ANÁLISIS DE VIDEOS DE FÚTBOL

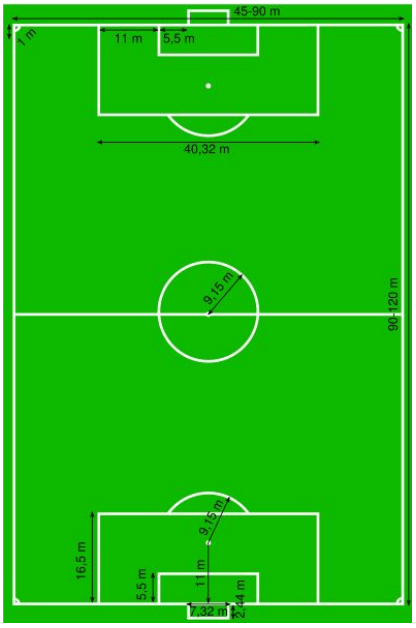
Emmanuel Antonio Cuevas
Universidad de Guanajuato
CIMAT

En este proyecto desarrollamos un analizador de videos de fútbol. Para lograr esta meta hemos tomado como entrada videos panorámicos de encuentros de fútbol. A partir de esto, hemos logrado realizar algunas estadísticas con información extraída de los jugadores y las dimensiones del campo.

Video y calibración

El video, como ya se mencionó, es un video de toma panorámica donde se alcanza a ver todo el campo de juego. El video fue tomado con dos cámaras, cada una de ellas filmaba una mitad del campo y luego el video fue puesto junto para tener una sola toma.

Aprovechando que es posible ver todo el campo y las cámaras son estáticas, entonces podemos extraer información de esto.



El el fútbol profesional, se deben cumplir algunas restricciones sobre las medidas del campo de juego.
El *largo y ancho de la cancha* no poseen un tamaño fijo definido, ambas medidas solo tiene un rango de valores en el que pueden caer. Para el largo de la cancha, tiene que ser por lo menos de **90 m** y no debe exceder los **120 m** de longitud. Para el ancho del campo debe ser de al menos **45 m** hasta **90 m**. Por tanto, al no ser una medidas fijas, por el momento, no nos serán muy útiles. Entonces, nos fijamos en los dimensiones del área grande y área chica. El *área grande* tiene una longitud de **16.5 m** y **40.32 m** de ancho, donde el tamaño de la portería es de **7.32 m** y se encuentra centrada una a cada extremo del campo. El *área chica* posee dimensiones de **5.5 m** de largo y **18.32 m** de ancho. A **11 m** del centro de la portería, sobre el área grande, se encuentra el manchón de penal el cual es centro de una semicircunferencia de **9.15 m** de radio que solo se dibuja la parte que queda fuera del área grande. Estas medidas, dado que son fijas para todos los campos de fútbol profesional, son las que usaremos como referencia.

Para extraer estos puntos se hace de manera manual, el programa mostrará una venta con el primer frame del video y el usuario deberá seleccionar los puntos del área más cercana al eje de la **Y** (en este caso más a la izquierda) como se muestra en la imágen. Seguido de esto, se abrirá otra ventana mostrando el primer frame con los puntos que se seleccionaron previamente del área 1, en este paso el usuario deberá seleccionar los puntos del área dos en el mismo orden en que se seleccionaron los del primer área.
Luego de esto, estimamos la longitud del campo. Llamaremos A_{1i} a el i-ésimo punto seleccionado del área de la izquierda y A_{2i} al i-ésimo del área de la derecha, de acuerdo al orden en que se eligieron. Sabemos que A_{11} y A_{21} se encuentran sobre un mismo segmento de línea que es paralelo a las líneas del costado del campo, y sobre esta línea se encuentran los puntos A_{19} y A_{29} . También sabemos que

la distancia entre A_{11} y A_{19} es de 16.5 metros, dado que la imágen no es muy grande supondremos que las distancias sobre la misma línea paralela son proporcionales, por lo tanto podemos hacer una estimación de la longitud del campo

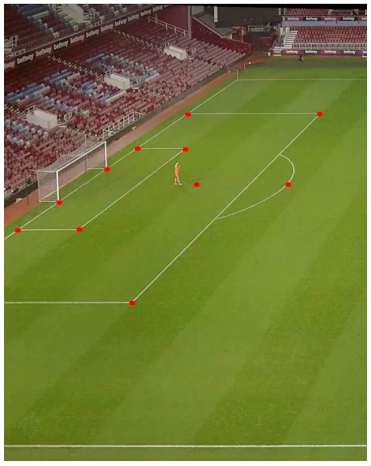
$$L_1 \approx d(A_{11}, A_{21}) * d(A_{11}, A_{19})/16.5$$
$$L_2 \approx d(A_{12}, A_{22}) * d(A_{12}, A_{16})/5.5$$
$$L_3 \approx d(A_{15}, A_{25}) * d(A_{15}, A_{17})/5.5$$

donde d es la distancia entre puntos.

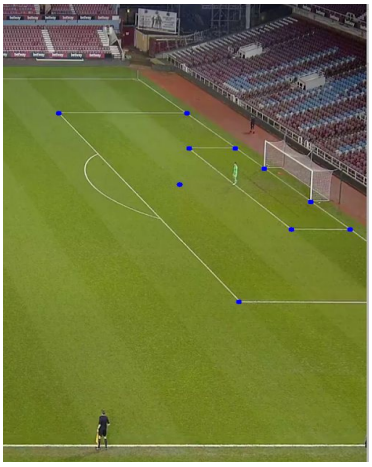
Tenemos 3 estimaciones de la longitud, escogemos la que minimice el error,

$$L \text{ t.q. minimice } e = (L_1 - L)^2 + (L_2 - L)^2 + (L_3 - L)^2$$

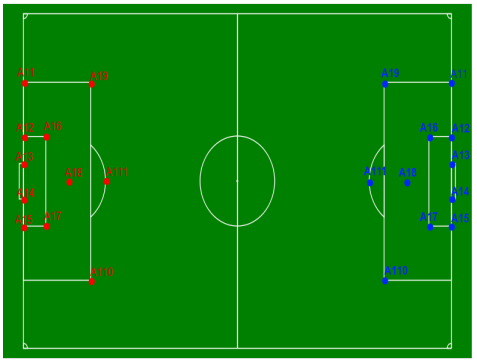
En las ejecuciones se tiene una estimación entre 103 y 105 metros lo cual está dentro del rango de dimensiones reglamentarias y la mayoría de los campos tiene una longitud de 100 a 110 metros por lo cual podríamos decir que la estimación es buena.
Dada la estimación de la longitud del campo podemos estimar también las posiciones de los puntos del área 2 (azules). Con más puntos la estimación de la homografía es mejor, usamos todos estos puntos para calcular la matriz de homografía entre el modelo plano y la imagen.



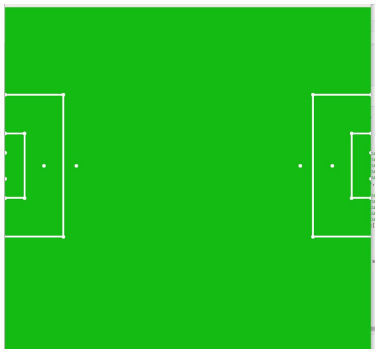
Los puntos rojos son los puntos seleccionados de el área 1.



Los puntos azules son los puntos seleccionados del área 2.



Orden en que se seleccionan los puntos de las áreas.



Modelo plano de la cancha.

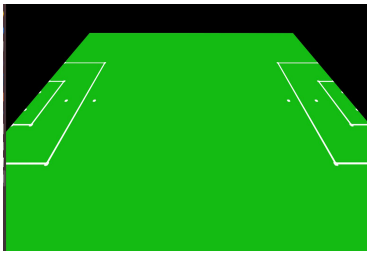
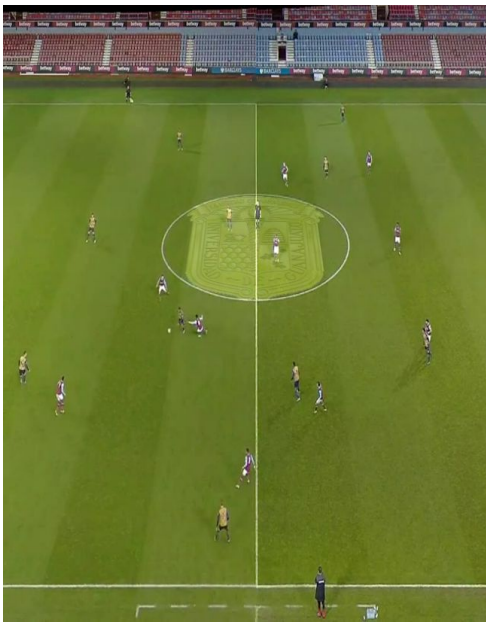


Imagen del modelo proyectado usando la homografía entre el modelo y los puntos seleccionados

Proyección sobrepuesta. Podemos ver que la aproximación encaja bien por lo tanto la aproximación es buena.

Proyección de imágenes



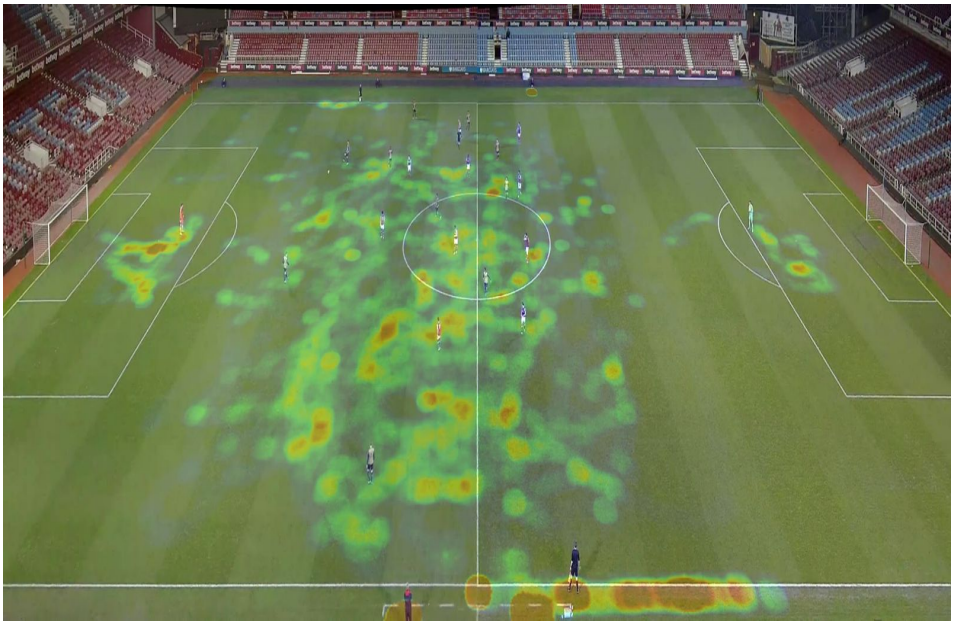
Ya tenemos una matriz H de homografía del plano hacia la imagen del video. Para proyectar imágenes al centro del campo primero calculamos en donde está el punto central, que tiene coordenadas en el plano $x = L/2$ y para y tomamos el mismo valor que se eligió para la coordenada en ***y*** del punto penal.

Conociendo el punto central en el modelo plano, para colocar una imagen de ancho y largo (w, h), colocamos el vértice superior izquierdo en la coordenada $(x - w/2, y - h/2)$. Luego proyectamos la imagen con la matriz de homografía H sobre el campo con una suma pesada. Para cuestiones de estética redimensionamos la imagen de forma que entre en el círculo central. El círculo central tiene un diámetro de 18.3 metros, entonces redimensionamos la imagen tal que su longitud es igual a el diámetro y el ancho conserva el aspecto de ratio entre largo y ancho de la imagen original.

Presionando la tecla **s** al ejecutar el código este mostrará la imagen en el centro del campo como se hace en la figura.

Mapa de calor

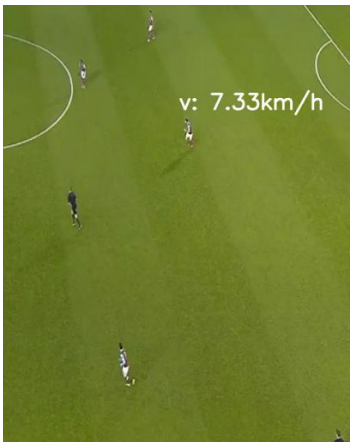
Desarrollamos un mapa de calor, que va coloreando las zonas de acuerdo a la cantidad de tiempo que han estado los jugadores ahí. Esto es usado por algunos equipos de fútbol profesional para hacer estadísticas y diseñar estrategias de equipo. La forma de desarrollarlo fue detectando los jugadores que están en la imagen (sobre esto se hablará más adelante) y estimar su posición en el modelo plano, luego agregar intensidad a las zonas en donde se encuentra cada jugador. Luego, proyectamos este mapa de calor hacia el frame actual usando el mapa de color COLORMAP_JET de la librería OpenCV.



Presionando la tecla **h**, se mostrará el frame con el mapa de calor hasta ese momento.

Sobre esta característica se puede notar que entre más cerca estén los jugadores la detección de ellos es mejor, podemos darnos cuenta en las regiones donde se encuentran los jueces de línea. El juez de línea más cercano podemos ver que la región que cubre es recta mientras que el juez de línea del fondo tiene ciertas oscilaciones en la detección.

Velocidad de jugadores



También desarrollamos la herramienta para estimar la velocidad de los jugadores. Para desarrollar esta herramienta seleccionamos un jugador al azar, luego inicializamos una instancia de tracking sobre el jugador, para esto usamos **MIL tracker** implementado en la librería OpenCV, escogí este algoritmo porque fue el que demostró mayor robustez en la tarea pues los demás perdían fácilmente el objetivo.

Estimamos la posición del jugador entre frames consecutivos, proyectamos estos puntos al modelo plano para estimar su posición en el modelo plano, y luego anotamos su velocidad en una tabla donde guardamos las últimas 5 velocidades reportadas, de estas 5 velocidades tomamos la mediana con la finalidad de descartar picos en las velocidades, al final promediamos con la velocidad anterior.

Al presionar la tecla **v** se muestra la velocidad de un jugador al azar, su velocidad se muestra sobre el jugador como se muestra en la imagen.

Aún así se logran ver alguno picos en las velocidades reportadas. Intenté usar filtro de Kalman pero no lograba hacer que se ejecutara adecuadamente, se cortaba el programa cuando se hacía la predicción del siguiente posición, esto es algo que se puede mejorar.

Detección de objetos

Una herramienta importante, aunque no se muestra explícitamente, es la detección de los jugadores en el campo. Lo mejor hubiese sido tomar un modelo y entrenarlo para detectar jugadores de fútbol pero esto implica preparar un set de entrenamiento y el tiempo de entrenamiento, por lo cual opté por usar un modelo ya existente.

El primer intento se hizo usando el detector de personas de OpenCV basado en descriptores de HOG, no se obtuvieron buenos resultados con este detector. Se lograban detectar los jugadores que estaban cerca pero los del centro o fondo no los detectaba.

El segundo intento fue usando los detectores de objetos preentrenados de tensorflow, se intentó con los modelos **Faster RCNN Inception v2 COCO**, **Faster RCNN Inception v2 ImageNet, SSD ResNet ImageNet, SSD MobileNet v2 ImageNet** con Faster RCNN Inception v2 ImageNet se logró mejorar los resultados pero aún fallaba detectando a los jugadores después de medio campo, por tanto decidí seguir buscando un modelo más adecuado.

El tercer intento, fue usando un detector de objetos de tensorflow que había sido usado para una tarea similar, por tanto este detectaba a la mayoría de los jugadores. Aunque se tuvo que bajar la tolerancia de aceptación pero debido a que las características de entrenamiento fue en videos de fútbol, las detecciones eran buenas. Por lo tanto, decidí usar este detector de objetos.

Conclusión

Al final los resultados obtenidos fueron buenos aunque se podrían mejorar, aún tengo muchas ideas para este proyecto que debido al tiempo no fueron posibles de implementar. Fue muy interesante trabajar con este proyecto pues siempre se descubren nuevos métodos que puedes agregar para mejorar los resultados, además de lo que puedes hacer con todas las herramientas sólo está limitado a la creatividad. En esta carpeta se adjunta un video con los resultados obtenidos y el código.

Trabajo futuro

Como ya lo mencioné aún tenía más ideas para implementar, en esta sección comentaré algunas de ellas para su futura elaboración.

Calibración. Para la calibración en este caso se hizo de manera manual, un aspecto a mejorar sería hacer esto de manera automática. Podría ser segmentando el campo y detectar las líneas que hay sobre este y así identificar esquinas. Podría hacerse para cualquier tipo de toma sobre el campo, no solo tomas panorámicas.

Proyección de imágenes. Otras ideas para proyectar podría ser el marcador y las insignias de los dos equipos. Faltó implementar el dibujar un círculo de 9.15 metros sobre el balón cuando hay un tiro libre, esta última implementación es casi directa de las herramientas que ya se poseen, pues así como se dibuja una imagen en el centro del campo sería sustituir esta imagen por un círculo y en lugar de usar el círculo central se usarían las coordenadas desde donde se llevará a cabo el tiro libre.

Detección de objetos. Esta es una parte que considero interesante de mejorar, pues creando un set de entrenamiento para detección de jugadores se podría mejorar la calidad de las detecciones. Se podrían mejorar los datos sobre velocidad de los jugadores, por ejemplo incluyendo filtros de Kalman o algo por el estilo. Con esto se podrían crear más estadísticas como una mejor aproximación de la velocidad, la distancia recorrida por cada jugador a lo largo del partido, que secciones cubre cada jugador (como mapa de calor pero por cada jugador), etc.

Realidad aumentada. Dado que se conoce un modelo plano y su correspondencia con la imagen, y la estimación de la distancia, se podrían usar estos datos para incluir elementos de realidad aumentada como publicidad o información sobre el encuentro en 3D.

Clasificación de equipos. Una vez que se detecten los jugadores, otra tarea a realizar sería determinar a qué equipo pertenecen.

Fuera de juego. Trazar las líneas correspondientes con los últimos jugadores de cada equipo para poder decidir si en una jugada había fuera de lugar o no.