Trabajo fin de grado

Análisis de eventos deportivos mediante visión artificial.

Pablo Saura Pérez

pablo.saura@um.es

48742017A

Tutor: Alberto Ruiz García

13 de septiembre de 2017

Índice

Declaración firmada sobre originalidad del trabajo.

Yo, Pablo Saura Pérez con documento nacional de identidad 48742017-A, y estudiante del Grado en Ingeniería Informática de la Facultad de Informática de la Universidad de Murcia, declaro que el Trabajo de Fin de Grado que he presentado para su evaluación es original y de elaboración personal.

1. Resumen.

En estos tiempos de continuos avances y cambios, la inteligencia artificial se está convirtiendo en una de las disciplinas en las que se está invirtiendo más tiempo y dinero. Una era en la que los datos masivos, las estadísticas, los patrones de comportamiento y en general cualquier dato que tenga potencia para ser explotado, es buscado y analizado. La mejora en la potencia de los dispositivos y en general de la computación está permitiendo que la inteligencia artificial avance de tal manera que tareas que antes solo eran realizadas por humanos cada vez mas se esté tratando de que sean realizadas por máquinas.

La visión artificial es una rama de inteligencia artificial que está sufriendo este avance gracias a que cada vez hay más tareas que necesitan de automatización y una manera de monitorizarlas es mediante cámaras. A veces no es posible instalar un sensor que almacene los datos que estamos buscando, pero, sí somos capaces de captar la escena, analizarla y obtener ese dato que buscábamos. El análisis de comportamientos o situaciones, la automatización de tareas industriales, la navegación automática (tanto en robots como vehículos) son algunas de las tareas que están sufriendo un cambio grande en los últimos años.

En el estado de arte realizamos una breve visión sobre el estado actual de esta disciplina, desde una definición propia, pasando por las distintas disciplinas que hacen uso de ella, hasta un esquema a seguir para la realización de proyectos de este tipo. Repasaremos las distintas librerías que pueden ser utilizadas para proyectos de visión artificial. Por último, comentaremos diferentes aplicaciones ya existentes que tienen relación o de las que se pueden tomar ideas relacionadas con el desempeño de este trabajo.

En este trabajo nos dedicamos al estudio de la visión artificial en el campo del deporte. Se pretende realizar una unión entre las técnicas de visón artificial y alguna problemática en el mundo del deporte, utilizando los recursos de los que disponemos hoy en día. Una propiedad del deporte es que crean afición, esa afición se transforma en forma de retransmisiones deportivas, las cuales se realizan mediante cámaras. Es por esto que existe muchísimo material hoy en día en forma de videos sobre cualquier deporte, la posibilidad de analizarlos y extraer datos de ellos ocupa a muchas empresas en el mundo. Se podría decir que hoy en día en cualquier retransmisión en televisión de algún deporte se nos enseña algún tipo de aplicación mediante visión artificial (cálculo de distancia recorrida, trayectorias, etc...).

A diferencia de otros trabajos o enfoques en este nos vamos a centrar en el análisis mediante el uso de una arquitectura bastante sencilla.

Los recursos que se van a utilizar son: una cámara y una librería de visión artificial. Una cámara para la grabación de las escenas a analizar, también se podrían buscar videos ya realizados , pero, se ha decidido tomarlos específicamente para el trabajo. La cámara será la de un teléfono móvil, de esta manera será un trabajo fácilmente reproducible. La librería de visión artificial utilizada es OpenCV junto con el lenguaje Python. OpenCV es una librería muy potente y muy utilizada hoy en día, por ello existe mucha información para guiarse.

Una vez tenemos los recursos se toman los videos con la cámara que queremos analizar y en los que queremos empezar a realizar las técnicas de visión. Se decide por unos videos de lanzamientos de balones ya que contienen distintos elementos que queremos tratar en el trabajo. La segmentación del objeto y su seguimiento abarcarán gran parte de este trabajo. Una vez se consigue aplicar las técnicas y los procedimientos a estos videos, se extrapolarán a situaciones parecidas que ocurren en los deportes, finalizando con una aplicación a una situación del mundo real.

Junto con la segmentación y el seguimiento del objeto en los videos se decide que sería interesante también realizar una predicción de la trayectoria, ya que se encuentra dentro de un modelo escalable como es la trayectoria balística.

Llegado este punto tenemos 4 grandes módulos en los que se va a centrar nuestro trabajo.

El primer módulo es de la segmentación y aislamiento del elemento de interés. En esta fase realizamos un análisis de los diferentes elementos que componen nuestra escena, dependiendo del comportamiento de estos elementos será necesario el uso de unas técnicas u otras. Debido a que las técnicas genéricas que se estudian normalmente no pueden ser aplicadas directamente, estas han de ser acomodadas a las necesidades de los distintos problemas. No existe un tipo de técnica específica para cada problema, si no que se realiza un conjunto de pruebas con las diferentes técnicas hasta llegar a un resultado satisfactorio.

En los videos tratados en este trabajo tenemos un elemento principal que es el balón y cual queremos aislar de la escena, este será nuestro principal elemento de interés. Para ello también tendremos que tener en cuenta otros elementos como son el fondo o el suelo en nuestra escena. Finalmente después del análisis se decide el uso de un eliminador de partes estáticas como es BackgroundSubtractorMOG2, zonas como el suelo, cielo, terreno de juego si lo hubiera, no son tenidas en cuenta. En cambio la pelota, los jugadores, es decir elementos en movimiento seguirían en la escena. Después mediante técnicas de transformación morfológicas como erosión y dilatación conseguimos que nuestros elementos a aislar se

encuentren mas definidos en nuestra escena.

La parte de segmentación puede diferir mucho entre distintas aplicaciones, por tanto es una parte importante y en la que encontrar la manera óptima no es trivial, es por eso, que esta segmentación pude llevar un tiempo considerable a la hora de realizar una aplicación.

El segundo módulo es del seguimiento o tracking del elemento de interés. En esta parte nos encargamos de aplicar las técnicas sobre el elemento de interés anteriormente aislado para realizar su seguimiento. Este seguimiento nos proporcionará los datos sobre la posición en nuestra escena lo que servirá para la realización de la posterior predicción. A su vez el tratamiento de técnicas de tracking puede utilizarse para otros proyectos en los que sea necesario no solo en proyectos de visión artificial en deporte.

El tracking es una de las mayores problemáticas a las que se pude enfrentar una aplicación de este tipo. Gracias a técnicas como Mean-shift, en concreto la técnica Camshift, se consigue un gran resultado para este trabajo. Al unir la parte de segmentación donde conseguimos separar la pelota correctamente del resto de elementos, el seguimiento se consigue de una manera satisfactoria. Esta técnica de Camshift utiliza un modelo de color mediante la creación de un histograma del elemento a buscar y lo compara con la escena. Esta técnica busca las zonas de mayor densidad de pixeles en común entre el histograma y la escena. Una vez encontrado es capaz de seguir este punto de mayor densidad a lo largo de las siguientes imágenes, este punto de mayor densidad corresponde al elemento de interés en nuestra escena.

Una vez conseguimos almacenar las posiciones del elemento de interés damos por concluido este modulo.

El tercer módulo es del predicción del estado del elemento de interés. Los videos que se han tomado son escenas en las que el elemento de interés realiza un movimiento en un plano vertical perpendicular al eje óptico de la cámara. Esta restricción a la hora de tomar los video se realiza porque esto permite aplicar técnicas de predicción en 2D. Si no fuera así el uso de estas técnicas 2D no nos darían los resultados que esperamos, tendríamos que hacer uso de las técnicas 3D. Las técnicas tradicionales de visión geométrica 3D requieren de múltiples vistas, es decir, distintas cámaras. El uso de estas cámaras se saldría de nuestro objetivo del trabajo que es el uso de una única cámara y recursos ligeros.

Para esta fase se ha implementado un filtro de Kalman que nos permite esta predicción en sistemas 2D. Al tener un sistema dinámico lineal como la balística este filtro es aplicable con resultados satisfactorios. Gracias a esta técnica podemos obtener propiedades a las que

no podemos acceder junto con una incertidumbre sobre ese valor. Esto nos va a servir para predecir posiciones futuras de nuestro elemento de interés junto con esta incertidumbre que hemos comentado. Esta incertidumbre la podemos utilizar como un valor de probabilidad sobre la estimación de la posición, a mayor valor tendremos que "creernos menos"que en una situación futura el elemento se encontrará en esa posición. Pero con una incertidumbre con valor bajo podemos tener mayor certeza de que si estará o estará muy cerca. Básicamente es la idea en la que se va a basar el este predictor.

También tratamos en esta parte como se podría abordar el problema con un sistema no lineal mediante el uso de un filtro UKF.

El cuarto y último módulo es de la creación de una aplicación relacionada con el deporte. En esta fase juntamos todos los conocimientos tratados hasta este punto y los llevamos a una aplicación real en relación al trabajo. La aplicación consiste en un predictor de lanzamientos al aro en baloncesto. De esta manera utilizamos la segmentación, el tracking y la predicción, los módulos que hemos desarrollado en el trabajo.

Al ser este objetivo último del trabajo llegado este punto la investigación e implementación terminan y solo nos queda el balance final del trabajo y las posibles continuaciones.

Como lineas de trabajo futuras se abordan distintos temas: La aplicación de este mismo sistema a otros deportes con lanzamientos y trayectorias parecidas, como por ejemplo, lanzamiento de jabalina o disco; El perfeccionamiento de la aplicación conseguida mediante e aumento de recursos, como podrían ser más cámaras; El cálculo de la trayectoria 3D mediante el uso de otros algoritmos o más cámaras como hemos comentado; La utilización de la base de este trabajo para aplicaciones deportivas que no conlleven trayectorias balísticas, pero en las que sean necesarias también las etapas de segmentación y seguimiento. El código de las aplicaciones realizadas también forma parte de este documento para facilitar la reproducción de los experimentos y las posibles continuaciones sobre el trabajo.

Para concluir un breve repaso de los objetivos cumplidos. Definición de una arquitectura a seguir y cumplimiento de los objetivos con ella. Realización de forma satisfactoria de la obtención de los recursos a utilizar, en el caso de este trabajo, los videos para analizar. Aplicación de las diferentes técnicas de visón artificial propuestas en la fase de análisis de manera satisfactoria. Realización de una aplicación que uniera todos los objetivos anteriores y con ello cumplir todos los objetivos establecidos.

Podemos afirmar que la visión artificial y el deporte son dos grandes campos y que además

se ha demostrado que su funcionamiento conjunto da lugar a aplicaciones muy útiles y con muchas posibilidades hoy en día.

2. Extended Abstract.

In these times of continuous advances and changes, artificial intelligence is becoming one of the disciplines in which more time and money is being invested. One era in which massive data, statistics, behavior patterns and generally any data that has the power to be exploited is searched and analyzed. The improvement in the power of devices and computing in general is allowing artificial intelligence to advance in tasks that previously were performed only by humans and now are increasingly being attempted to be performed by machines.

Computer vision is a branch of artificial intelligence that is undergoing this advance thanks to the fact that there are more and more tasks that need to be automated and one way of monitoring them is through cameras. Sometimes it is not possible to install a sensor that stores the data we are looking for, but we are able to capture the scene, analyze it and obtain that data we were looking for. Behavior or situation analysis, automation of industrial tasks, automatic navigation (both in robots and vehicles) are some of the tasks that have undergone a major change in recent years.

In the state of art we take a brief look at the current state of this discipline, from its own definition, through the different disciplines that make use of it, to a scheme to follow for the realization of projects of this type. We will review the different libraries that can be used for artificial vision projects. Finally, we will discuss different applications that are related or from which ideas related to the development of this work can be taken.

In this work we are interested in the study of artificial vision in the field of sport. The aim is to make a union between artificial vision techniques and some problems in the world of sport, using the resources we have today. A property of sport is that they create hobby, this hobby is transformed into sports broadcasts, which are made by cameras. That's why there is a lot of material nowadays in the form of videos about any sport, the possibility of analyzing and extracting data from them occupies many companies in the world. It could be said that nowadays, in any television broadcast of a sport, there are many applications using computer vision: calculation of distance travelled, trajectories, etc..

Unlike other works or approaches in this one we will focus on the analysis by using a fairly simple architecture.

The resources to be used are: a camera and an artificial vision library. A camera for the recording of the scenes to be analyzed, you could also search for videos already made, but, it has been decided to take them specifically for this work. The camera will be that of a mobile phone, so teh computations will be easily reproducible. The artificial vision library used is OpenCV along with Python language. OpenCV is a very powerful and widely used library today, so there is a lot of information to guide you.

Once we have the resources, we take with the camera the videos that we want to analyze and in which we want to start performing the vision techniques. we chose videos of ball throwing because they contain different elements that we want to deal with in the work. The segmentation of the object and its monitoring will cover much of this work. Once the techniques and procedures are applied to these videos, they will be extrapolated to similar situations that occur in sports, ending with an application to a real-world situation.

Along with the segmentation and tracking of the object in the videos, it was decided that it would also be interesting to make a prediction of the trajectory, since it is within a scalable model such as the ballistic trajectory.

At this point we have 4 major modules on which our work will focus.

The first module is the segmentation and isolation of the element of interest. In this phase we carry out an analysis of the different elements that compose our scene. Depending on the behaviour of these elements it will be necessary to use some techniques or others. Since the generic techniques that are usually studied cannot be applied directly, they must be adapted to the needs of the different problems. There is not a specific type of technique for each problem. Instead, there is a set of tests with the different techniques to reach a satisfactory result.

In the videos treated in this work we have a main element that is the ball and which we want to isolate from the scene, this will be our main element of interest. For this we will also have to take into account other elements such as the background or the floor in our scene. Finally, after the analysis, it is decided to use a static parts eliminator such as BackgroundSubtractorMOG2, for areas such as the ground, sky, pitch if any, are not taken into account. Instead the ball, the players, i. e. moving elements would remain in the scene. Afterwards, through morphological transformation techniques such as erosion and dilatation, the elements to be isolated are more defined in our scene.

The segmentation part can differ a lot between different applications, therefore it is an

important part and finding the best way is not trivial, that is why this segmentation can take a considerable amount of time to make an application.

The second module is the tracking of the element of interest. In this part we apply the techniques on the previously isolated element of interest to follow it up. This follow-up will provide us with data on the position in our scene, which will be useful for the subsequent prediction. At the same time, tracking techniques can be used for other projects where it is necessary, not only in artificial vision projects in sport.

Tracking is one of the biggest problems that an application of this type can face. Thanks to techniques such as Mean-shift, specifically the Camshift technique, a good result is achieved for this work. By joining the segmentation part where we manage to separate the ball correctly from the rest of the elements, the tracking is achieved in a satisfactory way. This Camshift technique uses a color model by creating a histogram of the item to search for and comparing it with the scene. This technique looks for the areas with the highest pixel density in common between the histogram and the scene. Once found it is able to follow this point of greater density along the following images, this point of greater density corresponds to the element of interest in our scene.

Once we have managed to store the positions of the element of interest, we conclude this module.

The third module is the prediction of the state of the element of interest. The videos that have been taken are scenes in which the element of interest moves in a vertical plane perpendicular to the optical axis of the camera. This restriction when it comes to taking the video is made because this allows us to apply 2D prediction techniques. If this were not the case, the use of these 2D techniques would not give us the results we expect, we would have to make use of 3D techniques. Traditional 3D geometric vision techniques require multiple views, i. e. different cameras. The use of these cameras would go beyond our job objective which is the use of a single camera and light resources.

A Kalman filter has been implemented for this phase, which allows us to use this prediction in 2D systems. Having a linear dynamic system such as ballistics this filter is applicable with satisfactory results. Thanks to this technique we can estimates properties that we cannot access together with an uncertainty about this value. This will help us to predict future positions of our element of interest along with this uncertainty that we have discussed. We can use this uncertainty as a probability value on the estimation of the position, the greater

the value we will have to "believe ourselves less"than in a future situation the element will be in that position. But with an uncertainty of low value we can be more certain that it will be close. Basically it's the idea on which this predictor will be based.

We also discussed in this part how the problem could be addressed with a non-linear system by using a UKF filter.

The fourth and final module is the creation of a sports-related application. In this phase we bring together all the knowledge treated up to this point and take them to a real application in relation to the work. The application consists of a basketball hoop pitch predictor. In this way we use segmentation, tracking and prediction, the modules that we have developed in our work. We took different videos of basketball throws, so we could see how it works in different situations, from balls entering to ones not doing it. We would have liked to implement that while the ball is travelling the program gives some percentage of the probability to enter, but we leave this for the future because of lack of time.

As this is the ultimate objective of the work at this point, the research and implementation are finished and we only have the final balance of the work and the possible follow-ups.

We are very pleased with the way in which the work and the results we have achieved have been developed. It is true that the final application is quite simple, but it has been tried to generalize the maximum and give the possibility to transfer to other projects. For my part I finish this work with some knowledge about computer vision that I would not have acquired otherwise and wanting to continue other work related to this discipline. Undoubtedly applying the computer vision to the creation of autonomous cars or using techniques of vision to detect possible dangers are one of the areas that are the most interesting for me. Equally the realization of new techniques and applications in the sport I think they will change the way we know them. In the near future we expect many applications of this type in professional competition.

Different topics are addressed as future lines of work. The application of this same system to other sports with similar throwing and trajectories, such as javelin or disc throwing. The improvement of the application achieved through increased resources, such as more cameras. The calculation of the 3D trajectory by using other algorithms or more cameras as we have discussed. The use of the basis of this work for sports applications that do not involve ballistic trajectories, but in which the stages of segmentation and monitoring are also necessary. The code of the applications made is also part of this document to facilitate the reproduction

of the experiments and the possible continuity on the work.

To conclude, a brief overview of the tasks accomplished. Define an architecture to be followed and fulfill the objectives with it. To perform in a satisfactory way the obtaining of the resources to use, in the case of this work, the videos to analyze. Apply the different artificial vision techniques proposed in the analysis phase in a satisfactory way. Make an application that unites all the above objectives and thereby meet all the established objectives.

We can say that artificial vision and sport are two big fields and that it has also been demonstrated that their joint operation gives rise to very useful applications with many possibilities today.

3. Introducción.

Vivimos en un mundo en el que la informática avanza a pasos agigantados, cada año salen a la venta nuevos procesadores más potentes a precios menores, también, cámaras cada vez mejores y con mayores opciones. No es de extrañar que la visión artificial se esté convirtiendo en una de las disciplinas que más progresión esta sufriendo hoy en día. La visión necesita de estos dos pilares y el que haya máquinas y cámaras más potentes ayuda a que cada vez haya más aplicaciones que se apoyen en esta disciplina.

A su vez, este incremento en la potencia de las tecnologías está produciendo que campos en los que no se utilizaba empiecen a hacer también uso de ella. El deporte se ha practicado desde siempre, pero es hoy en día debido a la alta competitividad, al gran volumen de dinero que mueve y a la gran potencia tecnológica antes comentada, cuando se está uniendo estas dos disciplinas.

Este trabajo va a intentar realizar esta unión por el medio de la creación de una aplicación de visión artificial que pueda ser utilizada en el deporte. Una parte importante va a ser la de la creación de la aplicación mediante recursos que puedan después ser reproducidos a posteriori y que no conlleven un gran desembolso económico.

El objetivo será el de detectar y analizar el movimiento de elementos de interés en una escena deportiva (pelota, jugadores...). En este trabajo se estudiará la diferente problemática que puede haber para conseguir el objetivo y se propondrá una solución para casos sencillos que sirva como punto de partida. Esta solución a su vez se intentará llevar a cabo mediante un planteamiento de bajo coste, donde solo necesitemos una única cámara y un ordenador

con una librería de visión artificial.

El llegar a una solución factible con unos recursos tan limitados nos ayudará a imaginar como de potente podría llegar a ser la visión artificial con mayores recursos.

A su vez, en el marco del trabajo, se explicarán y solucionarán distintos problemas que pueden surgir a la hora de desarrollar una aplicación de visión artificial como pueden ser la segmentación y el seguimiento de objetos en video. Por último, se desarrollará un predictor de estados futuros que será el que aplicaremos a la aplicación deportiva.

3.1. Organización del documento.

Este documento se estructura de la siguiente manera: Un breve resumen del estado del arte actual del marco en el que se engloba este trabajo. En el siguiente capitulo se realiza un análisis y un establecimiento de objetivos a conseguir en el trabajo. A continuación se realiza la fase de diseño e implementación y el en último capitulo se estudian posibles vías futuras y se llevan a cabo las conclusiones generales del trabajo. Las palabras en color azul tienen un enlace que se puede seguir para obtener más información o que llevan a videos de demostración

4. Estado del arte.

4.1. Vision por computador.

La visión por computador o visión artificial, es una disciplina de la inteligencia artificial, la cual se centra en la adquisición de información mediante el análisis de imágenes mediante un ordenador. La información que un humano puede percibir de su entorno mediante el sentido de la vista no es una tarea sencilla para un ordenador, la visión artificial trata de convertir una imagen digital, es decir, una serie de datos numéricos, en información utilizable igual que haría nuestro cerebro.

The British Machine Vision Association and Society for Pattern Recognition Retrieved recoge la siguiente definición sobre la visión artificial:

"Computer vision is concerned with the automatic extraction, analysis and understanding of useful information from a single image or a sequence of images. It involves the development of a theoretical and algorithmic basis to achieve automatic visual understanding."[5]

Cuya traducción sería: "La visión por ordenador se ocupa de la extracción automática, el análisis y la comprensión de la información útil de una sola imagen o una secuencia de imágenes. Implica el desarrollo de una base teórica y algorítmica para lograr la comprensión visual automática ".

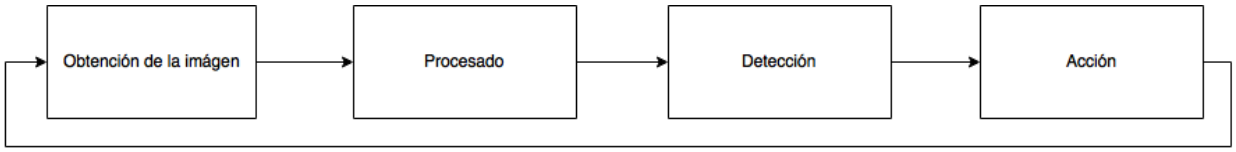
Y es en esta definición donde vemos reflejada la pertenencia a la rama de la inteligencia artificial. Conceptos como extracción automática, análisis y comprensión de información mediante algoritmia reflejan esta parte de las ciencias de la computación y a su vez han convertido la visión artificial una de las disciplinas con mayor crecimiento en esta era de la información.

Hoy en día podemos encontrar que esta disciplina se puede utilizar en numerosos campos y en tareas muy diferentes. Algunas de estas tareas en las que la visión artificial interviene son las siguientes:

- Robótica: Tanto tareas de navegación como de control en procesos en industria.
- Medicina: Procesamiento de imágenes para diagnóstico.
- Topografía: Mediante modelado de superficie mediante imágenes (Ej. Google maps.)
- Clasificadores: Utilizando reconocimiento de patrones y formas sobre objetos. (Ej. escáneres biométricos.)

- Navegación: Utilizado en coches autónomos o vehículos aéreos no tripulados.
- Detección de eventos: Mediante variaciones en las imágenes. (Ej. Conteo de personas, control de entradas y salidas.)
- Aprendizaje automático.
- Muchas otras...

Pero para poder realizar todas estas tareas primero se ha de obtener la información a tratar. Para ello vamos a explicar de forma breve como se realiza esta obtención de información y su posterior análisis.



En el diagrama se puede ver una manera de entender este proceso. A continuación se verá cada fase en mas detalle.

- Obtención de la imagen: En esta fase es donde conseguimos la imagen o secuencia de imágenes que queremos analizar, se utiliza una cámara para ello.
- Procesado: Esta fase consiste en la aplicación de técnicas o instrucciones de modificación de la imagen para conseguir una nueva imagen donde sea mas fácil la detección que queremos llevar a cabo. Entre estas técnicas podemos encontrar: Suavizado, modelos de color, dilataciones, filtrados y muchas otras técnicas.
- Detección: Una vez hemos procesado la imagen se utilizan técnicas de detección para encontrar en la imagen la información que para nosotros será valiosa. (Ej. detección de un objeto).
- Acción: Ya con con el resultado de la detección podemos aplicar la acción para la que hemos desarrollado nuestro sistema (Ej. seguimiento del objeto antes detectado).

4.2. Representación del color.

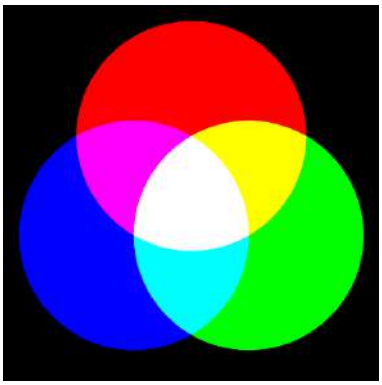
Las representación de color son modelos matemáticos que permiten representar los colores mediante valores numéricos. Debido a que un ordenador no tiene la percepción que puede tener un humano de los colores, estos han de ser codificados y representados mediante valores que los ordenadores si puedan entender.

La forma más común de representación es mediante vectores numéricos, a continuación se comentan distintos tipos que utilizamos en este trabajo.

4.2.1. Representación en RGB.

En este modelo los colores se representan mediante tripletas de normalmente enteros de 8 bits, valores enteros entre 0 y 255. Por lo tanto tendremos un vector de la forma (R, G, B) donde R, G, B representarán la intensidad de cada color (rojo, verde y azul) y la tripleta representará un color por adición de estos tres. Algunos ejemplos son:

- $(255, 0, 0)$ Color rojo.
- $(0, 0, 0)$ Ausencia de color, es decir, color negro.
- $(255, 255, 255)$ Color blanco.



Cabe destacar que el modelo no define con exactitud los diferentes colores, dependiendo del dispositivo que use este modelo un mismo valor de RGB puede verse de diferente manera. El modelo de color es el mismo pero el espacio de color puede variar.

4.2.2. Escala de grises.

Una imagen en escala de grises no es más que un valor para cada píxel, en 8 bits de 0 a 255, donde cada valor es un tipo de gris. Para el tratamiento de imágenes se utiliza mucho la escala de grises gracias a su sencillez en la aplicación de técnicas de visión artificial.

Computacionalmente una imagen en escala de grises es una matriz ancho x alto donde cada posición es un píxel con el valor de gris.

Para finalizar esta sección decir que existen otras representaciones de color aparte de estas como: HSL, YUV, LAB, LUV...

4.3. Librerías de visión artificial.

A la hora de realizar una aplicación en la que utilicemos visión por computador deberemos escoger que librería de entre las que existen queremos utilizar. En esta sección comentaremos distintas librerías que existen hoy en día para la realización de este tipo de aplicaciones.

- Accord.NET Es un framework .Net para machine learning, combinado con procesamiento de audio e imágenes.
- Ilastik Como ellos se definen: Herramienta amigable y simple para la clasificación, segmentación y análisis de imágenes.
- Librerías sobre Matlab Matlab provee de una serie de funciones y algoritmos sobre visión artificial para ser utilizados en su plataforma.
- OpenCV Es la librería que utilizamos en este trabajo, de carácter libre tanto en uso academico como comercial. Contiene interefaces para C++, C, Python y java. Fue diseñado para que fuera computacionalmente eficiente y poder ser usado en aplicaciones en tiempo real. Tiene una de las comunidades más grande con mas de 47 millones de usuarios según su página web.
- SimpleCV Una librería de visión artificial para gente que no necesita la potencia de OpenCV y de manera mas sencilla, así es como se definen ellos en su página web.

4.4. Sistemas de predicción.

Dentro de las aplicaciones de la visión artificial se encuentran las aplicaciones de predicción mediante el uso de cámaras.

Un ejemplo en gran desarrollo son los vehículos con conducción autónoma. Estos vehículos utilizan cámaras para modelar el entorno y la vez intentan predecir el comportamiento de otros vehículos en la carretera para anticipar la toma de decisiones.

Otro ejemplo de aplicación es el desarrollado por la universidad de el Algarve llamada Pool Live Aid. Esta aplicación ayuda a los jugadores novatos en billar a mejorar sus habilididades. Por medio de una cámara se calcula la trayectoria que va a recorrer la bola sobre la mesa.

Imagen obtenida de su página de web

4.5. Visión aplicada al deporte.

La gran pasión con la que se viven hoy en día las competiciones deportivas unido a la a veces complicada toma de decisiones sobre el reglamento, han derivado en el uso de la tecnología en el deporte.

La visión es uno de los campo en emergencia a la hora de aplicar el reglamento de los diferentes juegos, pero a su vez la visión esta presente a la hora de la preparación de los encuentros, en entrenamientos o análisis de encuentros que ya han ocurrido. En un sector que mueve mucho dinero al cabo del año, la implantación de tecnología como la visión artificial deparará mejor preparación y unos encuentros mas justos.

Existe numerosas aplicaciones de la visión artificial en el deporte, vamos a enumerar y comentar a continuación unas de las mas presentes en el día a día deportivo.

4.5.1. Aplicaciones de tracking.

Se puede decir que el deporte ha experimentado en sus carnes la era de los datos masivos, en cada partido o encuentro se nos avasalla con numerosas estadísticas y datos sobre los equipos o participantes. La visión artificial tiene mucho que ver en esto y es que a partir del desarrollo de aplicaciones que pueden tomar un partido grabado y analizar lo ocurrido se han podido obtener numerosas estadísticas que son usadas desde los propios equipos, equipos rivales, televisiones o incluso apostantes.

Un sistema bastante conocidos es el que mide la distancia recorrida por los jugadores en partidos de futbol. La FIFA (Fédération Internationale de Football Association) utiliza en sus competiciones la tecnología MATRICS por medio de cámaras situadas en el estadio

se puede calcular la posición de los jugadores en todo momento y por tanto calcular su movimiento y distancia recorrida durante todo el encuentro.

4.5.2. Análisis de comportamientos.

SECOND SPECTRUM es una empresa que utiliza un sistema de cámaras para capturar y analizar los comportamientos de los jugadores en baloncesto.
Este sistema por ejemplo analiza los distintos tiros de un jugador y predice la probabilidad de encestar un lanzamiento dependiendo de la zona en la que se encuentre, esto puede servir al jugador para entrenar desde posiciones en las que no tiene tanto acierto.

4.5.3. Toma de decisiones.

Sin duda el juez deportivo más conocido del mundo es el Ojo de Halcón. Este sistema se implantó en el tenis en el año 2005 y consistía en una serie de cámaras estrategicamente colocadas que por medio de triangulación y calculo de la velocidad calculaban la trayectoria de la bola. Dado el modelo de juego y las imágenes, que son recogidas con una cámara de alta velocidad, calcula la posición 3D de la bola, de esta manera se puede saber si una pelota es válida o no. Este sistema fue diseñado por ingenieros de Roke Manor Research Limited aunque posteriormente el proyecto fue continuado por Hawk-Eye Innovations Ltd.

Posteriormente en 2012 el **Ojo de Halcón** fue aceptado en uno de los deportes más practicados del mundo, el fútbol. En este caso se utilizan 7 cámaras colocadas en el interior de la portería, estas cámaras toman imágenes que son procesadas para recrear la posición 3D junto con un microchip en el interior de la pelota. De esta manera se puede saber si una pelota ha sido introducida completamente dentro de la portería y por tanto un gol debe subir al marcador. El sistema esta avisa al arbitro en el momento en el que debe de conceder la anotación.

Ambos sistemas han contribuido a disminuir la polémica que en ocasiones se produce en estos deportes debido al gran seguimiento que tienen.

5. Análisis de objetivos y metodología.

5.1. Definición del trabajo.

El propósito de este trabajo es la realización de una aplicación mediante el uso de visión artificial donde comprobar la gran potencia de este área de la inteligencia artificial sin tener que usar numerosos recursos.

El trabajo consta de varias partes, primero el estudio de técnicas de visión como son la segmentación o el seguimiento. En segundo lugar técnicas de predicción aplicadas a modelos de la vida real, como el modelo balístico. En último lugar la aplicación de las conclusiones obtenidas de las dos primeras partes a aplicaciones relacionadas con el deporte, más concretamente a la predicción de trayectorias en lanzamientos.

Para la realización del trabajo se busca la utilización de recursos de fácil acceso y no tener que contar con una gran infraestructura para su reproducción. Así solo se utilizará una única cámara, en el caso de este trabajo, la de un teléfono móvil para la toma de las imágenes a analizar. A su vez se utilizarán herramientas informáticas de libre acceso que cualquiera puede descargar y utilizar. De esta manera se consigue que el trabajo sea sencillo y de fácil reproducción, pensando también en posibles mejoras o la continuación en el futuro.

5.2. Metodología.

Una vez definido el trabajo definimos la metodología a seguir para la realización del mismo. A continuación pasamos a definir cada una de las partes:

- **Elección de herramientas:** A la hora de la realización de un proyecto en visión artificial se han de tener en cuenta las diversas herramientas disponibles para la obtención de recursos y el procesamiento de estos. Por una parte tenemos la obtención de imágenes, por lo tanto necesitaremos o bien cámaras para la toma de imágenes por nuestra cuenta o hacernos con imágenes en algún servicio de almacenamiento de estas. En nuestro caso vamos a tomar nosotros mismos las imágenes o videos a analizar y para ello utilizaremos la cámara de un dispositivo móvil. Una vez obtenidos los datos a procesar necesitamos una librería de visión artificial, en el caso de este trabajo se utilizará OpenCV, el cual explicaremos en la sección de Tecnologías y herramientas. A la hora de elegir una librería siempre es mejor escoger un lenguaje de programación con el que nos sintamos cómodos, para este caso se utilizará el lenguaje Python. Una vez tenemos todo esto ya podemos empezar el procesamiento de las imágenes.

- **Estudio de las técnicas de visión:** Una vez con las imágenes debemos realizar las técnicas para centrarnos las partes importantes, en el caso de este proyecto es necesario encontrar por medio de estas técnicas el objeto al que se le va a realizar la predicción. Por medio de técnicas de segmentación y de tracking se conseguirá saber en que posición se encuentra el objeto en cada momento.
- **Estudio de las técnicas de predicción:** Gracias a los datos de la posición obtenidos en la parte anterior, se podrá realizar la predicción de la posición en instantes siguientes. En esta parte se deberá de adecuar el procedimiento de predicción que mejor se ajuste al trabajo.
- **Desarrollo de una aplicación:** En esta parte juntaremos todos los avances conseguidos en los pasos anteriores para realizar una aplicación que sirva para predecir una trayectoria de un objeto, comprobaremos la bondad de los resultados para ver si se adecuan a lo que estábamos buscando.
- **Aplicación al deporte:** Por ultimo buscaremos aplicaciones del sistema creado anteriormente al sector deporte.

5.3. Objetivos a cumplir.

Se podría decir que hay unos objetivos intermedios que hemos detallado en la parte de metodología pero el objetivo último del trabajo es comprobar si es posible la realización de un sistema que consiga predecir una trayectoria y a su vez este sistema aplicarlo a algún deporte obteniendo unos datos que sea razonables, también se valora el que no sean necesarios un número grande de cámaras como en otras aplicaciones ya existentes.

5.4. Tecnologías y herramientas.

- **Cámara:** La toma de imágenes y videos se realiza mediante la cámara de un dispositivo Iphone 7, se ha decidido elegir la cámara de un dispositivo móvil para que no fuera necesario un gran desembolso en cámaras especificas de alta velocidad utilizadas en otros proyectos de visión artificial. En concreto la cámara de este dispositivo permite la realización de videos con gran calidad y la posibilidad de elegir un alto número de fps que nos permitirán un mejor análisis al tener más imágenes por segundo.

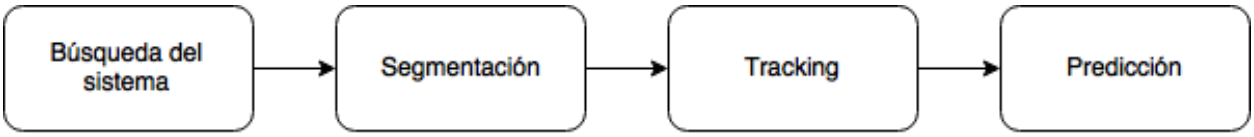
- **Librería y lenguaje de programación:** De entre las librerías comentadas anteriormente se ha escogido OpenCV por su gran potencia y por su gran número de usuarios e información que se puede encontrar sobre ella. A su vez se ha intentado aplicar y continuar de alguna manera los conocimientos obtenidos en la asignatura de Visión Artificial.

  Aunque hay versiones de OpenCV para programar en C++ y Java la que se utilizará en este trabajo será la de Python. La curva de aprendizaje de este lenguaje es menor al ser un lenguaje que tiene este propósito, unido a haber tratado con el durante el curso. Python adolece a veces de problemas de velocidad pero se ha preferido al ser un lenguaje más fácil de aprender.

6. Diseño y resolución del trabajo realizado.

En esta sección se va a desarrollar la parte del diseño y de implementación del proyecto, como ya hemos comentado anteriormente, este trabajo esta orientado sobre unos objetivos intermedios que son los que se vamos a desarrollar a continuación.

Estos objetivos intermedios están conectados unos con otros de manera que se apoyan en los anteriores para ser realizados. En la siguiente figura tenemos una representación de cuales son los pasos a realizar para la implementación:



6.1. Búsqueda del sistema a analizar.

A la hora de realizar un proyecto de visión artificial es muy importante analizar primero el sistema sobre el que se quiere ejecutar. Debido a que hay que usar cámaras para grabar las acciones a realizar hay que estudiar su colocación para la mejor captación de información de acuerdo a posibles limitaciones que tengamos.

El caso que abarca este proyecto es el de predicción de trayectorias balísticas. Se ha escogido este tipo de sistema porque puede ser utilizado en aplicaciones al deporte, como puede ser el análisis de lanzamientos en baloncesto, fútbol, voleibol, lanzamiento de peso, jabalina o martillo entren otros deportes donde también se realicen lanzamientos que se ajusten a los lanzamientos de trayectoria balística.

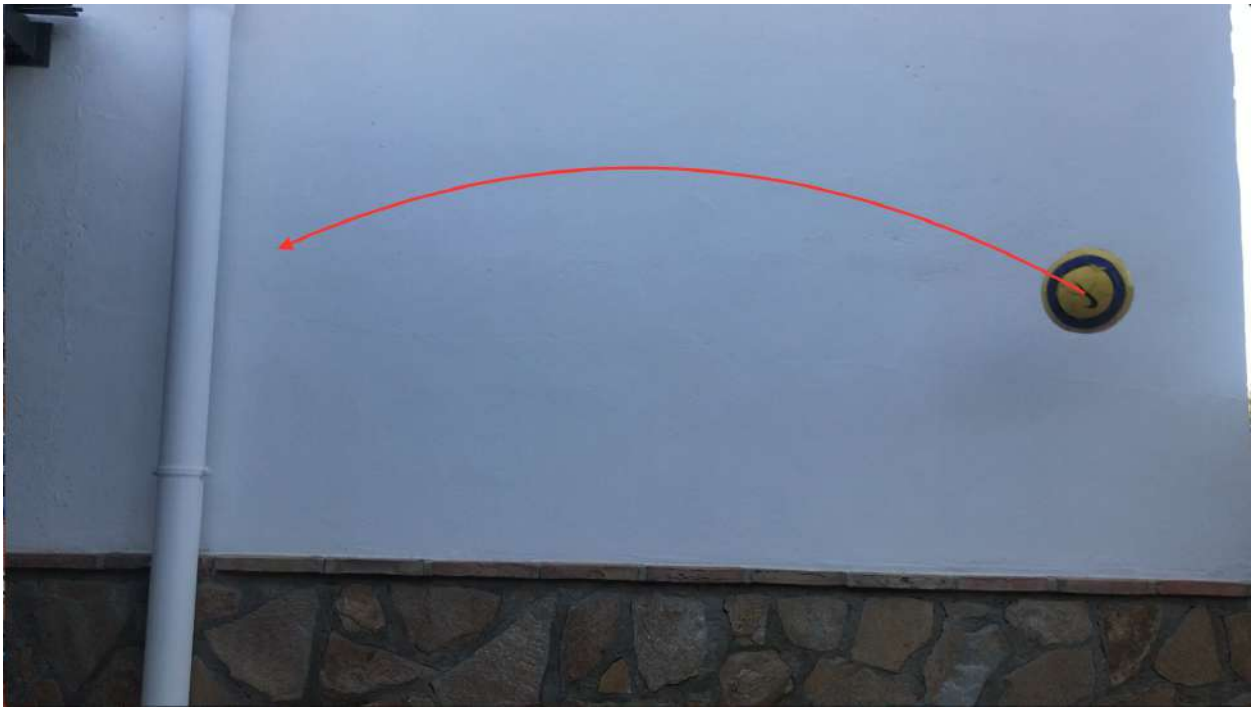
"La trayectoria balística es la trayectoria de vuelo que sigue un proyectil sometido únicamente a su propia inercia y a las fuerzas inherentes al medio en el que se desplaza, principalmente la fuerza gravitatoria.

Cuando sobre el proyectil tan solo actúa la gravedad, la trayectoria balística es una parábola. Sin embargo, la presencia de otras fuerzas, tales como la resistencia aerodinámica (atmósfera), la fuerza de sustentación, la fuerza de Coriolis (efecto de la rotación terrestre), etc. hace que la trayectoria real sea algo diferente de una parábola." [7]

En este trabajo no merece la pena considerar más que la gravedad ya que otras fuerzas (fricción, etc.) pueden tratarse como pequeñas perturbaciones.

Una vez elegido el problema a resolver debemos tomar las muestras para la experimenta-

ción, en nuestro caso videos de trayectorias de tipo balística. Los videos que se han escogido son los de pelotas recorriendo un movimiento en un plano perpendicular al eje óptico de la cámara. Esta restricción a la hora de la toma de videos permite aplicar técnicas de predicción 2D y por ello solo necesitamos una cámara y un video para poder llevar a cabo el objetivo. Justo este era uno de los objetivos a tratar en este trabajo, intentar predecir la trayectoria utilizando solo una cámara. Para la utilización de técnicas tradicionales dede visión geométrica en 3D sería necesario el tener múltiples vistas.



Otro detalle importante es el saber cuantos datos van a ser necesarios para poder empezar a realizar los cálculos de manera acertada, en una aplicación de visión artificial donde utilizamos videos lo que se va analizando son las imágenes una a una.

Un video grabado de manera normal realiza de 24 a 30 imágenes por segundo de video. Cámaras mas potentes que tenemos hoy en día son capaces de realizar videos a 60 imágenes por segundo lo que nos ayudaría bastante de cara al análisis. Debido a que en un lanzamiento su trayectoria depende de la velocidad a la que se realice, podría ocurrir que se saliera de la cámara o llegara al suelo donde produciría un rebote y cambiaría la trayectoria, para estos casos en los que el tiempo de recogida de datos es menor cuantas mas imágenes por segundo tengamos para analizar más podremos aproximarnos al mejor resultado.

Para hacer una comparativa de como repercuten las imágenes por segundo a las que esta grabado el video realizaremos diferentes videos a 60 y 120 imágenes por segundo, ya que el

dispositivo que vamos a utilizar nos permite esta opción.

Otro aspecto importante a la hora de realizar los videos es la distancia al objeto que queremos analizar, cuanto mas lejano sea menos píxeles abarcará en las imágenes y más complicado será la realización de la segmentación y seguimiento que veremos en las próximas secciones. En la siguiente imagen tenemos una comparativa de como afecta lo comentado anteriormente, la primera imagen será mas complicado el seguir la pelota que en la segunda al ser más pequeña, pero a su vez la pelota realizará un vuelo más largo y podremos predecir la trayectoria de mejor manera.

Encontrar el balance óptimo dependiendo de nuestra aplicación será un factor importante a tener en cuenta antes de comenzar a realizar la implementación.

6.2. Segmentación de imagen.

Una vez disponemos del sistema y el material a analizar comenzamos la fase de segmentación. En nuestro caso el proceso de segmentación se llevará a cabo en los videos realizados con el objetivo de localizar la pelota a la cual vamos a hacer el seguimiento.

"El objetivo de la segmentación es simplificar y/o cambiar la representación de una imagen en otra más significativa y más fácil de analizar. La segmentación se usa tanto para localizar objetos como para encontrar los límites de estos dentro de una imagen. Más precisamente, la segmentación de la imagen es el proceso de asignación de una etiqueta a cada píxel de la imagen de forma que los píxeles que compartan la misma etiqueta también tendrán ciertas características visuales similares." [8]

Aunque existen numerosas técnicas de segmentación en procesamiento de imágenes en esta sección nos vamos a centrar en comentar y explicar las que se han utilizado para este proyecto.

Para realizar el tracking o seguimiento del objeto vamos a utilizar un modelo de color de ese objeto, es en esta sección en la que tenemos que conseguir mediante las técnicas de segmentación que sea sencillo realizar ese modelo de color. Para ello lo que vamos a hacer es aislar de alguna manera ese objeto en movimiento de manera que otros objetos o escenarios no interfieran.

Al ser un objeto que se mueve entre frames[1] podemos usar esta diferencia entre una imagen y la anterior para detectar objetos que estén en movimiento. De esta manera podemos eliminar objetos o fondos estáticos acotando el objeto que estamos buscando. Ejemplo en video.

La técnica utilizada para esta eliminación es **BackgroundSubtractorMOG2** contenida dentro de la librería OpenCV. Estos eliminadores de fondo constituyen una parte importante en numerosas aplicaciones de visión artificial.

El ejemplo que siempre se toma para explicarlos es el de una aplicación que cuenta visitantes que entran a un lugar o vehículos que circulan por una carretera mediante una cámara estática. Para realizar el conteo primero se necesita aislar este visitante o vehículo que serán los elementos en movimiento frente al fondo estático. Si tuviéramos un fondo uniforme sería muy sencillo eliminarlo pero en las aplicaciones reales los fondos no son así. Estas técnicas se encargan de mediante sustracción de un frame con el anterior encontrar las diferencias y por tanto los objetos que se han desplazado. [9][10]

Un problema que puede aparecer es el del tratamiento de las sombras de los objetos. La sombra de un objeto se mueve con este y por tanto podría aparecer como un falso positivo en nuestra aplicación.

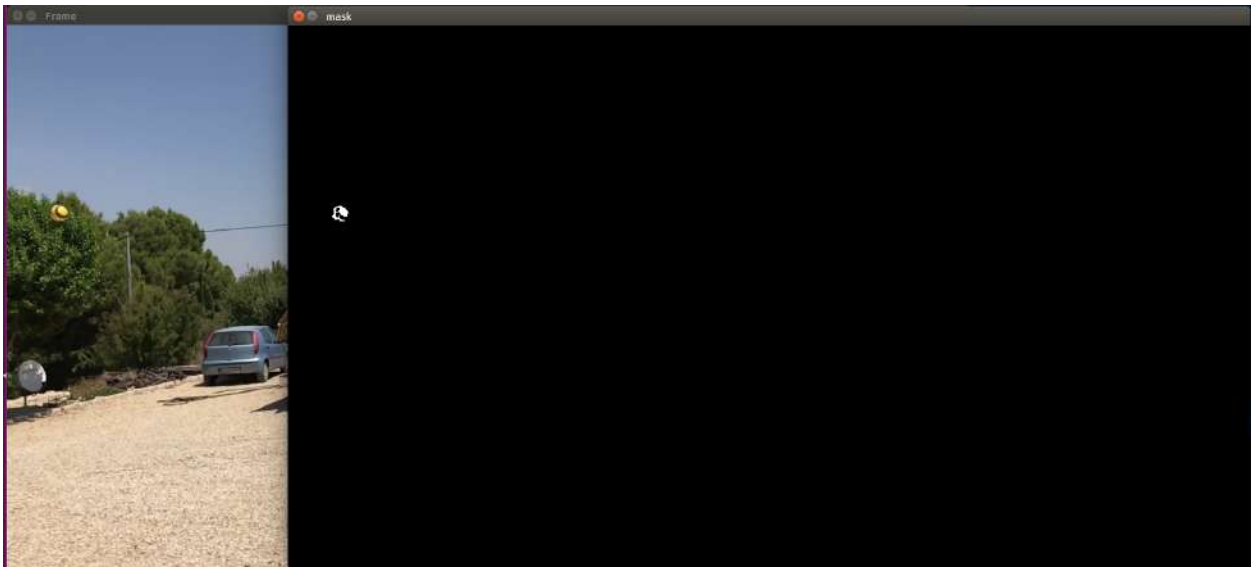
El uso de **BackgroundSubtractorMOG2** conlleva una mejor adaptación frente a cambios de luz que **BackgroundSubtractorMOG**, a su vez contiene una opción de detección de sombras que, aunque disminuye la velocidad de procesamiento, es muy útil para nuestra aplicación. La diferencia en el algoritmo de MOG es que utiliza un método para modelar cada píxel de fondo mediante una mezcla de distribuciones de K gaussianas (K = 3 a 5). Los pesos de la mezcla representan las proporciones de tiempo que esos colores permanecen en la escena. Los colores de fondo probables son los que permanecen más largos y más estáticos. Mientras que MOG2 selecciona el número apropiado de distribución gaussiana para cada píxel.

En la siguiente imagen vemos como quedaría el uso de la técnica en nuestra aplicación:

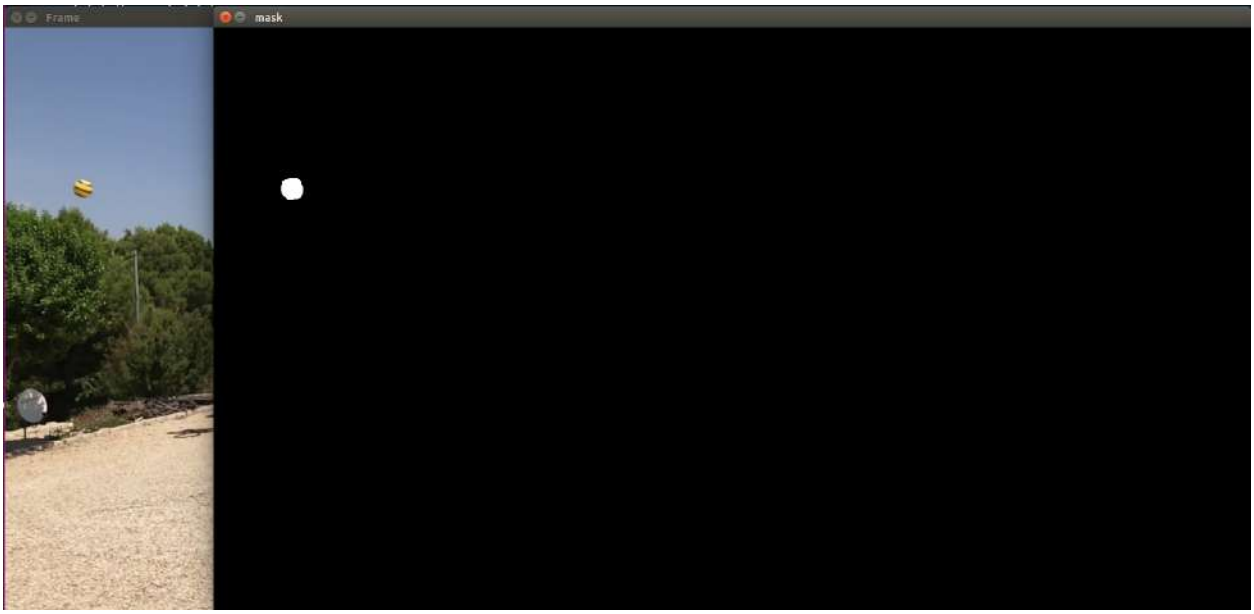
[1] Un frame es cada una de las imágenes que forman un video.

Hemos conseguido aislar la pelota como estábamos buscando, pero también tenemos es una serie de puntos debido a pequeñas variaciones o ruido entre las imágenes. Estas pequeñas variaciones pueden estar ocasionadas por ejemplo por el viento que mueve las hojas y el algoritmo lo detecta. El siguiente paso es eliminar en lo posible esas variaciones para que el objeto más importante sea la pelota. Vamos a realizar tres pasos para ello mediante tres funciones de OpenCV :

- **Erode**: Elimina los píxeles exteriores de los objetos. [11][12]
- **MedianBlur:** Toma un conjunto de píxeles y realiza la media de estos y los remplaza, nos va a servir para eliminar puntos blancos sueltos en zonas negras. [13]
- **Dilate:** Al contrario que la erosión, nos servirá para aumentar los píxeles en la pelota, porque se hayan perdido con las otras dos técnicas o porque queramos hacer más grande esta parte.[12]



Erode.

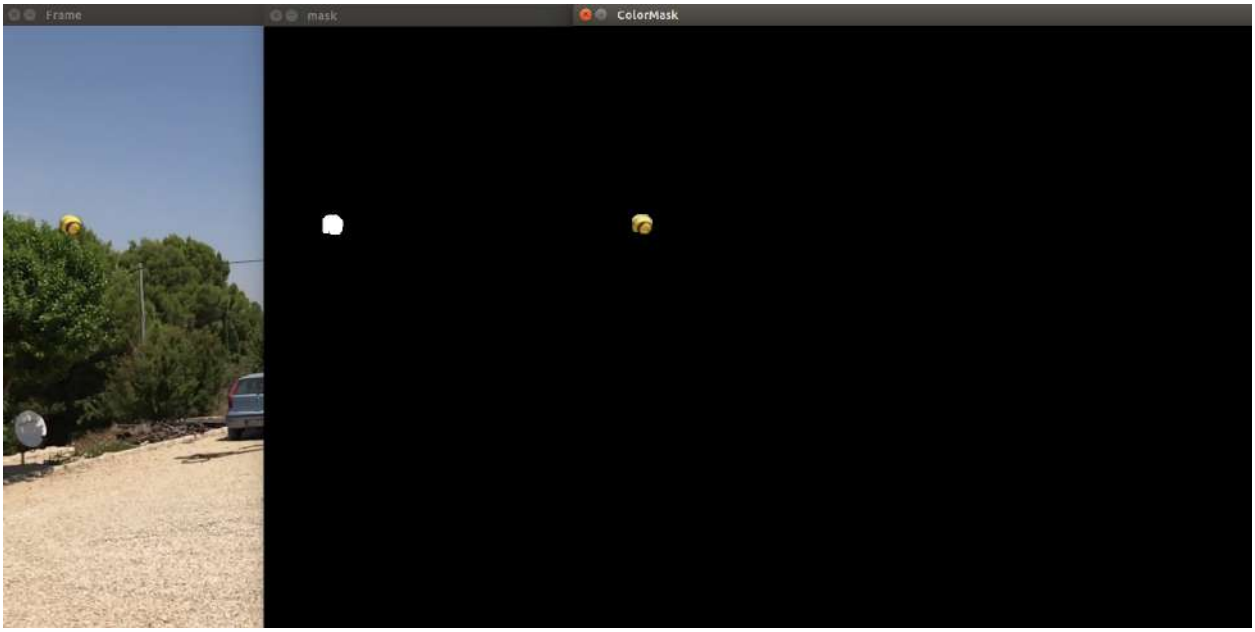


Median + Dilation.

La diferencia es notable después de realizar estos pasos. Por último convertiremos el resultado anterior en una máscara (una máscara es una imagen donde el valor de los píxeles es un valor verdadero o falso).

Al aplicar esta máscara a la imagen original nos dará como resultado las zonas que antes teníamos como blancas que es donde nuestra máscara tendrá valor verdadero. El resultado

es el de la pelota a color junto con el fondo negro. Se puede ver el resultado también en este ejemplo en video.



Original, Máscara, Aplicación de la Máscara al original

Hemos conseguido aislar el objeto que necesitábamos para nuestra aplicación de una manera razonable tal y como queríamos en esta sección.

6.3. Tracking.

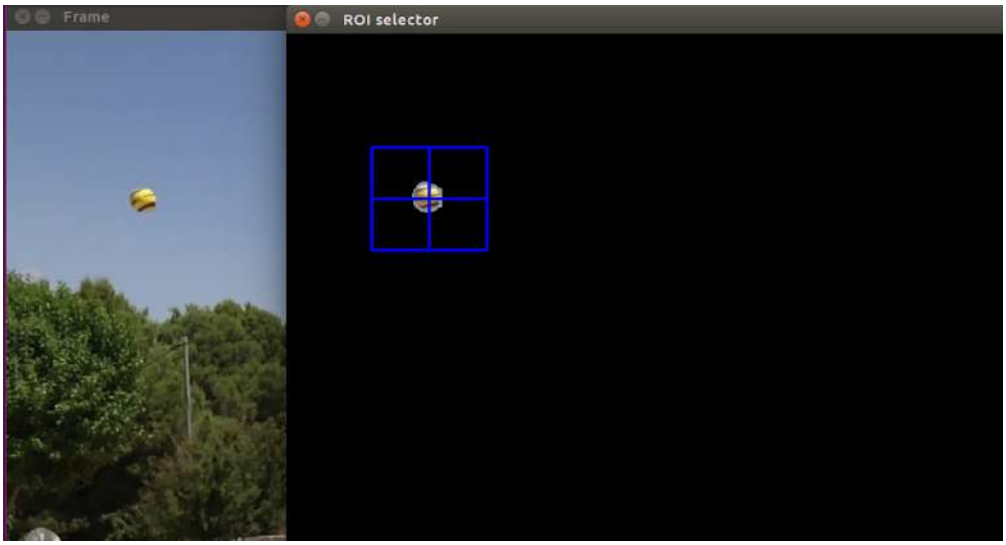
Supongamos que queremos detectar la posición de un objeto en movimiento en una secuencia de imágenes. Si el objeto es más o menos compacto y tiene un color característico podemos aplicar una técnica similar. A partir de la imagen de verosimilitud del modelo (la reproyección del histograma), podemos obtener la elipse de incertidumbre que engloba la mayor parte del objeto y con ella definir una región de interés del tamaño adecuado para trabajar en el frame siguiente.

Esto es la base de las técnicas Mean shift y Camshift. La diferencia entre ellas es que Mean shift siempre realiza la búsqueda con el mismo tamaño y Camshift (Continuously Adaptive Mean Shift) varia el tamaño de la ventana de búsqueda en cada ejecución. Puede ocurrir que un objeto se vaya acercando o alejando de la cámara y por tanto haciendose más grande o más pequeño en número de píxeles y es aquí donde Camshift funciona de mejor manera que Mean shift.

Es por esto que se ha escogido Camshift[14][15]para realizar el seguimiento de nuestro objeto.

Para realizar el seguimiento primero habrá que crear el modelo de color del objeto que la función Camshift deberá de seguir. Este modelo es el que pasaremos a la función junto con la posición de la imagen donde debe empezar a realizar la búsqueda.

El modelo en nuestro caso serán los valores del histograma de color de la pelota. Mediante la selección de la región de interés y las funciones que tiene OpenCV para calcular histogramas realizamos el modelo.



Una vez creado el modelo y con la posición obtenida al seleccionar la región debemos

utilizar la función con los parámetros necesarios. Como la función lo que busca es la zona con mayor densidad de pixeles para el histograma creado lo que hay que hacer es crear la reproyección del histograma. Esta reproyección junto con la posición inicial de búsqueda y un criterio de terminación (número de iteraciones o movimiento del objeto) son los parámetros que tomará la función. Lo que ocurrirá y veremos en la siguiente imagen es que cuando el objeto se mueve, la reproyección del histograma se mueve con él, esto será lo que haga que el algoritmo mueva la ventana de búsqueda a la siguiente posición de mayor densidad.

El resultado de la ejecución devuelve la nueva ventana donde buscar junto con el resultado de la búsqueda. En verde la zona que la función ha detectado y en azul la zona por donde seguirá buscando en la siguiente iteración.

Por último, nuestro objetivo con el seguimiento es el de tomar una serie de datos que nos puedan servir para el fin último, el de la predicción. Gracias a Camshift podemos almacenar las posiciones en las que se encuentra nuestro objeto en cada frame, estas posiciones nos van a servir para calcular cual será la posición estimada en un instante posterior aplicando el modelo matemático que afecta en cada caso, en el de este trabajo el modelo balístico.

Cuántos mas datos tengamos al realizar la predicción mas fiable será esta. En este momento entra en juego el modo de grabación que estemos utilizando, en concreto, los frames

por segundo a los que estemos realizando la grabación.

En las siguientes imágenes vamos a poder ver la comparación entre un video grabado a 60fps y otro a 120fps (podemos también imaginar como sería en videos grabados a 30fps o 240fps).

Video grabado a 60fps

Video grabado a 120fps

Al tener el doble de imágenes por segundo podemos tomar el doble de mediciones, es un

dato importante a tomar en cuenta a la hora de realizar una aplicación de este tipo. Para terminar esta sección un ejemplo en que el objeto desaparece de la escena y vuelve a entrar en ella.

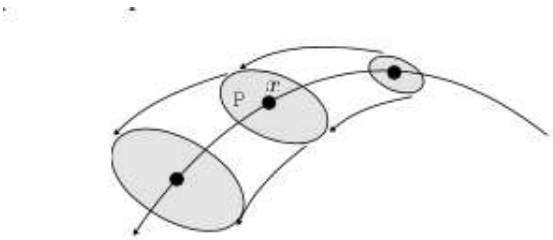
Una ejecución completa en video se encuentra en el siguiente enlace para más detalle.

6.4. Predicción.

Una vez es posible aislar el objetivo, realizar su seguimiento y obtener datos de la posición de este, podemos utilizarlos para realizar una predicción de un estado futuro. A la hora de buscar un predictor hay que tener en cuenta el tipo de sistema en el que estamos trabajando. El sistema de este trabajo es el de predicción de trayectorias de tipo balístico, por tanto, es un sistema dinámico lineal que evoluciona de acuerdo a una ley dinámica. En este tipo de sistemas se puede utilizar el **filtro de Kalman**[16].

"El **filtro de Kalman** es un algoritmo desarrollado por Rudolf E. Kalman en 1960 que sirve para poder identificar el estado oculto (no medible) de un sistema dinámico lineal, que sirve además cuando el sistema está sometido a ruido aditivo. Ya que el Filtro de Kalman es un algoritmo recursivo, este puede correr en tiempo real usando únicamente las mediciones de entrada actuales, el estado calculado previamente y su matriz de incertidumbre, no requiere ninguna otra información adicional. El filtro de Kalman tiene numerosas aplicaciones en tecnología. Una aplicación común es la guía, navegación y control de vehículos, especialmente naves espaciales. Además el filtro es ampliamente usado en campos como procesamiento de señales y econometría." [17]

Por lo tanto tenemos un sistema que evoluciona de acuerdo a una ley dinámica $x_{t+1} = F(x_t)$. Nuestra información acerca del estado tiene incertidumbre, que va aumentando a medida que pasa el tiempo:



Suponemos que tenemos alguna propiedad $z = H(x)$ que podemos observar. Lo ideal sería poder medir ese estado en cuyo caso $z = x$ y el problema sería trivial. Al tratarse de un problema de predicción lo normal es que no tengamos siempre esa información y además puede que tengamos ruido. Esta z será alguna propiedad que depende de unas pocas componentes de x. Por ejemplo, si deseamos estimar la trayectoria de un objeto móvil usando una cámara, el estado incluye coordenadas de posición y velocidad, pero la observación sólo registra información de posición (como es nuestro caso).

Aunque el problema parece imposible de resolver por falta de información, habría que crear una especie de modelo inverso de H que reconstruyera el estado a partir de información parcial. Llegar a ese modelo inverso es imposible pero mediante información conseguida mediante observación podemos reducir la incertidumbre sobre el estado completo. En la práctica tendremos una serie de observaciones z_t que son tomadas a lo largo de la evolución de sistema, con las que realizar una secuencia de pasos predicción-corrección para ajustar el estado x_t. La reducción de la incertidumbre (con información incompleta) se produce porque la ley de evolución dinámica establece una dependencia entre las distintas variables de estado obtenidas por la observación y las de tipo latente.

En nuestro ejemplo de posición y velocidad, la observación solo obtiene la posición en un instante, pero la velocidad puede estimarse a través de la dependencia de la posición con respecto a posiciones y velocidades anteriores. Es interesante ver como esa estimación de la velocidad se consigue de forma automática, sin la necesidad de construirla explícitamente.

El caso más simple, tanto la ley dinámica F como el modelo de observación H son funciones lineales y la incertidumbre sobre el estado, el ruido del sistema y el de la observación son aditivos y de tipo gaussiano, se puede aplicar el algoritmo del **filtro de Kalman**.

El algoritmo quedaría así: dadas las transformaciones F y H, con ruido modelado mediante matrices de covarianza S y R respectivamente, la información sobre x, en forma de media μ y matriz de covarianza P, se actualiza con una nueva observación z de la siguiente forma:

- $\mu' \leftarrow F\mu$
- $P' \leftarrow FPF^T + S$
  (Evolución del estado)
- $\epsilon \leftarrow z - H\mu'$
  (Discrepancia entre la observación y su predicción)
- $K \leftarrow P'H^T(HP'H^T + R)^{-1}$
  (Ganancia de Kalman)
- $\mu \leftarrow \mu' + K\epsilon$
- $P \leftarrow (I - KH)P'$ (Corrección)

Una vez explicado el modelo teórico alrededor del **filtro de Kalman**, vamos a llevarlo a la práctica en nuestro trabajo. En este caso, a partir de las mediciones anteriores se va a intentar predecir la posición en instantes posteriores junto con la incertidumbre. La función que se va a utilizar es la desarrollada por los profesores de la facultad dentro del paquete de funciones **umucv** para OpenCV y que se encuentra en el siguiente enlace.

```
import numpy as np
import numpy.linalg as la

def kalman(mu,P,F,Q,B,u,z,H,R):
    # mu, P : estado actual y su incertidumbre.
    # F, Q  : sistema dinamico y su ruido.
    # B, u  : control model y la entrada.
    # z     : observacion.
    # H, R  : modelo de observacion y su ruido.

    mup = F @ mu + B @ u;   #Estado predicho sin observacion.
    pp  = F @ P @ F.T + Q;  #Incertidumbre cuando no hay observacion.

    zp = H @ mup      #Prediccion respecto al modelo.

    # si no hay observacion solo hacemos prediccion.
    if z is None:
        return mup, pp, zp

    epsilon = z - zp      #Discrepancia entre la observacion y su
       prediccion.

    k = pp @ H.T @ la.inv(H @ pp @ H.T +R) #Ganancia de Kalman.

    new_mu = mup + k @ epsilon; #Nuevo estado actual.
    new_P  = (np.eye(len(P))-k @ H) @ pp; #Nueva incertidumbre.
    return new_mu, new_P, zp
```

Las ecuaciones del movimiento balístico en 2D son las siguientes:

$$v'_x \leftarrow v_x + a_x \Delta t$$

$$v'_y \leftarrow v_y + a_y \Delta t$$

$$x' \leftarrow x + v_x \Delta t + \frac{1}{2} a_x (\Delta t)^2$$

$$y' \leftarrow y + v_y \Delta t + \frac{1}{2} a_y (\Delta t)^2$$

donde x', y', v'_x, v'_y representan las variables de estado en el instante siguiente.

La forma matricial adecuada para el desarrollo del **filtro de Kalman** general es la siguiente:

$$F = \begin{bmatrix} 1 & 0 & dt & 0 \\ 0 & 1 & 0 & dt \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

La matriz F implementa la parte de la ley dinámica en la que no aparecen los términos de la aceleración. Estos términos se añaden con la matriz B, la de acción sobre el sistema. No se puede juntar todo en una misma matriz porque la matriz de acción es un término aditivo (no multiplica a ningún elemento del estado) y supondría que la dinámica del sistema no fuera puramente lineal.

$$B = \begin{bmatrix} dt^2/2 & 0 \\ 0 & dt^2/2 \\ dt & 0 \\ 0 & dt \end{bmatrix}$$

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

H es la forma matricial de expresar que observamos la posición (pero no la velocidad).

$$\mu = \begin{bmatrix} x & y & vx & vy \end{bmatrix}$$

$$P = \begin{bmatrix} i_x & 0 & 0 & 0 \\ 0 & i_y & 0 & 0 \\ 0 & 0 & i_{vx} & 0 \\ 0 & 0 & 0 & i_{vy} \end{bmatrix}$$

La matriz P determina la incertidumbre inicial tanto en posición como velocidad.

$$Q = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} * ruidoModelo$$

El ruido del modelo es un valor pequeño, determina las pequeñas perturbaciones del modelo balístico.

$$R = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} * ruidoObservacion$$

R es el ruido del proceso de toma de imágenes. Los valores de ruidos son ajustados empíricamente.

La observación z es un punto de la imagen: $z = (x_o, y_o)$.

u es el valor de la aceleración.

$$u = \begin{bmatrix} a_x \\ a_y \end{bmatrix} = \begin{bmatrix} 0 \\ -9{,}8\gamma \end{bmatrix}$$

donde γ es el factor de conversión pixel-metros a la distancia del objeto.

Es el momento de unir las partes de segmentación y tracking junto con la implementación del filtro. En cada frame que sea posible obtener un punto de observación mediante **Camshift**, este será pasado al algoritmo de **filtro de Kalman** para actualizar su estado en incertidumbre.

En el caso de que no haya observación el filtro generará un nuevo estado estimado a partir de los datos anteriores. A su vez, lo que se va a realizar es una predicción de estados posteriores al actual, veremos que obtendremos una posición estimada y una incertidumbre. Cuanto más lejos estemos del estado actual veremos que la incertidumbre es más grande, a su vez, cuando tengamos un mayor número de observaciones veremos que esa incertidumbre es menor.

En el caso en el que el objeto desaparece de la escena, la predicción actual se mantendrá hasta volver a tener una nueva observación que pueda repercutir en ella.

Vamos a ver una serie de ejemplos en funcionamiento:

En la primera imagen al haber pocas mediciones la incertidumbre de la predicción es muy grande, tiene algo de idea de por donde puede ser pero no es muy concreto.

En la segunda ya con más mediciones es capaz de acotar la posición que ocupará el objeto en el futuro.

En la última imagen vemos que la predicción ha sido correcta y el objeto siguió la trayectoria predicha.

Realizamos los mismos pasos pero ahora con un video en que se pierde la bola para ver si es capaz de predecir de manera correcta:

Vemos que la trayectoria se ajusta en gran medida a la predicción. También se ha implementado la detección de rebotes y el cálculo de la nueva trayectoria como se puede ver en la última imagen. Enlace a video demostración.

El objetivo de esta sección, unir las partes anteriores y conseguir la predicción, se ha conseguido de manera satisfactoria. El siguiente y último paso es el de llevar la implementación a una aplicación relacionada con el deporte.

Como apunte final, si quisiéramos realizar el cálculo de trayectorias en 3D al tratarse de un modelo de observación no lineal se podría utilizar una clase de filtro llamado UKF[16], aunque esto se dejará como posibles trabajos futuros.

6.5. Sistema en ejecución.

Para llevar acabo el objetivo último, el de el desarrollo de una aplicación orientada al deporte, hemos decidido utilizar los pasos anteriores para la creación de un predictor de trayectorias en lanzamientos de baloncesto.

Como ya hemos visto las técnicas a utilizar pasaremos directamente a enseñar el sistema en ejecución:

Al principio la incertidumbre es bastante grande, pero se empieza a atisbar por donde pasará la pelota.

Vemos que la pelota termina realizando la trayectoria predicha.

Lanzamiento a tablero.

Lanzamiento fuera.

Como en las otras secciones añadimos un video de demostración de diferentes lanzamientos. El objetivo de conseguir una aplicación de visión artificial orientada al deporte se ha conseguido y con ello el fin del trabajo.

7. Conclusiones y vías futuras.

7.1. Conclusiones.

Durante la fase de análisis se establecieron unos objetivos a cumplir para la fase de diseño. El primer objetivo consistía en un estudio y aplicación de técnicas de visión artificial y análisis de imágenes. Este objetivo se ha cumplido satisfactoria mente en las partes de segmentación y tracking, aunque como ya comomentamos, cada aplicación y escena puede ser diferente y estas mismas técnicas, o bien, habría que aplicarlas de diferente manera, o bien, habría que aplicar otras distitntas. Al realizar una aplicación de visión artificial hay que estudiar muy bien el escenario para conseguir un buen resultado. Pero en general estamos bastante contentos con el resultado obtenido en esta parte.

El siguiente objetivo era el de aplicación de técnicas de predicción. Hemos demostrado que mediante la utilización de un filtro de Kalman que se puede conseguir una predicción de estados futuros con alguna incertidumbre. Esta parte suponía un reto en el trabajo y también podemos concluir que se ha resuelto de manera satisfactoria.

Por último el desarrollo y aplicación a una situación real. Aunque es una aplicación sencilla nos da una imagen del tipo de aplicaciones que se podrían conseguir y por ello damos el objetivo también como cumplido.

En general la satisfacción con el trabajo es bastante grande ya que se ha tratado un tema bastante interesante como es la visión artificial, enfrentado a un problema real y obtenido unos conocimientos más allá de las competencias del grado.

7.2. Posibles trabajos futuros.

Para trabajos futuros tenemos distintas líneas que se podrían abordar. Utilizando los resultados de este mismo trabajo se podría conseguir la aplicación a otros sistemas parecidos, un ejemplo podría ser distintos tipos de lanzamientos, desde lanzamientos de jabalina o disco hasta calculo de trayectorias en lanzamientos en fútbol americano, rugby o cualquier deporte en el que las condiciones sean comunes a las que se han tratado en este trabajo.

Como continuación de este trabajo se podría también buscar la manera de conseguir un análisis computacionalmente más rápido mediante el uso de otras técnicas. Este trabajo se ha centrado en el análisis de videos, pero también se podría adaptar para el uso en tiempo real.

Otra línea sería continuar este proyecto desde el punto de vista del cálculo de trayectorias en 3D, o bien, mediante uso de distintas vistas y cámaras, o bien, aplicando un tipo de técnicas especiales para sistemas no lineales.

También este trabajo puede servir como punto de inicio para la realización de otras aplicaciones que no traten el cálculo de trayectorias, pero, que se sirvan para el análisis de deportes utilizando las partes de segmentación y seguimiento. Una posible aplicación podría ser la de seguimiento de la posición de los jugadores en un terreno de juego.

8. Anexo.

8.1. Código.

```
import cv2 as cv
import numpy as np
from scipy.ndimage.filters import gaussian_filter
from umucv.kalman import kalman, ukf
import umucv.htrans as ht

REDU = 8

def rgbh(xs,mask):
    def normhist(x): return x / np.sum(x)

    def h(rgb):
      return cv.calcHist([rgb]
                         , [0, 1, 2]
                         , imCropMask
                         , [256//REDU, 256//REDU, 256//REDU]
                         , [0, 256] + [0, 256] + [0, 256]
                         )

    return normhist(sum(map(h, xs)))

def smooth(s,x):
    return gaussian_filter(x,s,mode='constant')

bgsub = cv.createBackgroundSubtractorMOG2(500, 60, True)  #El valor de
   threshold podria variar(60)
cap = cv.VideoCapture("Videos/l1.MOV")
key = 0

kernel = np.ones((3,3),np.uint8)
crop = False
camshift = False

termination = (cv.TERM_CRITERIA_EPS | cv.TERM_CRITERIA_COUNT, 10, 1)

```

```
font = cv.FONT_HERSHEY_SIMPLEX

pause= False
###################### Kalman inicial ########################

# estado que Kalman va actualizando. Este es el valor inicial
degree = np.pi/180
a = np.array([0, 900])


#fps = 60
fps = 120
dt  = 1/fps
t = np.arange(0,2.01,dt)
noise = 3

F = np.array(
    [1, 0,  dt,  0,
     0, 1,  0, dt,

     0, 0,  1,  0,
     0, 0,  0,  1 ]).reshape(4,4)

B = np.array(
         [dt**2/2, 0,
          0,       dt**2/2,
          dt,      0,
          0,       dt      ]).reshape(4,2)


H = np.array(
    [1,0,0,0,
     0,1,0,0]).reshape(2,4)



             # x, y, vx, vy
mu = np.array([0,0,0,0])
            # sus incertidumbres
P  = np.diag([1000,1000,1000,1000])**2
```

```
#res = [(mu,P,mu)]
res=[]
N = 15  # para tomar un tramo inicial y ver que pasa si luego se pierde
    la observacion

sigmaM = 0.0001   # ruido del modelo
sigmaZ = 3*noise  # deberia ser igual al ruido de media del proceso de
   imagen. 10 pixels pje.

Q = sigmaM**2 * np.eye(4)
R = sigmaZ**2 * np.eye(2)

listCenterX=[]
listCenterY=[]
listpuntos=[]
while(True):
    key = cv.waitKey(1) & 0xFF
    if key== ord("c"): crop = True
    if key== ord("p"): P  = np.diag([100,100,100,100])**2
    if key==27: break
    if key==ord(" "): pause =not pause
    if(pause): continue

    ret, frame = cap.read()
    #frame=cv.resize(frame,(800,600))
    frame=cv.resize(frame,(1366,768))

    bgs = bgsub.apply(frame)

    bgs = cv.erode(bgs,kernel,iterations = 1)
    bgs = cv.medianBlur(bgs,3)
    bgs = cv.dilate(bgs,kernel,iterations=2)

    bgs = (bgs > 200).astype(np.uint8)*255
    colorMask = cv.bitwise_and(frame,frame,mask = bgs)


    if(crop):
      fromCenter= False
      img = colorMask
```

```
        r = cv.selectROI(img, fromCenter)
114     imCrop = img[int(r[1]):int(r[1]+r[3]), int(r[0]):int(r[0]+r[2])]
        crop = False
116     camshift = True
        imCropMask = cv.cvtColor(imCrop, cv.COLOR_BGR2GRAY)
118     ret,imCropMask = cv.threshold(imCropMask,30,255,cv.THRESH_BINARY)
        his = smooth(1,rgbh([imCrop],imCropMask))
120
        roiBox = (int(r[0]), int(r[1]),int(r[2]), int(r[3]))
122     cv.destroyWindow("ROI selector")

124     if(camshift):

126       cv.putText(frame,'Center roiBox',(0,10), font, 0.5,(0,255,0),2,cv
             .LINE_AA)
          cv.putText(frame,'Estimated position',(0,30), font,
             0.5,(255,255,0),2,cv.LINE_AA)
128       cv.putText(frame,'Prediction',(0,50), font, 0.5,(0,0,255),2,cv.
             LINE_AA)

130       rgbr = np.floor_divide( colorMask , REDU)
          r,g,b = rgbr.transpose(2,0,1)
132       l = his[r,g,b]
          maxl = l.max()
134       aa=np.clip((1*l/maxl*255),0,255).astype(np.uint8)
          #cv.imshow("Backprojection", cv.resize(aa,(400,250))) #
             Backprojection
136


138       # Aplicamos camshift y dibujamos en la pantalla los puntos
          (rb, roiBox) = cv.CamShift(l, roiBox, termination)
140       cv.ellipse(frame, rb, (0, 255, 0), 2)

142       ##########Kalman filter############

144       xo=int(roiBox[0]+roiBox[2]/2)
          yo=int(roiBox[1]+roiBox[3]/2)
146       error=(roiBox[3])
            #Calculos centro del roibix
148
```

```
      #print(yo,error)
150
      if(yo<error or bgs.sum()<50 ):
152     mu,P,pred= kalman(mu,P,F,Q,B,a,None,H,R)
        m="None"
154     mm=False

156   else:
        mu,P,pred= kalman(mu,P,F,Q,B,a,np.array([xo,yo]),H,R)
158     m="normal"
        mm=True
160
      if(mm):
162     listCenterX.append(xo)
        listCenterY.append(yo)
164
      listpuntos.append((xo,yo,m))
166   res += [(mu,P)]

168   ##### Prediccion #####
      mu2 = mu
170   P2  = P
      res2 = []
172
      for _ in range(fps*2):
174       mu2,P2,pred2= kalman(mu2,P2,F,Q,B,a,None,H,R)
          res2 += [(mu2,P2)]
176
      xe = [mu[0] for mu,_ in res]
178   xu = [2*np.sqrt(P[0,0]) for _,P in res]
      ye = [mu[1] for mu,_ in res]
180   yu = [2*np.sqrt(P[1,1]) for _,P in res]


182


184   xp=[mu2[0] for mu2,_ in res2]
      yp=[mu2[1] for mu2,_ in res2]
186   xpu = [2*np.sqrt(P[0,0]) for _,P in res2]
      ypu = [2*np.sqrt(P[1,1]) for _,P in res2]
188
```

```
190       for n in range(len(listCenterX)): # centro del roibox
            cv.circle(frame,(int(listCenterX[n]),int(listCenterY[n])),3,
               (0, 255, 0),-1)
192
          for n in [-1]:#range(len(xe)): # xe e ye estimada
194         #incertidumbre = (xu[n]*yu[n])
            #cv.circle(frame,(int(xe[n]),int(ye[n])),int(incertidumbre),
               (255, 255, 0),-1)
196         incertidumbre=(xu[n]+yu[n])/2
            cv.circle(frame,(int(xe[n]),int(ye[n])),int(incertidumbre),
               (255, 255, 0),1)
198
          for n in range(len(xp)): # x e y predicha
200         incertidumbreP=(xpu[n]+ypu[n])/2
            cv.circle(frame,(int(xp[n]),int(yp[n])),int(incertidumbreP),
               (0, 0, 255))
202
          print("Lista de puntos\n")
204       for n in range(len(listpuntos)):
            print(listpuntos[n])
206
          if(len(listCenterY)>4):
208         if((listCenterY[-5] < listCenterY[-4]) and(listCenterY[-4] <
               listCenterY[-3]) and (listCenterY[-3] > listCenterY[-2]) and
                (listCenterY[-2] > listCenterY[-1])):
              print("REBOTE")
210           listCenterY=[]
              listCenterX=[]
212           listpuntos=[]
              res=[]
214           mu = np.array([0,0,0,0])
              P  = np.diag([100,100,100,100])**2
216


218     cv.imshow('ColorMask',colorMask)
        #cv.imshow('ColorMask',cv.resize(colorMask,(800,600)))
220     cv.imshow('mask', bgs)
        #cv.imshow('Frame',cv.resize(frame,(800,600)))
222     cv.imshow('Frame', frame)
```

8.2. Enlaces a videos.

https://www.youtube.com/watch?v=vg1kAZxveQg

https://www.youtube.com/watch?v=Mg40mE0zwBA

https://www.youtube.com/watch?v=65Tq_nlP788

https://www.youtube.com/watch?v=YlPOTxYvt6U

https://www.youtube.com/watch?v=MxwVwCuBEDA

9. Bibliografía.

Referencias

[1] Richard Szeliski (2010). Computer Vision: Algorithms and Applications. Springer-Verlag. ISBN 978-1848829343.

[2] Alberto Ruiz, Apuntes asignatura visión artificial, http://dis.um.es/profesores/alberto/material/percep.pdf

[3] Rafael C. Gonzalez, Richard E. Woods, Digital Image Processing.

[4] Documentación OpenCV, http://docs.opencv.org/3.3.0/d6/d00/tutorial_py_root.html

[5] The British Machine Vision Association and Society for Pattern Recognition Retrieved, February 20, 2017, http://www.bmva.org/visionoverview

[6] Draw.io, Generador de diagramas online, https://www.draw.io

[7] Wikipedia, Trayectoria Balística, https://es.wikipedia.org/wiki/Trayectoria_bal%C3%ADstica

[8] Wikipedia, Segmentación, https://es.wikipedia.org/wiki/Segmentacion_(procesamiento_de_imagenes)

[9] OpenCV, Background Substraction, http://docs.opencv.org/master/db/d5c/tutorial_py_bg_subtraction.html#gsc.tab=0

[10] Hanzi Wang and David Suter, A RE-EVALUATION OF MIXTURE-OF-GAUSSIAN BACKGROUND MODELING,Monash University,Victoria, Australia

[11] OpenCV, Erode and Dilate, http://docs.opencv.org/3.0-beta/doc/py_tutorials/py_imgproc/py_morphological_ops/py_morphological_ops.html

[12] Jean Serra, Image Analysis and Mathematical Morphology ISBN 0-12-637240-3 (1982)

[13] OpenCV, MedianBlur, http://docs.opencv.org/3.1.0/d4/d13/tutorial_py_filtering.html

[14] Wikipedia, Camshift, https://fr.wikipedia.org/wiki/Camshift

[15] Bradski, G.R., Real time face and object tracking as a component of a perceptual user interface,.ªpplications of Computer Vision, 1998. WACV '98. Proceedings., Fourth IEEE Workshop on , vol., no., pp.214,219, 19-21 Oct 1998.

[16] Kalman, R.E. (1960). .ª new approach to linear filtering and prediction problems"(PDF). Journal of Basic Engineering. 82 (1): 35–45. doi:10.1115/1.3662552

[17] Wikipedia, Filtro de Kalman, https://es.wikipedia.org/wiki/Filtro_de_Kalman

[18] Julier, Simon J.; Uhlmann, Jeffrey K. (1997). .ª new extension of the Kalman filter to nonlinear systems"(PDF). Int. Symp. Aerospace/Defense Sensing, Simul. and Controls. Signal Processing, Sensor Fusion, and Target Recognition VI. 3: 182. Bibcode:1997SPIE.3068..182J. doi:10.1117/12.280797.