



Práctica 2

NOTA: el código fuente completo se puede consultar en los ficheros adjuntos a este documento. Cada apartado tiene su correspondiente método. Lo mismo para las imágenes.

Se ha usado la versión 2.4.3 de OpenCV

Bernardo Miquel Riera

Juan Gabriel Florit Gomila

ÍNDICE

A)Captura (módulo 1).....	3
B)Segmentación (módulo 5).....	4
C)Ext. de características y reconocimiento (módulo 6).....	7
D)Ext. conclusiones y posibles limitaciones y mejoras.....	11

A) Captura (módulo 1)

Objetivos:

- ESCENARIO: campo de fútbol con jugadores, cancha de básquet, pista de tenis, pista de pádel, fútbol americano, rugby. (cualquier deporte de equipo, en grupo).
- Realizar una captura básica de imágenes a través de cámaras propias o bien en búsqueda de bases de datos en Internet o de referencia.

Procesos realizados:

Hemos Utilizar el buscador de imágenes de Google para encontrar imágenes relacionadas con el fútbol. Se han elegido imágenes diversas para probar mejor los algoritmos. Algunas de ellas son sacadas de un videojuego de fútbol y nos resultan útiles ya que todas tienen la misma perspectiva y no hay ruido.

Ficheros de imágenes:

- entrada
 - entrada/furbol1.jpg
 - entrada/furbol2.jpg
 - entrada/furbol3.jpg
 - entrada/furbol4.jpg
 - entrada/furbol5.jpg
 - entrada/furbol6.jpg
 - entrada/furbol7.jpg
 - entrada/furbol8.jpg
 - entrada/furbol9.jpg
 - entrada/furbol10.jpg
 - entrada/furbol11.jpg

B) Segmentación (módulo 5)

Objetivos:

- Segmentar la imagen de manera que se separe los elementos móviles (pelota, jugadores, etc.) de fondo de los terrenos.
- Detectar los límites de los terrenos (segmentos de rectas, porterías, etc.).

Procesos realizados:

Como el deporte elegido es el fútbol, deberemos reconocer a los jugadores de ambos equipos, la pelota y los límites de los terrenos.

JUGADORES

Hay dos equipos de fútbol jugando a la vez un partido. Para reconocer a los jugadores y a la vez distinguirlos por equipo, tendremos que segmentar por el color de sus camisetas. En este punto debemos hacer ciertas suposiciones: Suponemos, como ocurre en realidad, que los dos equipos llevaran camisetas suficientemente diferentes en colores (segunda equipación, si hace falta); y suponemos que las camisetas no serán de un tono verdoso similar al del terreno de juego o sería muy complicada la detección.

Primero pasamos la imagen a HSV, ya que será más fácil obtener los tonos:

```
cvtColor(imatge_RGB, imatge_HSV, CV_BGR2HSV);
```

Hue values of basic colors

Orange 0-22

Yellow 22- 38

Green 38-75

Blue 75-130

Violet 130-160

Red 160-179

En base a lo anterior y jugando con las demás bandas S y V, tratamos a los jugadores de ambos equipos según su color de camiseta (la función `inRange` se encarga de filtrar sólo los colores deseados):

```
if (equip1) {
    inRange(imatge_HSV, Scalar(75,150,40), Scalar(130,255,255), imatge_thresh);
} else {
    inRange(imatge_HSV, Scalar(22,120,100), Scalar(38,255,255), imatge_thresh);
}
```

Después, se aplica un cierre a la imagen (dilatación + erosión):

```
dilate(imgatge_thresh, imgatge_thresh, elemStruc);  
erode(imgatge_thresh, imgatge_thresh, elemStruc);
```

Y finalmente, por cada equipo, contamos sus jugadores en base al número de contornos encontrados por findContours. La imagen que se ha usado para probar es futbol7.jpg. Hemos contado los jugadores del equipo azul y nos ha devuelto que hay 7, que coincide con el número real. El resultado es bastante bueno para las demás pruebas que hemos hecho, pero no siempre devuelve el número exacto (puede confundirse con vallas publicitarias, público, marcadores...).

PELOTA

Pasamos la imagen a escala de grises, aplicamos canny para la detección de contornos. Luego le pasamos la imagen a HoughCircle, que ajustando sus parámetros según lo grande o pequeño que sea el balón de la imagen, nos devuelve un vector con todos los círculos detectados. Aprovechamos ese vector para pintar sobre la imagen original los círculos detectados.

```
HoughCircles(imatge, circles, CV_HOUGH_GRADIENT, 10, 1, 200, 400, 10, 50);
```

Se ha hecho la prueba con futbol9.jpg, dónde nos ha detectado correctamente el balón. Las otras pruebas que hemos hecho han sido dispares en cuanto a calidad. A veces se detectaban varios balones debido a que canny no había dado buenos resultados. Otras veces no se detectaba ninguno ya que era demasiado pequeño o lo tapaban los jugadores.

LÍMITES CAMPO

Lo haremos combinando parte de las dos soluciones anteriores. Como antes, pasamos la imagen RGB original al modelo HSV y luego aplicaremos la función inRange para filtrar el blanco (suponemos que las líneas que delimitan los límites del campo son de ese color):

```
cvtColor(imatge_RGB, imatge_HSV, CV_BGR2HSV);  
inRange(imatge_HSV, Scalar(0,0,150), Scalar(255,100,255), imatge_Thresh);
```

Después, usamos canny y acto seguido se aplica un cierre a la imagen (dilatación + erosión).

Y finalmente, se usa el operador HoughLinesP, que de manera homónima a HoughCircle devuelve las líneas detectadas. Usamos el vector con las líneas detectadas para pintar los puntos sobre la imagen original.

```
HoughLinesP(imatge_canny, Plines, 1, CV_PI/150, 40, 40, 20);
```

La prueba se ha realizado con futbol2.jpg y los resultados son regulares, al igual que con las demás imágenes. El problema es que nuestro algoritmo confunde las líneas del campo con patrones en el público y los marcadores sobre la imagen. Esto no es tan grave ya que en un sistema para la vida real, podríamos recoger la señal sin los marcadores y recortar la imagen para que saliese sólo el campo.

Ficheros de imágenes:

- Entrada:
 - entrada/futbol7.jpg
 - entrada/futbol9.jpg
 - entrada/futbol2.jpg

- Salida:
 - salida/detecciones/futbol7_deteccionjugadores_equipoazul.jpg
 - salida/detecciones/futbol9_deteccionpelotas.jpg
 - salida/detecciones/futbol2_deteccionlineas.jpg

C) Ext. de características y reconocimiento (módulo 6)

Objetivos:

- Contar el número de jugadores de cada equipo.
- Detectar el balón o balones.
- Mirar trayectorias de pelotas y jugadores.
- Definir posiciones o estrategias.
- Determinar posibles situaciones de:
 - Fuera de juego.
 - Posiciones tácticas (4-4-2, 3-5-1, etc...).
 - Goles fantasma.
 - Pelotas fuera del terreno de juego.

Procesos realizados:

Los siguientes puntos que se piden en el enunciado ya se han hecho y explicado en el anterior apartado:

- Contar el número de jugadores de cada equipo. Contamos sus jugadores en base al número de contornos encontrados por findContours.
- Detectar el balón o balones.

Nos quedan los demás puntos. Se han intentado implementar pero no se han conseguido buenos resultados. De todas formas, a continuación se explican la forma en que queríamos intentar resolverlos:

ESTRATEGÍA

Para saber de que tipo de estrategia se trata, lo que hacemos es considerar por separado los jugadores de un equipo y luego del otro. Así sabemos que estrategia/alineación lleva a cabo cada equipo.

Para detectar una alineación concreta, antes debemos saber la posición de los jugadores. Si las imágenes que analizamos tienen una perspectiva como futbol1.jpg .. futbol8.jpg, resulta más fácil, puesto que podemos suponer que no hay problema con extraer directamente las posiciones (x,y) de los jugadores. Estas posiciones no resultan difíciles de conseguir, ya que previamente hemos reconocido a los jugadores. Basta tenerlos guardados en un array y para cada uno de ellos calcular un punto que esté en el

centro de sus contornos. Ese punto será su posición (x,y).

Según lo que hemos indagado sobre alineaciones de fútbol, podemos extrapolar lo siguiente: hay un número N de jugadores distribuidos a lo largo del campo y a diferentes alturas. Unos cerca de la banda derecha, otros en el centro y otros en la banda izquierda. Entonces, podemos considerar 3 líneas imaginarias que recorren el campo de portería a portería y establecer una ventana de espacio, en la cual los jugadores que se encuentren dentro, se considerará que están alineados.



Luego basta contabilizar el número de jugadores alineados en cada ventana y mirar a que altura se encuentran para determinar si siguen una estrategia 4-4-2, 3-5-1... o ninguna.

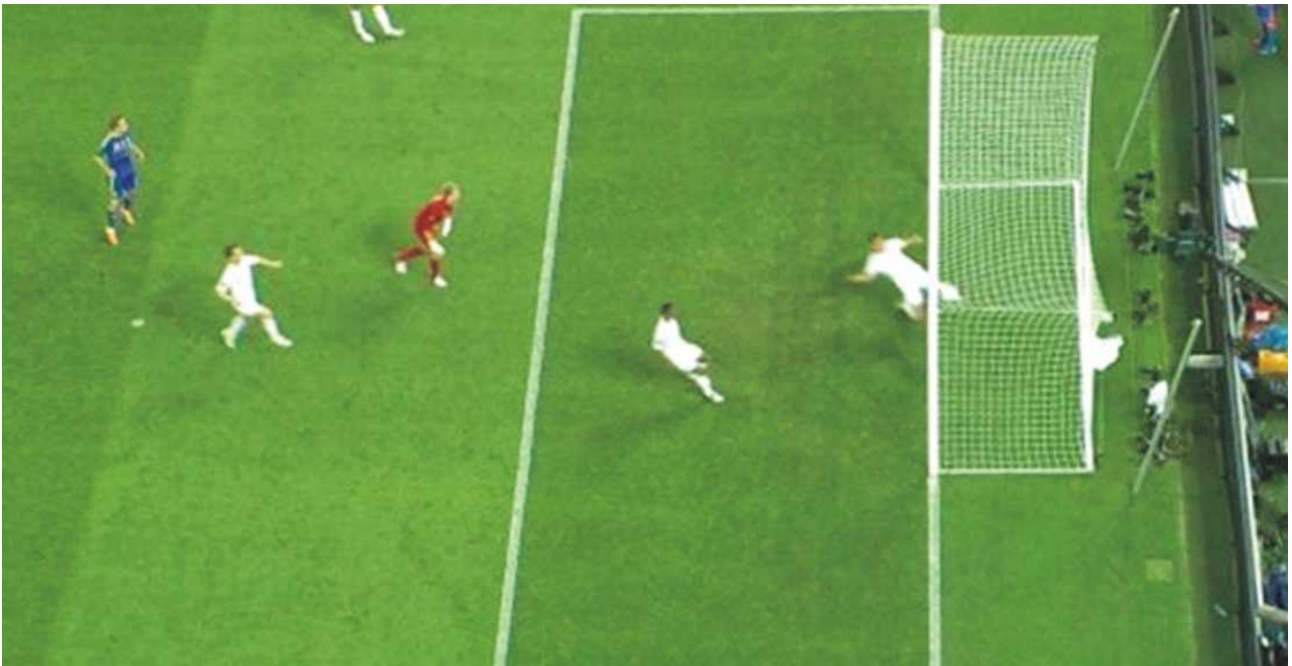
GOLES FANTASMA

La detección de goles fantasma se basa en una estrategia parecida al Hawk-Eye u ojo de halcón en tenis. Dicha técnica se encarga de asegurar que la pelota ha golpeado sobre la línea. Pero en el caso del fútbol no basta con saber si el balón está sobre la línea sino que se debe estudiar si la ha atravesado. Para comprobar dicho resultado es conveniente conocer los límites del campo y la posición del balón.

La detección de los límites del campo se ha realizado en uno de los apartados anteriores y, aunque a veces detecta límites que no debería en función de la imagen, la mayoría de veces los extremos del campo son fácilmente diferenciables ya que se pueden aproximar por rectas mediante la transformada de Hough. Dicha técnica destaca los límites y, a partir de la perspectiva y los palos de la portería se puede conocer el segmento de recta que implicaría gol si la esfera lo atraviesa.

Tal como hemos dicho, ahora conocemos los límites del campo y solamente necesitamos conocer la ubicación del balón. Para eso es necesario detectar los círculos de la imagen después aplicar umbralización y reconocer el propio balón. Dicha tarea también se ha realizado en un apartado anterior.

Una vez que tenemos la posición del balón y el segmento de recta que debe atravesar se puede, mediante la perspectiva de la imagen se puede calcular si el centro del balón está más allá del segmento. En casos más complejos y no tan claro, se puede ajustar el cálculo tanto como sea posible mediante trigonometría.



TRAYECTORÍAS

Entendemos que el análisis de trayectorias sólo es posible si tuviésemos un vídeo o por lo menos, varias imágenes consecutivas. Se trataría de hacer un tracking de los objetos (jugadores, pelota) y en base a eso calcular la trayectoria seguida e incluso inferir próximas posiciones.

FUERA DE JUEGO

Partimos de que tenemos dos arrays, cada uno de ellos con los jugadores de cada equipo. También sabemos el rol del equipo. Igualmente, sabemos la posición de la pelota. Es decir, que equipo está en el campo contrario atacando y cual está defendiendo. Volvemos a suponer que disponemos de una perspectiva como en las imágenes futbol1.jpg .. futbol8.jpg.

Según la wikipedia: “un jugador se encuentra en posición de fuera de juego si se

encuentra más cerca de la línea opuesta que el balón y el penúltimo adversario”. Así pues , -y simplificando-, debemos identificar el jugador de cada equipo que está más cerca de la portería. La portería se hallará a la derecha o izquierda, pero recordemos que suponemos que sabemos en que campo nos encontramos. Así resulta fácil, recorrer los dos arrays y que cada uno devuelva el jugador que se encuentra más a la derecha o izquierda según el campo.

Ahora se comprueba si el jugador atacante más avanzado se encuentra por delante del jugador defensor más avanzado. De ser así, dicho jugador se haya en fuera de juego. Si hay más jugadores del equipo atacante por delante del defensor más avanzado, ellos también se encuentran en fuera de juego. Mediante análisis de trayectoria de la pelota, podríamos saber si se ha lanzado hacia uno de los jugadores en fuera de juego y por tanto el árbitro pitaría el fuera de juego.

D) Ext. conclusiones y posibles limitaciones y mejoras.

Objetivos:

- Determinar posibles situaciones de:
 - Cambios de iluminación.
 - Velocidad del movimiento.
 - Diferentes tipos de superficies.
 - Sistemas complementarios no basados en VPO.

Procesos realizados:

Esta práctica ha servido para profundizar más en OpenCV. Mientras que la práctica 1 era todo más desde un punto de vista teórico, en esta nos hemos servido de lo aprendido anteriormente para aplicarlo en el ámbito de una aplicación real, como puede ser un sistema de visión para analizar partidos de fútbol. También nos hemos dado cuenta de la gran complejidad que supone hacer trabajos medianamente ambiciosos. A parte de los problemas ya citados en la memoria, explicaremos algunas hipotéticas situaciones:

CAMBIOS DE ILUMINACIÓN

Como se ha visto en la teoría, la iluminación es clave durante el proceso de captura, puesto que de ellas depende la calidad de los resultados tras la aplicación de las diferentes etapas.

Si se producen cambios en la iluminación entre imágenes, el sistema de VPO debería ser capaz de adaptarse a ellos. Decir también, que los sistemas de VPO trabajan dentro de ciertos límites. En casos de iluminaciones muy malas, no se podrá hacer nada para paliar tal situación.

VELOCIDAD DEL MOVIMIENTO

Entendiendo que por velocidad del movimiento, se refiere a la velocidad a la que se mueve la cámara u objeto en un vídeo, podemos afirmar que cuanto mayor sea esta velocidad, más complicado será aplicar técnicas de VPO con buenos resultados.

Es por ello, que si trabajásemos con vídeo, sería un aspecto clave a considerar.

DIFERENTES TIPOS DE SUPERFICIES

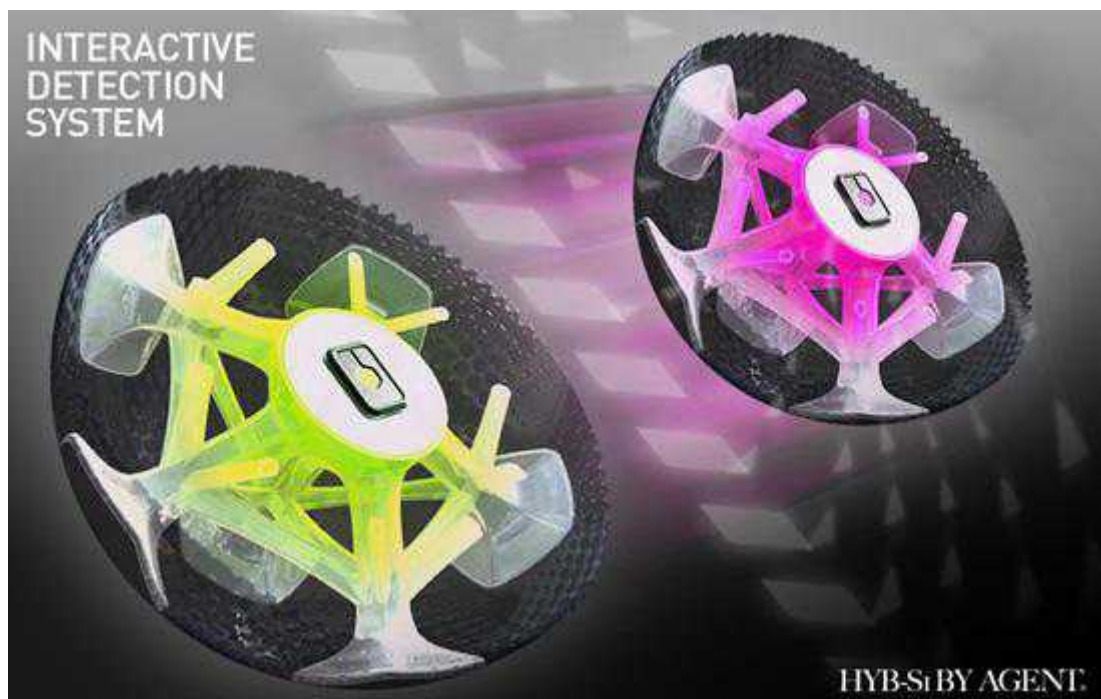
Si la superficie de las imágenes siempre es la misma (en nuestro caso, el césped del campo de fútbol), una vez ajustados los parámetros para saber distinguirla del resto, ya no vamos a tener que preocuparnos por ella. No obstante, si de un campo de fútbol a otro (o si también analizamos las canchas de otros deportes) se producen diferencias sustanciales en las superficies, la cosa se complica. El proceso de segmentación resultará más tedioso y se pueden producir más errores.

SISTEMAS COMPLEMENTARIOS NO BASADOS EN VPO

Los sistemas de VPO, por muchos avances que se hayan producido en los últimos años, no dejan de sufrir de ciertas limitaciones para según que tipo de aplicaciones. Estos sistemas de VPO pueden complementarse con otro tipo de sistemas que aporten más información a la aplicación y esta lleve a cabo su desempeño más eficientemente.

Un sistema no basado en VPO son los sensores, que pueden aportar la información que perciben a la aplicación. A continuación mostramos algunos ejemplos:

- Balón CTRUS. Este en concreto es un prototipo, pero existen productos similares en el mercado. Este balón cuenta con tecnología GPS y RFID (Radio Frequency Identification). La primera tecnología serviría para geolocalizar en todo momento el balón y la segunda, entre otros, para conocer si el balón ha traspasado la portería. También cuenta con acelerómetros.



- Adidas 1. Se trata de una zapatilla que incluyen un procesador encargado de realizar cálculos sobre diferentes parámetros: ajustes del terreno y ajustes de la zapatilla. Las posibilidades son muy amplias. Por ejemplo, poder seguir a los jugadores mediante añadir GPS.

