0, P(t)\right) R\left(\bar{e}^1, T(t)\right), \tag{7.4}$$

$$\bar{t}(P(t), T(t)) = \bar{X}_0 - R\left(P(t), T(t)\right) \bar{X}_0, \tag{7.5}$$

$$A\left(Z(t)\right) = \begin{pmatrix} Z(t) & 0 & x_c \\ 0 & r \cdot Z(t) & y_c \\ 0 & 0 & 1 \end{pmatrix}, \tag{7.6}$$

$$R_0 = \begin{pmatrix} r^0_{00} & r^0_{01} & r^0_{02} \\ r^0_{10} & r^0_{11} & r^0_{12} \\ r^0_{20} & r^0_{21} & r^0_{22} \end{pmatrix}, \tag{7.7}$$

$$\left[Id, -c^0\right] = \begin{pmatrix} 1 & 0 & 0 & -\bar{c}^0_x \\ 0 & 1 & 0 & -\bar{c}^0_y \\ 0 & 0 & 1 & -\bar{c}^0_z \end{pmatrix}, \tag{7.8}$$

R_0 y $\bar{c}^0$ corresponden a la rotación inicial del trípode y su traslación. Para más detalles sobre el modelo *pinhole* véase, por ejemplo, [33] y [35].

7.3. Función de calibración $F(\mathbf{u}(t), t)$

Calibrar una imagen es obtener los parámetros $\mathbf{u}(t) = (P(t), T(t), Z(t))$ asociados que determinan la matriz de proyección $\mathcal{P}(\mathbf{u}(t))$. Para estimar $\mathbf{u}(t)$ se usa la información observable que se puede obtener de la imagen. En general se define con $\Omega(t) \subset R^2$ la colección finita de curvas visibles en la imagen que se usan para calibrar el fotograma t. Se asume que se conoce la posición 3D real de $\Omega(t)$ en la escena real. Es decir, para cualquier curva $\tilde{s}(.) \in \Omega(t)$, se conoce su posición $3D$ real $\tilde{S}(.)$. Por tanto, la función de calibración $3D$ $F(\mathbf{u}(t), t)$ queda definida como:

$$F(\mathbf{u}(t), t) \equiv \sum_{\tilde{s} \in \Omega(t)} \oint_{\tilde{s}} \text{distancia}(\mathcal{P}(\mathbf{u}(t))\tilde{S}(.), \tilde{s}(q))^2 dq, \tag{7.9}$$

donde distancia$(C, \mathbf{x})$ es la distancia Euclídea entre un punto $\mathbf{x}$ y una curva C. Se observa que para cualquier $\mathbf{u}(t)$, $F(\mathbf{u}(t), t) \geq 0$ y, entre menor es $F(\mathbf{u}(t), t)$, mejor es la coincidencia entre la escena 3D y su proyección en la imagen 2D. La forma usual de calibrar una cámara (que es obtener $\mathbf{u}(t)$) es minimizando la función de calibración $F(\mathbf{u}(t), t)$. También se ha visto que $F(\mathbf{u}(t), t)$ no es convexo, fuertemente no lineal y, en general, no se puede esperar la existencia de un único mínimo porque la función es muy dependiente de la geometría de las curvas observables $\Omega(t)$ (de hecho, en algunos casos $\Omega(t)$ podría estar vacío).

7.4. Formulación variacional del problema de calibración de vídeo

Normalmente, cuando se minimiza la función de calibración $F(\mathbf{u}(t), t)$ con respecto a $\mathbf{u}(t)$ no se asume nada con respecto a la regularidad temporal t de $\mathbf{u}(t)$. Para añadir la condición de regularidad al modelo de calibración, se propone minimizar el funcional:

$$I[\mathbf{w}] = \int_{t_0}^{t_1} L(D\mathbf{w}(t), \mathbf{w}(t), t)dt, \tag{7.10}$$

donde $[t_0, t_1]$ es el intervalo de tiempo y

$$L(p, z, t) = \|p\|^2 + \alpha F(z, t), \tag{7.11}$$

$\alpha \geq 0$ es un peso para equilibrar los distintos componentes de la energía.

Seguidamente se muestra la existencia del minimizador de $I[.]$. Sea

$$\mathcal{A} = \left\{\mathbf{w} \in W^{1,2}((t_0, t_1); R^3) \quad \text{tal que} \quad \mathbf{w}(t_0) = (P_0, T_0, Z_0) \text{ y } \mathbf{w}(t_1) = (P_1, T_1, Z_1)\right\}.$$

Teorema 1. (existencia de minimizador) : Existe $\mathbf{u} \in \mathcal{A}$ resolviendo:

$$I[\mathbf{u}] = \min_{\mathbf{w} \in \mathcal{A}} I[\mathbf{w}].$$

Demostración. Para demostrar la existencia de minimizador se usa el resultado clásico que se presenta en [27]: *(Evans [27] (Pg. 453 THEOREM 5): Asumir que L satisface la desigualdad:*

$$L(p, z, x) \geq \alpha \|p\|^q - \beta, \tag{7.12}$$

para constantes $\alpha > 0$, $\beta \geq 0$ y $q > 1$ y es convexo en la variable p. Suponer también que el conjunto admisible A no está vacío. Entonces existe $u \in A$ resolviendo:

$$I[\mathbf{u}] = \min_{\mathbf{w} \in \mathcal{A}} I[\mathbf{w}].$$

En este caso, se observa que $L(p, z, t)$ es convexa con respecto a p. Por otro lado, como $F(z, t) \geq 0$ y $\alpha \geq 0$, $L(p, z, t)$ satisface la desigualdad:

$$L(p, z, t) \geq \|p\|^2,$$

y por lo tanto, la condición (Ecuación (7.12)) queda satisfecha con $q = 2$ y $\beta = 0$. Obviamente $\mathcal{A}$ no está vacío ($\mathcal{A}$ contiene funciones lineales simples).

A continuación, se estudia si los mínimos de $I[.]$ son soluciones del sistema Euler-Lagrange asociado. Usando la teoría clásica se necesita mostrar algunas condiciones de crecimiento en $L(p, z, t)$. Primero se observa que, en práctica, la imagen de proyección viene dada por un rectángulo $[0, a] \times [0, b]$ y después las curvas observadas $\Omega(t)$ se incluyen en los rectángulos. En la práctica, el interés está en estimar la función de distancia $distance(\mathcal{P}(z)\tilde{S}(.), \tilde{s}(q))$ cuando la curva corta el rectángulo de la imagen $[0, a] \times [0, b]$. Por lo tanto, sin pérdida de generalidad, se puede cambiar la función de la distancia en la Ecuación (7.9) por

$$\text{distance}_M(x, y) = \begin{cases} \text{distance}(x, y) & si \quad \text{distance}(x, y) \leq M \\ M & si \quad \text{distance}(x, y) > M \end{cases}$$

, donde $M = \sqrt{a^2 + b^2}$. Se define la función de calibración modificada como:

$$F_M(z, t) \equiv \oint_{\Omega(t)} \text{distance}_M(\mathcal{P}(z)\tilde{S}(.), \tilde{s}(q))^2 dq,$$

se puede formular el siguiente resultado:

Teorema 2. (Solución del sistema Euler-Lagrange). Si $\Omega(t)$ se compone por un número finito de curvas cuya longitud está acotada uniformemente en $[t_0, t_1]$, entonces $\mathbf{u}(t) = (P(t), T(t), Z(t))$ satisface:

$$I_M[\mathbf{u}] = \min_{\mathcal{A}} I_M[\mathbf{w}],$$

es una solución débil del sistema:

$$\begin{cases} -P''(t) + \alpha \frac{\partial F_M}{\partial x_1}(\mathbf{u}(t), t) = 0 & en \quad (t_0, t_1) \\ -T''(t) + \alpha \frac{\partial F_M}{\partial x_2}(\mathbf{u}(t), t) = 0 & en \quad (t_0, t_1) \\ -Z''(t) + \alpha \frac{\partial F_M}{\partial x_3}(\mathbf{u}(t), t) = 0 & en \quad (t_0, t_1), \end{cases} \tag{7.13}$$

donde

$$I_M[\mathbf{w}] = \int_{t_0}^{t_1} L_M(D\mathbf{w}(t), \mathbf{w}(t), t)dt,$$

y

$$L_M(p, z, t) = \|p\|^2 + \alpha F_M(z, t). \tag{7.14}$$

Demostración. Para demostrar el resultado, se usa el siguiente resultado clásico presentado en [27] :

(Evans [27] (Pg. 454 THEOREM 7)): Asumir que L verifica las condiciones de crecimiento:

$$\begin{cases} \|L(p, z, x)\| \leq C\left(\|p\|^q + \|z\|^q + 1\right) \\ \|D_p L(p, z, x)\| \leq C\left(\|p\|^{q-1} + \|z\|^{q-1} + 1\right) \\ \|D_z L(p, z, x)\| \leq C\left(\|p\|^{q-1} + \|z\|^{q-1} + 1\right) \end{cases} \tag{7.15}$$

para constantes $C > 0$ y $q > 1$ y $\mathbf{u} \in \mathcal{A}$ satisface:

$$I[\mathbf{u}] = \min_{\mathcal{A}} I[\mathbf{w}].$$

Entonces u es una solución débil (7.13)

En este caso, se aplica el teorema anterior a L_M definido en la Ecuación (7.14). Primero se observa que, como las longitudes de las curvas de $\Omega(t)$ están acotadas uniformemente en $[t_0, t_1]$, y la función distancia$_M(.,.)$ está limitada, entonces la función $F_M(z, t)$ está limitada en $[t_0, t_1]$. Por otro lado los dos primeros componentes del vector z son ángulos, por tanto $F_M(z, t)$ es periódica con respecto a z_x y z_y y tiene derivadas acotadas con respecto a z. Por ello, se puede deducir que existe $C > 0$ tal que

$$\begin{cases} \|L_M(p, z, t)\| \leq C\left(\|p\|^2 + 1\right) \\ \|D_p L_M(p, z, t)\| \leq C(\|p\|) \\ \|D_z L_M(p, z, t)\| \leq C \end{cases}$$

7.5. Resultados experimentales

Para ilustrar el rendimiento del modelo variacional propuesto, se comparan los resultados obtenidos para una secuencia de vídeo usando el minimizado de la función de calibración $F(\mathbf{u}(t), t)$ independientemente para cada instante t y los resultados obtenidos con el modelo variacional propuesto. En las imágenes de las Figuras 7.2, 7.3 y

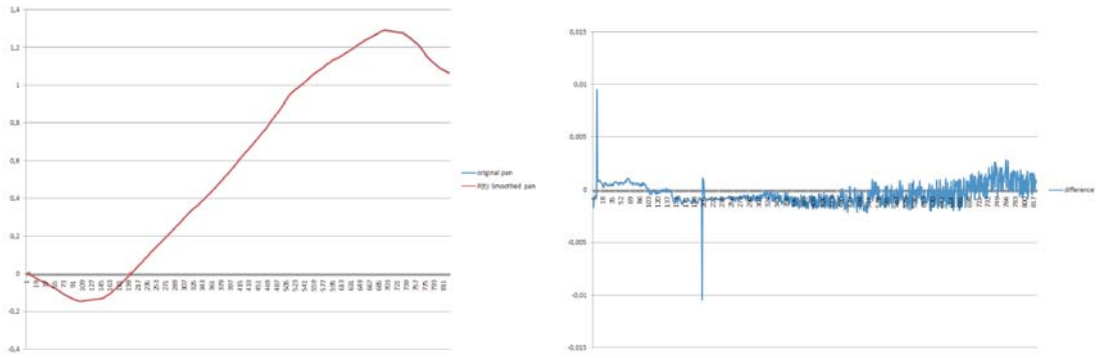

Figura 7.2: $P(t)$ obtenido minimizando $F(\mathbf{u}(t), t)$ independientemente para cada tiempo t y $P(t)$ obtenido minimizando $I[\mathbf{u}]$ (izquierda). Diferencia entre la estimación de $P(t)$ usando ambos métodos (derecha).

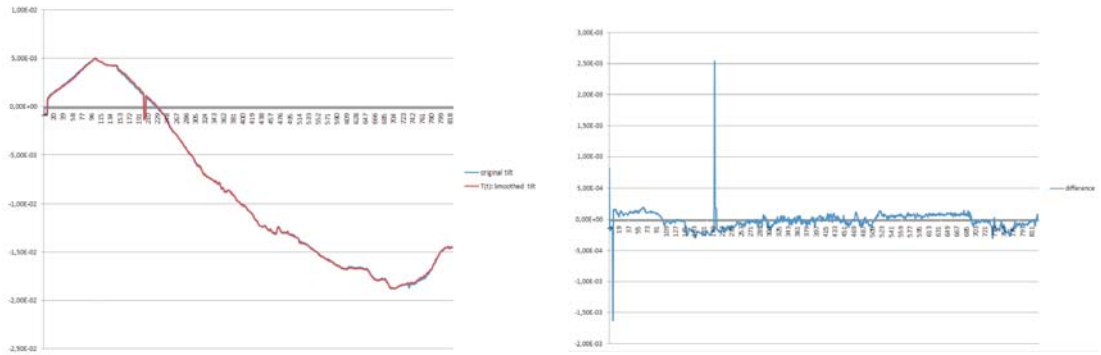

Figura 7.3: $T(t)$ obtenido minimizando $F(\mathbf{u}(t), t)$ independientemente para cada tiempo t y $T(t)$ obtenido minimizando $T[\mathbf{u}]$ (izquierda). Diferencia entre la estimación de $T(t)$ usando ambos métodos (derecha).

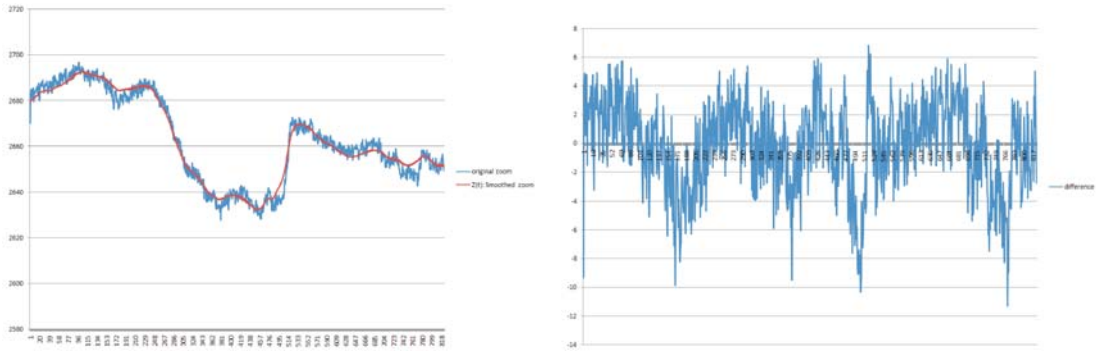

Figura 7.4: $Z(t)$ obtenido minimizando $F(\mathbf{u}(t), t)$ independientemente para cada tiempo t y $Z(t)$ obtenido minimizando $I[\mathbf{u}]$ (izquierda). Diferencia entre la estimación de $Z(t)$ usando ambos métodos (derecha).

7.4, se muestran las gráficas de los $P(t)$, $T(t)$ y $Z(t)$ obtenidos usando ambos métodos así como sus diferencias.

Para inspeccionar visualmente la calidad del método variacional, se presentan unas secuencias de vídeo en http://www.ctim.es/demo102, las cuales se usaron en los experimentos. En ellas se han incluido algunos objetos artificiales usando la información de calibración obtenida (una vez que la cámara está calibrada se pueden incluir fácilmente objetos artificiales en la escena usando técnicas de gráficos por computador como se explicará en el Capítulo 8).

7.6. Conclusiones

En este trabajo se ha estudiado el problema variacional derivado de la calibración de cámaras de vídeo con restricción de suavizado. Se trabajó con cámaras montadas en un trípode. La función de calibración $t \to \mathbf{u}(t) = (P(t), T(t), Z(t))$ se obtiene como el mínimo de una función de energía $I[\mathbf{u}]$. Por ello, se estudió la existencia de un mínimo de dicha función de energía a la vez que las soluciones de las ecuaciones de Euler-Lagrange asociadas. Para comprobar la calidad de la calibración, se insertaron objetos virtuales en una secuencia de vídeo, y se observaron las pequeñas oscilaciones que sufrían los objetos sintéticos. Se pudo apreciar que al aplicar el método de suavizado propuesto, la calibración mejora, las oscilaciones de los objetos virtuales observados se atenúan considerablemente o son eliminadas. Hay que destacar que los resultados obtenidos son muy prometedores y usando la técnica variacional propuesta se reducen notablemente

las oscilaciones que afectan a los gráficos que se incluyen en las secuencias.

Capítulo 8

Inserción de gráficos en escenas reales de escenarios deportivos.

8.1. Introducción

Recientemente, se está incrementando el uso de gráficos virtuales mezclados con el vídeo real en muchas aplicaciones, ver por ejemplo [22]. Estos objetos se pueden usar para mejorar la comprensión de la escena, como pueden ser las banderas de los países superpuestas en la piscina de las competiciones de natación, la línea amarilla de *down* en fútbol americano, la línea de fuera de juego en fútbol o la ruta del disco en hockey sobre hielo. También, dichos elementos virtuales pueden añadirse para mostrar anuncios en diferentes lugares durante el evento sin perturbar al espectador. La mayoría de los métodos de inserción de objetos virtuales en un vídeo están formados por dos fases principales, que son: calibración de la cámara e inserción del contenido virtual. La calibración de la cámara consiste en varias etapas como se ha visto anteriormente en este trabajo (inicialización, estimación, seguimiento de primitivas y mejora de la calibración). La inserción de contenido virtual se divide a su vez en dos fases, sincronización de las cámaras y renderizado.

El método propuesto en este capítulo, se compone de varias etapas para incluir objetos virtuales en el vídeo como se muestra en la Figura 8.1. La fase de inicialización consiste en dos procesos diferentes, inicialización de la calibración de la cámara (descrita en la Sección 4.3.1) y la configuración de los objetos virtuales. La inicialización de la calibración de la cámara, se divide en tres pasos, que son ejecutados solamente en el primer fotograma: lectura de información calculada previamente (como los parámetros

geométricos del trípode y las clases de entrenamiento del árbol de decisión), detección de primitivas y la calibración de la cámara para el primer fotograma. La detección de primitivas se realiza con un método morfológico descrito en el Capítulo 2, mientras que la técnica de calibración de la cámara está descrita en el Capítulo 3. Por otro lado, la configuración de los elementos virtuales consiste en definir su apariencia y su posición en el escenario, en este caso en el estadio de fútbol. Para los siguientes fotogramas, se comienza directamente en la fase de estimación de la calibración y se considera la información obtenida de los dos fotogramas previos. Se asume que los cambios entre fotogramas consecutivos no son muy grandes, y además, como se trata de cámaras colocadas en un trípode, los movimientos de la cámara están restringidos. Después, se continúa el proceso con el seguimiento de primitivas descrito en el Capítulo 5, el cual busca las primitivas en la imagen usando un árbol de decisión y la proyección de las primitivas de referencia. Las primitivas que se usan son las líneas y círculos en un modelo de un campo de fútbol con dimensiones reales, dichas primitivas son proyectadas usando la homografía estimada de los dos fotogramas previos. Finalmente, con las primitivas detectadas y la geometría del trípode, se mejora la calibración del fotograma actual. Cuando se tiene la cámara calibrada en un fotograma, se puede sincronizar la cámara real con una cámara virtual y proyectar los objetos virtuales en la imagen real usando OpenGL [71]. Se usa OpenGL porque proporciona funciones para manejar fácilmente cámaras y objetos virtuales. Por ejemplo cambiando el punto de vista o la posición de la cámara virtual, añadir texturas y transparencias al contenido virtual, etc.

8.1.1. Contribución de este capítulo

Realidad aumentada en escenarios deportivos usando cámaras montadas en un trípode:
En esta aportación, se aborda el problema de insertar contenido virtual en una secuencia de vídeo. El método que se propone usa solamente información de la imagen. En él se realiza un seguimiento de primitivas, calibración de cámaras, sincronización de cámara real y virtual, y por último el renderizado para insertar los gráficos virtuales en el vídeo real. El hecho de que se trabaje con cámaras montadas en un trípode simplifica la calibración, véase [14]. El procedimiento de seguimiento de primitivas usa líneas y círculos y se realiza por medio de un *CART*, ver [11]. Finalmente, la sincronización de la cámara real y la virtual, y el renderizado se llevan a cabo usando funciones de OpenGL *(Open Graphic Library)*. Para ilustrar el rendimiento del método se ha aplicado a vídeos de alta definición. La calidad del método propuesto se ha validado insertando elementos virtuales en dichos vídeos.

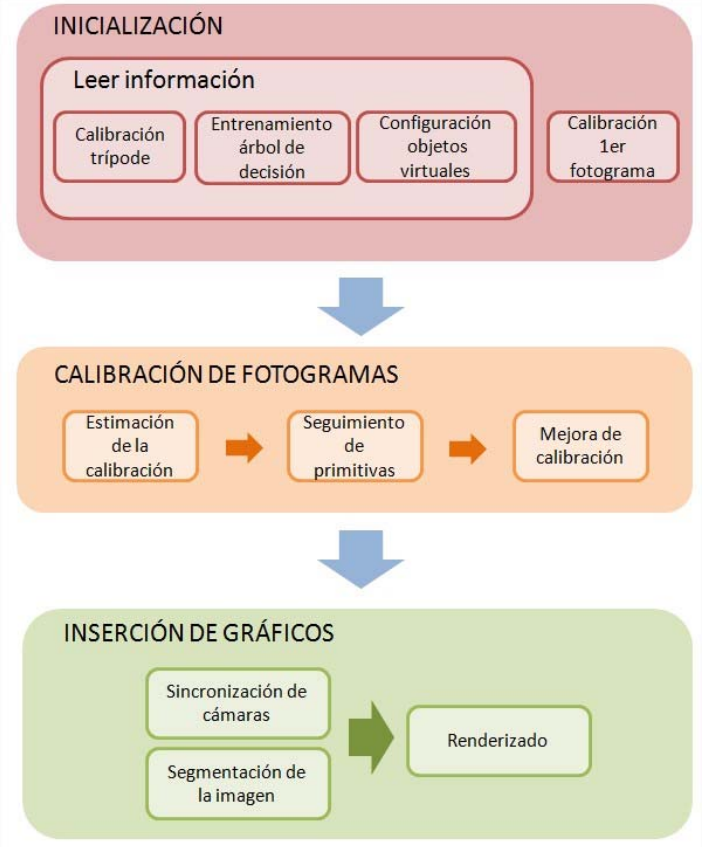

Figura 8.1: Etapas de la implementación para insertar contenido virtual en una secuencia de video.

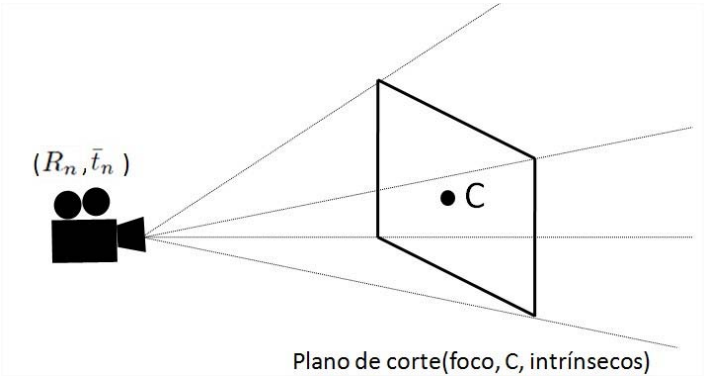

Figura 8.2: Cámara real. Donde R_n es la rotación, $\bar{t}_n$ es la traslación y C es el centro de la imagen, los componentes del centro son x_c y y_c que forman parte de los parámetros intrínsecos de la Ecuación (7.6)

8.1.2. Sincronización de cámaras

El primer paso, es usar OpenGL para crear un mundo 3D virtual el cual se mezclará con la imagen del mundo real. Para poder insertar objetos en la imagen real con la misma perspectiva, se necesita sincronizar la cámara virtual con la cámara real. Esto significa que hay que situar la cámara virtual en la misma posición que la cámara real y con la misma configuración de rotación y *zoom*. La sincronización se hace calculando los parámetros de la cámara virtual a partir de la cámara real. Los parámetros que definen una cámara real son rotación, traslación y plano de corte, como se puede observar en la Figura 8.2.

El plano de corte está definido por el foco, el centro y los parámetros intrínsecos de la cámara. Para realizar la sincronización de la cámara, se necesita configurar la cámara virtual con los parámetros de la cámara real. OpenGL tiene funciones que implementan este proceso, pero necesitan algunas entradas que deben ser calculadas previamente. Dichas entradas son: el centro de la cámara, el centro de proyección y un vector indicando la dirección del eje vertical de la cámara (VUP). Además, se necesita definir el volumen de vista, el cual determina cómo se proyecta un objeto 3D en una imagen 2D. Para una proyección en perspectiva, el volumen de vista es el denominado *frustum*. Para determinar el *frustum* en OpenGL se necesitan las distancias desde el centro de proyección a los planos de corte (izquierda, derecha, arriba y abajo) y las distancias desde la cámara a los planos de corte *near* y *far*, como se muestra en la Figura 8.3.

Se usa la calibración de la cámara Euclídea llevada a cabo en la etapa anterior para

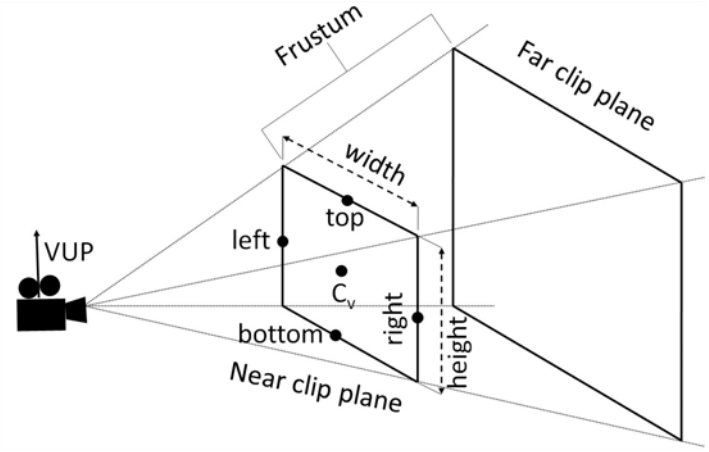

Figura 8.3: Cámara virtual. Donde *VUP* es el vector que indica el eje vertical de la cámara, *width* y *height* son las dimensiones de la imagen real. C_v es el punto principal. Los puntos *top*,*left*,*right* y *bottom* son puntos conocidos para definir los planos de corte.

obtener todos los parámetros requeridos. Primero, se calcula la matriz de proyección inversa de la cámara Euclídea, siendo la matriz de proyección como se muestra en la Ecuación (8.1).

$$P_n\left(f_n, \theta_0^n, \theta_1^n\right) \equiv A\left(f_n\right) R_0 R\left(\theta_0^n, \theta_1^n\right) \left\lfloor Id, R^T\left(\theta_0^n, \theta_1^n\right)\left(\bar{t}\left(\theta_0^n, \theta_1^n\right) - \bar{c}^0\right)\right\rfloor . \tag{8.1}$$

La matriz de proyección inversa se obtiene como sigue:

$$P^{-1} \equiv R_0^T A^{-1}\left(f_0\right) \left\lfloor Id, \bar{c}^0\right\rfloor . \tag{8.2}$$

Ahora, se puede calcular el punto principal multiplicando esta matriz por el centro de la imagen, $C_v = P^{-1}C$, el cual pertenece a los parámetros intrínsecos de la cámara Euclídea.

Después, hay que definir los planos de corte del *frustum* como se muestra en la Figura 8.3. Para calcularlos, se obtienen las distancias desde el punto principal a los lados del plano de corte cercano. Se sabe que las dimensiones del plano de corte cercano son las dimensiones de la imagen real. Primero, se obtienen cuatro puntos, uno por cada plano de corte. Estos puntos son *top*, *right*, *left*, y *bottom*, esto se puede ver en la Figura 8.3. Estos puntos se definen usando las dimensiones del plano de corte cercano y las coordenadas del centro de la imagen $C = (x_c, y_c)$, donde x_c y y_c se extraen de los parámetros intrínsecos (Ecuación (7.6)). Luego, se multiplican por la matriz de

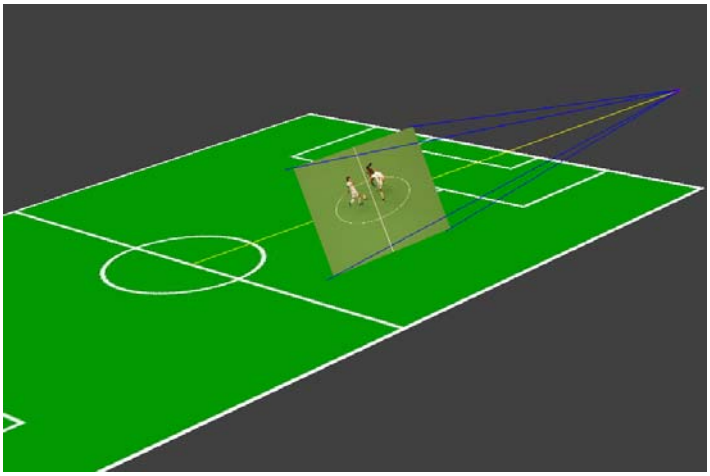

Figura 8.4: Cámara virtual sincronizada. La imagen real está en el plano cercano, donde se renderizan los objetos virtuales.

proyección inversa:

$$top = P^{-1}\left(x_c, height - 1, 1, 1\right)^T, \tag{8.3}$$

$$bottom = P^{-1}\left(x_c, 0, 1, 1\right)^T, \tag{8.4}$$

$$right = P^{-1}\left(width - 1, y_c, 1, 1\right)^T, \tag{8.5}$$

$$left = P^{-1}\left(0, y_c, 1, 1\right)^T. \tag{8.6}$$

Seguidamente hay que calcular las distancias entre el punto principal y los puntos calculados previamente para que estén disponibles para OpenGL como parámetros y definir el *frustum*. Finalmente, se obtiene el vector VUP proyectando el vector que va desde el punto principal al lado superior del plano *near*:

$$VUP = P^{-1}\left(x_c, 0, 1, 1\right)^T - \left(2C - C_v\right). \tag{8.7}$$

Como resultado de la sincronización, la cámara virtual es capaz de obtener imágenes con la misma perspectiva que la cámara real, como se puede observar en la Figura 8.4, la cual muestra una cámara virtual sincronizada con una cámara real.

8.2. Renderizado

En la fase de inicialización, se configuraron en el mundo virtual diferentes polígonos colocados en el césped y objetos 3D. Para realizar un renderizado correcto hay que tener en cuenta el modelo de distorsión de la lente. Aplicando un modelo de distorsión inverso a los puntos de los polígonos, estos encajarán mejor con la imagen real distorsionada. Cuando se han sincronizado las cámaras, ya se puede renderizar el fotograma resultante añadiendo el contenido virtual a la imagen real. Por ejemplo, si en la Figura 8.4 se añaden objetos en el círculo central, con OpenGL se pueden proyectar encima de la imagen real. Pero primero, hay que hacer una segmentación de la imagen real. Este segmentación permite diferenciar el césped de los jugadores y las líneas. Esto es útil para los anuncios virtuales que se colocan sobre el césped. Así, solamente se reemplazarán los píxeles que pertenezcan al césped, evitando la oclusión de jugadores o líneas. La segmentación se realiza convirtiendo primero la imagen al espacio *HSV*. Después se calcula el histograma del canal H y se obtiene el valor máximo. En este caso concreto, el valor máximo será el verde del césped, ya que es el color dominante en una escena de un partido de fútbol. Pero también es común encontrar diferentes tonos de verde en el césped, ya sea por la iluminación o por el corte del césped. Para resolver este problema, se usan unos umbrales para seleccionar valores de H que pertenezcan al césped. La máscara obtenida después de segmentar, se almacena en el canal alfa de la imagen original, este canal es usado por OpenGL para saber qué píxeles son transparentes. Para finalizar el proceso, se utiliza la imagen original como fondo del mundo virtual, y OpenGL reemplaza los píxeles transparentes con píxeles de objetos virtuales proyectados en el plano de la imagen real.

8.3. Experimentos y resultados

Este método se ha probado en los dos tipos de secuencias usadas en este trabajo de tesis: las secuencias del modelo a escala (1440×809) y las escenas de fútbol real (1920×1080). Antes de aplicar la fase de inicialización a la secuencia, hay que obtener cierta información para realizar la calibración, como se ha comentado en capítulos anteriores. Por ejemplo, el conjunto de datos de entrenamiento del árbol de decisión (Capítulo 5) y la calibración del trípode (Capítulo 4). Con esa información se computa la fase de inicialización y se calibra la secuencia completa. La calibración obtenida se usa para la sincronización de la cámara real con la virtual y el posterior renderizado de gráficos en el vídeo. Los resultados se muestran en las imágenes de la Figura 8.5, extraídas de los videos proporcionados como material adicional en http://www.ctim.es/demo104/.

Figura 8.5: Inserción de gráficos con calibración de cámaras usando la geometría del trípode y seguimiento de primitivas. Extraída de los videos proporcionados como material suplementario (http://www.ctim.es/demo104/).

8.4. Conclusiones

En este capítulo se ha estudiado la inserción de elementos virtuales en vídeos de escenarios deportivos usando cámaras montadas en un trípode. Para realizar dicha inserción, primero se ha tenido que calibrar la cámara usando los procedimientos descritos en el Capítulo 4. Después, para sincronizar las cámaras virtual y real, se hace una correspondencia entre ambas, calculando los parámetros de la cámara virtual a partir de la cámara real. Por último, se renderiza el resultado usando OpenGL porque proporciona un manejo fácil de cámaras virtuales y un procesado de gráficos optimizados en la tarjeta gráfica. Se presentaron algunos experimentos con vídeos de alta definición de eventos deportivos, en ellos se insertaron varios elementos virtuales.

Capítulo 9

Conclusiones y trabajo futuro

9.1. Conclusiones

El objetivo de esta tesis es el desarrollo de métodos de procesado de vídeo digital. En los trabajos expuestos en este documento se ha intentado contribuir a la literatura en varios procedimientos importantes del procesado de vídeo digital, como son la calibración de cámaras, la detección de primitivas, corrección de la distorsión de lentes e inserción de gráficos en secuencias de vídeo.

El primer trabajo presentado fue una nueva técnica para la localización de primitivas y de sus centros, basada en operadores morfológicos. El método propuesto se probó en situaciones reales y funciona correctamente incluso en escenarios con imágenes entrelazadas o con cambios de iluminación. En los experimentos se observó que los centros de las primitivas de interés se extraen con precisión. También cabe destacar que la cantidad de líneas falsas que se detectan es muy pequeña. Pudiendo eliminarse fácilmente en una fase de postprocesado donde se buscan líneas rectas y elipses en la imagen, basándose en el conjunto de centros de primitivas extraído.

Una de las partes importantes del trabajo, fue la propuesta de métodos para la calibración de cámaras. Primero, se estudió la calibración de cámaras aisladas, y luego se trabajó con métodos basados en las características de un trípode y las propiedades de continuidad del movimiento de la cámara en una secuencia de vídeo.

Para el problema de calibración de cámaras aisladas, el trabajo se centró en escenarios reales donde los objetos de interés están en un plano. En estos escenarios se pueden

encontrar ciertas complicaciones a la hora de calibrar, ya que existen situaciones donde hay pocas primitivas visibles que puedan considerarse para calcular la posición de la cámara o cambios de iluminación que dificultan la detección de primitivas. Con el método propuesto, se demostró que si hay un número mínimo de primitivas visibles en la escena, se puede calcular la transformación del plano de la imagen al plano de referencia. El método está basado en la construcción de homografías candidatas usando la colección de líneas que se extraen de la imagen. Luego hay que encontrar la homografía que minimiza la función de error. A partir de esta homografía se pueden recuperar los parámetros de la cámara, tanto la distancia focal como los parámetros extrínsecos.

Como extensión del trabajo anterior, se estudió el problema de calibración de cámaras de vídeo. Se desarrolló un método para la calibración de cámaras de vídeo en escenarios donde la cámara está montada en un trípode. Esta situación es muy común en la práctica, sobre todo para retransmisiones deportivas en televisión. Se estudió la geometría del trípode desde un punto de vista matemático, y sus características simplificaron enormemente el problema y permitieron calibrar fotogramas que las técnicas habituales no son capaces de calibrar. En el modelo propuesto, una de las principales novedades es el hecho de que no se supone que el centro de rotación del trípode y el centro de proyección de la cámara sean el mismo punto. Esto es importante dado que en las cámaras profesionales de gran tamaño, la distancia entre el centro de rotación del trípode y el centro de proyección de la cámara es significativa.

Al trabajar con secuencias de vídeo, se pueden aprovechar ciertas características del mismo para mejorar la calibración de la cámara. Para ello, se ha presentado un método de seguimiento del movimiento de la cámara en secuencias de vídeo. El procedimiento está basado en la estimación de la homografía y en el seguimiento de primitivas. Para dicho proceso de seguimiento, se realizó una nueva contribución basada en un árbol de decisión *CART (Classification and Regression Tree)*. El árbol se construye mediante un proceso de aprendizaje que utiliza un conjunto de entrenamiento creado a partir de la segmentación de un fotograma. Se realizaron experimentos en vídeos de alta definición, donde el procedimiento propuesto demostró ser rápido y preciso en la segmentación de las imágenes (diferenciando las primitivas del fondo). El error de clasificación máximo fue de un 0,16 % de los píxeles. Por otra parte, la combinación del método de seguimiento con la estimación de la homografía mejora el tiempo de procesado sin perder precisión en los resultados de la calibración. Para probar la eficiencia de esta estrategia, se llevaron a cabo varios experimentos en escenarios reales, concretamente en retransmisiones en alta definición de partidos de fútbol. Los resultados obtenidos son precisos y muy prometedores. El tiempo medio de procesado de un fotograma de alta definición es de 5 milisegundos. En términos de complejidad computacional, las principales novedades son que la computación de un árbol de decisión es muy rápida y el método de seguimiento es local. Es decir, no hace falta procesar todos los píxeles de la imagen, ya

que sólo se necesita analizar una vecindad de píxeles alrededor de la localización de las primitivas, de acuerdo con una estimación de su posición deducida de las posiciones de dichas primitivas en frames anteriores.

Una cuestión a tener en cuenta en el procesado de vídeo digital, es la distorsión de la lente, ya que durante la secuencia de vídeo, el modelo de distorsión puede variar. Dentro de esta tesis, se han desarrollado nuevos modelos matemáticos que contemplan la variación de la distorsión de lentes en cámaras con *zoom*. Tales modelos están basados en una aproximación polinómica de segundo orden que tiene en cuenta la variación de los parámetros de distorsión radial a lo largo del rango del *zoom* y, la minimización de la energía del error global. Dicho error se calcula midiendo la distancia entre secuencias de puntos alineados distorsionados y líneas rectas después de la corrección de la distorsión. La calidad de la corrección que ofrece el modelo propuesto, es tan buena como la obtenida por los métodos que corrigen fotograma a fotograma. Esto es destacable porque usando solamente 6 parámetros (3 para el polinomio asociado al primer coeficiente de distorsión k_1 y 3 parámetros para el segundo coeficiente k_2) se puede estimar el modelo de distorsión para cualquier distancia focal efectiva de la lente. Se experimentó con dos secuencias de vídeo diferentes para estimar el modelo de distorsión dependiente del *zoom*, dichas secuencias son un vídeo de un patrón de calibración y un vídeo real de fútbol grabado con una vídeo cámara profesional. Los resultados para ambos casos muestran la potencialidad del nuevo modelo.

Al calibrar una secuencia de vídeo, pequeñas perturbaciones sobre los valores de $P(t)$, $T(t)$ y $Z(t)$ producen pequeñas oscilaciones en el movimiento de la cámara, pudiendo ser percibidas por el espectador. Para corregir dichas perturbaciones, se ha añadido un nueva aproximación variacional para suavizar el movimiento de la cámara durante la secuencia de vídeo. Haciendo uso de este método se consigue atenuar y eliminar la mayoría de las vibraciones que puedan detectarse al incluir gráficos en la secuencia calibrada.

Como aplicación de todos los métodos propuestos, se abordó la inserción de elementos virtuales en vídeos de escenarios deportivos usando cámaras montadas en un trípode. Se eligió este tipo de aplicación porque requiere una calibración rápida y precisa. En el procedimiento que se propone en este trabajo, después de haber calibrado la cámara, se procede a sincronizar la cámara virtual y la real. Para ello, se hace una correspondencia entre ambas, calculando los parámetros de la cámara virtual a partir de la cámara real. Luego, se renderiza el resultado usando OpenGL porque proporciona un manejo fácil de cámaras virtuales y un procesado de gráficos optimizado en la tarjeta gráfica. Se presentaron algunos experimentos con los vídeos de alta definición de eventos deportivos que se han usado en el resto de trabajos de la tesis. A los vídeos se les añadieron varios elementos virtuales, tanto en el césped como a diferentes alturas

sobre el campo.

9.2. Trabajo futuro

El desarrollo de esta tesis da lugar a un amplio abanico de posibilidades de trabajo futuro:

Automatización de la segmentación para el aprendizaje del *CART*. En el método propuesto de seguimiento de primitivas, el entrenamiento del árbol de decisión se hace a partir de una segmentación del primer fotograma, en la que se diferencian dos clases: primitivas y césped. Esta segmentación podría hacerse de forma automática usando una parte del método [6] para detectar las líneas. Además, podría usarse para actualizar las clases durante el procesado del vídeo y volver a entrenar el árbol de decisión, ya que en una retransmisión larga, las condiciones de iluminación podrían cambiar.

Sustitución de elementos del vídeo. Además de la aplicación de inserción de gráficos, los métodos de calibración y detección que se han planteado, proporcionan información que podría ser útil para sustituir elementos del vídeo real por elementos sintéticos o simplemente eliminarlos del mismo.

Mejora del rendimiento de algunos métodos. Sería interesante poder mejorar el rendimiento de algunos métodos con el fin de añadirlos al procedimiento de calibración de cámaras en tiempo real que se desarrolló en este trabajo:

- *Algoritmo de desentrelazado simple:* La mayoría de las veces es necesario desentralazar el vídeo, con lo cual es importante en un proceso de tiempo real que esta parte sea lo más rápida posible.
- *Detección de centros de primitivas:* Este método propuesto ([6]), necesita procesar todos los píxeles de la imagen, y al trabajar con fotogramas de alta definición puede tardar algunos segundos en ejecutarse. Podría ser interesante optimizar este algoritmo para ejecutarlo en la GPU. De esta forma podría usarse en la calibración de vídeo en tiempo real, para poder detectar automáticamente cuando se produzcan cambios de iluminación y poder actualizar el árbol de decisión durante el procesado del vídeo.
- *Corrección de la distorsión de lentes:* El proceso tiene un coste elevado, así que en el procesado en tiempo real, se optó por usar el mismo modelo de distorsión

para toda la secuencia. Pero como se ha visto en [8], en los casos reales el modelo de distorsión cambia al variar el *zoom*. Haciendo que el método propuesto en [8] se ejecute más rápido, se conseguirían resultados más precisos en la calibración en tiempo real.

- *Inserción de gráficos:* Supone un gran reto conseguir que la combinación de la calibración de cámaras y la inserción de elementos virtuales ofrezcan resultados en tiempo real. Esto sería muy útil para las retransmisiones en directo. Los resultados obtenidos en esta tesis acercan la consecución de este objetivo, por ello, es muy interesante seguir trabajando en esta línea para acelerar la ejecución de todos los procedimientos implicados en esta aplicación.

Bibliografía

[1] L. Alvarez and J. Esclarin : Image quantization using reaction-diffusion equations. SIAM Journal on Applied Mathematics, 57(1) (1997) 153–175.

[2] L. Agapito, E. Hayman and I. Reid: Self-calibration of rotating and zooming cameras. International Journal of Computer Vision, 45 (2001) 107–127.

[3] R. Atienza, A. Zelinsky: A practical zoom camera calibration technique: an application on active vision for human-robot interaction. In Proceedings of Australian Conference on Robotics and Automation, Sydney (Australia). (2001) 85–90 .

[4] L. Alvarez, J. Esclarín , A. Trujillo: A model based edge location with subpixel precision. Proceedings IWCVIA 03: International WorkShop on Computer Vision and Image Analysis, Las Palmas de Gran Canaria (SPAIN). (2003) 29–32.

[5] L. Alvarez, L. Gomez and J.R. Sendra: An algebraic approach to lens distortion by line rectification. Journal of Mathematical Imaging and Vision, 35 (2009) 36–50.

[6] M. Aleman-Flores and L. Alvarez and P. Henriquez and L. Mazorra: Morphological thick line center detection. In: 7th International Conference on Image Analysis and Recognition, 6111 (2010) 71–80.

[7] L. Alvarez and V. Caselles. Calibration method for a TV and video camera. European Patent 09380137.1. Issued February 2011.

[8] L. Alvarez, L. Gomez, P. Henriquez: Zoom dependent lens distortion mathematical models. Journal of Mathematical Imaging and Vision, 44(3) (2012) 480–490.

[9] L. Alvarez, L. Gomez, P. Henriquez, L. Mazorra: A variational approach to camera motion smoothing. In: Differential Equations and Applications - DEA, 4 (2011) 555–564.

[10] L. Alvarez, L. Gomez and J.R. Sendra: Accurate Depth dependent lens distortion models: An Application to Planar View Scenarios. Journal of Mathematical Imaging and Vision, 39 (2011) 75–85.

[11] L. Alvarez, P. Henriquez, J. Sanchez: CART application to image primitive tracking. CAEPIA 11: Conferencia de la Asociacion Española para la Inteligencia Artificial. (2011).

[12] M. Alemán-Flores, L. Alvarez, P. Henriquez, A. Trujillo: Augmented reality in sport scenarios using cameras mounted on a tripod. Technical Report - Centro de Tecnologías de la Imagen, (2012).

[13] L. Alvarez, L. Gomez, P. Henriquez, L. Mazorra: Automatic camera pose recognition in planar view scenarios. 17th Iberoamerican Congress on Pattern Recognition, 7441 (2012) 406–413.

[14] L. Alvarez, P. Henriquez, L. Mazorra: Mathematical models for the calibration of cameras mounted on a tripod using primitive tracking. In: 9th International Conference on Image Analysis and Recognition, 7324 (2012) 304–311.

[15] L. Breiman, JH. Friedman, RA. Olshen and CJ. Stone: Classification and Regression Trees. Belmont, CA: Wadsworth, 1984.

[16] S.Benhimane, E. Malis: Self-calibration of the distortion of a zooming camera by matching points at different resolutions. In Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems, Sendai(Japan). (2004) 2307–2312.

[17] C. Bräuer-Burchardt, M. Heinze, C. Munkelt, P. Kühmstedt, G. Notni: Distance dependent lens distortion variation in 3D measuring systems using fringe projection. In BMVC 2006, (2006) 327–336.

[18] V. Babaee-Kashany and H. Reza Pourreza: Camera pose estimation in soccer scenes based on vanishing points. 9th IEEE International Symposium on Haptic Audio-Visual Environments and Games, (2010) 157–162.

[19] T. Battikh, I. Jabri: Camera calibration using court models for real-time augmenting soccer scenes. In: Multimedia Tools Applications, 51 (2011) 997–1011.

[20] B. Jiang, L. Songyang and B. Liang: Automatic line mark recognition and its application in camera calibration in soccer video. IEEE International Conference on Multimedia and Expo (ICME), (1–6) (2011).

[21] B. Chapman, G. Jost, R. van der Pas: Using OpenMP: Portable Shared Memory Parallel Programming. 2007.

[22] J. Chandaria, G. Thomas, D. Stricker: The MATRIS project: real-time markerless camera tracking for Augmented Reality and broadcast applications. Journal of Real-Time Image Processing, 2(2–3) (2007) 69–79.

[23] C. Chang, K. Hsieh, M. Chiang, J. Wu: Virtual spotlighted advertising for tennis videos. In: Journal of visual communication and image representation, 21 (2010) 595–612.

[24] J.I. Díaz: Nonlinear partial differential equations and free boundaries. Vol. I, Elliptic equations. Pitman Advanced Publishing Program (N. 106), 1985.

[25] F. Devernay, O. Faugeras: Straight lines have to be straight. Machine Vision and Applications. 13(1) (2001) 14–24.

[26] D. Peña: Análisis de datos multivariantes. Madrid, 2002.

[27] L.E. Evans: Partial Differential Equations. AMS, 1985.

[28] A. Ekin, A. M. Tekalp, R. Mehrotra: Automatic soccer video analysis and summarization. In: IEEE Transactions on Image Processing, 12(7) (2003).

[29] M. Emre Celebi and H. Iyatomi and W. V. Stoecker and R. H. Mossd and H. S. Rabinovitz and G. Argenziano and H. P. Soyer: Automatic detection of blue white veil and related structures in dermoscopy images. In: Computerized Medical Imaging and Graphics, 32 (2008) 670–677.

[30] B. Ergum: Photogrammetric observing the variation of intrinsic parameters for zoom lenses. Scientific Research and Essays. 5(5) (2010) 461–467.

[31] M. Fischler, R. Bolles: Random Sample Consensus: A Paradigm for model fitting with applications to image analysis and automated cartography. In: Comm. of the ACM, 24 (1981).

[32] C. S. Fraser, M. R. Shortis: Variation of distortion within the photographic field. Photogrammetric Engineering Remote Sensing. 58(6) (1992) 851–855.

[33] O. Faugeras: Three-Dimensional Computer Vision. MIT Press, 1993.

[34] J. Fayman, O. Sudarsky, E. Rivlin: Zoom tracking. Proceedings of the International Conference on Robotics and Automation, Leuven (Belgium). (1998) 2783–2788.

[35] O. Faugeras, Q-T. Luong, T. Papadopoulo: The Geometry of multiple images. MIT Press, 2001.

[36] D. Farin and S. Krabbe and PHN. de With and W. Effelsberg: Robust camera calibration for sport videos using court models. In: Storage and Retrieval Methods and Applications for Multimedia, 5307 (2004) 80–91.

[37] D. Farin and J.G. Han and PHN. de With: Fast camera calibration for the analysis of sport sequences. In: IEEE International Conference on Multimedia and Expo (ICME), (1–2) (2005) 482–485.

[38] C. Fraser, S. Al-Ajlouni: Zoom-dependent camera calibration in digital close-range photogrammetry. Photogrammetric Engineering Remote Sensing. 72(9) (2006) 1017–1026.

[39] L.A.F. Fernandes, M. M. Oliveira: Real-time line detection through an improved hough transform voting scheme. In: Pattern Recognition, 41(1) (2008).

[40] C. Fahn, C. Lo: A high-definition human face tracking system using the fusion of omni-directional and PTZ cameras mounted on a mobile robot. 5th IEEE Conference on Industrial Electronics and Applications (ICIEA), Taichung (China), (6–11) (2010).

[41] M. Giaquinta, S. Hildebrandt: Calculus of Variations, Vol 1-2. Springer, 1996.

[42] G. Thomas: Real-time camera tracking using sports pitch markings. Journal of Real-Time Image Processing, 2(2–3) (2007) (117–132).

[43] R. Hartley, A. Zisserman: Multiple View Geometry in Computer Vision. Cambridge University Press, 2000.

[44] T. Hastie, R. Tibshirani, Jerome Friedman: The Elements of Statistical Learning. Canada, 2001.

[45] E. Hayman and D. Murray: The effects of translational misalignment when self-calibration rotating and zooming cameras. IEEE Transactions on Pattern Analysis and Machine Intelligence, 25(8) (2003) 1015–1020.

[46] A. Hampapur, L.Brown, J. Connell, et al.: Smart video surveillance: exploring the concept of multiscale spatiotemporal tracking. IEEE Signal Processing Magazine. 22(2) (2005) 38–51.

[47] JB. Hayet and J. Piater: On-Line rectification of sport sequences with moving cameras. In: MICAI 2007: Advances in Artificial Intelligence, 4827 (2007) 736–746.

[48] J Han, D. Farin, P.H.N. de With: A mixed-reality system for broadcasting sports video to mobile devices. In: IEEE Multimedia, 18 (2011) (72–84).

[49] M. Irani, P. Anandan: A unified approach to moving object detection in 2D and 3D scenes. IEEE Transactions on Pattern Analysis and Machine Intelligence. 20(6) (1998) 577–589.

[50] N. Inamoto, H. Saito: Free viewpoint video synthesis and presentation from multiple sporting videos. In: Electronics and Communications in Japan, 90 (2007) 1693–1701.

[51] J. Gonzalez: Visión por Computador. 2000.

[52] H. Kim and KS. Hong: Robust image mosaicing of soccer videos using self-calibration and line tracking. In: Pattern analysis and applications, 4 (2001) 9–19.

[53] J. Knight, A. Zisserman, I. Reid: Linear auto-calibration for ground plane motion. In: 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1 (2003) 503–510.

[54] S.H. Khatoonabadi, M. Rahmati: Automatic soccer players tracking in goal scenes by camera motion elimination. In: Image and Vision Computing 27 (2009) 469–479.

[55] D. Kim, H. Shin, J. Oh, K. Sohn: Automatic radial distortion correction in zoom lens video camera. Journal of Electronic Imaging. 19(4) (2010) 43010–43017.

[56] D. L. Light: The new camera calibration system at the U.S. geological survey. Photogrammetric Engineering & Remote Sensing. 58(2) (1992) 185–188.

[57] M. Li, J. Lavest: Some aspects of zoom lens camera calibration. IEEE Transactions on Pattern Analysis and Machine Intelligence. 18(11) (1996) 1105–1110.

[58] Y. Liu, Q. Huang, Q. Ye, W. Gao: A new method to calculate the camera focusing area and player position on playfield in soccer video. In: Visual Communications and Image Processing 2005.

[59] Q. Li and Y. Luo: Automatic camera calibration for images of soccer match. Proceedings of World Academy of Science, Engineering and Technology, 1 (2005) 170–173.

[60] V. Lepetit, P. Fua: Keypoint recognition using randomized trees. In: IEEE Transactions on Pattern Analysis and Machine Intelligence, 28(9) (2006) 1465–1479.

[61] H. Li and C. Shen: An LMI approach for reliable PTZ camera self-calibration. IEEE International Conference on Video and Signal Based Surveillance 2006.

[62] S. Li, B. Lu: Automatic camera calibration technique and its application in virtual advertisement insertion system. In: 2nd IEEE Conference on Industrial Electronics and Applications. ICIEA 2007, 1 (2007) 288–292.

[63] J. Liu, X. Tong, W. Li, T. Wang, Y. Zhang, H. Wang: Automatic player detection, labeling and tracking in broadcast soccer video. In: Pattern Recognition Letters 30 (2009) 103–113.

[64] E. Martinez, C. Torras: Contour-based 3D motion recovery while zooming. Robotics and Autonomous Systems. 44(3-4) (2003) 219–227.

[65] G. Macchiavello, G. Moser, G. Boni, S. B. Serpico: Automatic unsupervised classification of snow-covered areas by decision-tree classification and minimun error thresholding. In: IEEE International Geoscience and Remote Sensing Symposium, 1–5 (2009) 1251–1254.

[66] M. Ozuysal, M. Calonder, V. Lepetit, P. Fua: Fast keypoint recognition using random ferns. In: IEEE Transactions on Pattern Analysis and Machine Intelligence, 2010.

[67] V. Peddigari, N. Kehtarnavaz: A relational approach to zoom tracking for digital still cameras. IEEE Transactions on Consumer Electronics. 51(4) (2005) 1051–1059 .

[68] R. Hartley: Self-calibration from multiple views with a rotating camera. European Conference on Computer Vision, 800 (1994) 471–478.

[69] J. Serra: Image Analysis and Mathematical Morphology. Academic Press, 1982.

[70] L. Shapiro, G. Stockman: Computer Vision. 2000.

[71] D.Shreiner, M. Woo, J. Neider, T. Davis: OpenGL Programming Guide. 2007.

[72] T.H. Eagles: Constructive geometry of plane curves. Macmillan and co. 1885.

[73] R. Y. Tsai: A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf tv cameras and lenses. IEEE Journal of Robotics and Automation. 3(4) (1987) 323–344.

[74] K. Tarabanis, R. Tsai, D. Goodman: Modeling of a computer-controlled zoom lens. In Proceedings of IEEE International Conference on Robotics and Automation. 2 (1992) 1545–1551.

[75] G.A. Thomas, J. Jin, T. Niblett, C. Urquhart: A versatile camera position measurement system for virtual reality TV production. In Proceedings of IBC 97, (1997) 284–289.

[76] J. Weng, P. Cohen, M. Herniou: Camera calibration with distortion models and accuracy evaluation. IEEE Transactions on Pattern Analysis and Machine Intelligence. 14(10) (1992) 965–980.

[77] R. Wilson, S. Shafer: A perspective projection camera model for zoom lenses. Proc. Second Conference on Optical 3-D Measurements Techniques, Switzerland, (1993).

[78] K. Wan, X. Yan, X. Yu, C. Xu: Real-Time Goal-Mouth Detection In MPEG Soccer Video. In: MM '03, (2003).

[79] Y. Watanabe and M. Haseyama and H. Kitajima: A soccer field tracking method with wire frame model from TV images. In: 2004 International Conference on Image Processing, (1–5) (2004) 1633–1636.

[80] L. Wang, B. Zeng, S. Lin, G. Xu, H. Shum: Automatic extraction of semantic colors in sports video. In: ICASSP 2004.

[81] F. Wang, L. Sun, B. Yang, S. Yang: Fast arc detection algorithm for play field registration in soccer video mining. In: IEEE International Conference on Systems, Man and Cybernetics, 6 (2006) 4932–4936.

[82] K. Wan, X. Yan: Advertising insertion in sports webcasts. In: IEEE Multimedia, 14 (2007) 78–82.

[83] A. Watve, S. Sural :Soccer video processing for the detection of advertisement billboards. In: Pattern Recognition Letters 29 (2008) 994–1006.

[84] H. Yoon, Y. J. Bae, Y. Yang: A Soccer image sequence mosaicking and analysis method using line and advertisement board detection. In: ETRI Journal, 24(6) (2002).

[85] H.K. Yuen, J. Illingworth, J. Kittler: Ellipse detection using the Hough transform. In: British Machine Vision Conference, 1988.

[86] X. Yu, N. Jiang, L. Cheong, H.W. Leong, X. Yan: Automatic camera calibration of broadcast tennis video with applications to 3D virtual content insertion and ball detection and tracking. In: Computer Vision and Image Understanding, 113 (2009) 643–652.

[87] E. Zeidler: Nonlinear Functional Analysis and its Applications, Part III. Springer, 1984.

[88] Z. Zhang: A flexible new technique for camera calibration. IEEE Transactions on Pattern Analysis and Machine Intelligence. 22(11) (2000) 1330–1334.

[89] Y. Zhou, Y. Cao, L. Zhang, H. Zhang: An SVM-based soccer video shot classification. In: International Conference on Machine Learning and Cybernetics, ICMLC 2005.

[90] Q. Zhang and I. Couloigner : Accurate centerline detection and line width estimation of thick lines using the radon transform. IEEE Transactions on Image Processing, 16(2) (2007) 310–316.

[91] Z.Zili, Q.Qiming, G. Junping, D. Yuzhi, Y. Yunjun, W. Zhaoqiang, D. Fanwei: CART-Based Rare Habitat Information Extraction For Landsat ETM+. In: IEEE International Geoscience and Remote Sensing Symposium, 3 (2008) 1071–1074.