

Documentación Primera Tarea Programada

Danny Chaves Chaves, dxnrx@me.com, Minor Sancho Valverde,
tivin.minor10@gmail.com, Oscar Chavarría Campos, ocach14@hotmail.com

August 25, 2016

1 Introducción

El análisis automatizado de videos se ha visto estimulado con la masificación de internet, las plataformas de computación de alto rendimiento a bajo costo, y el diseño de nuevos algoritmos para su procesamiento eficiente. Una aplicación del análisis automatizado de videos es la obtención de datos de alto nivel a partir de videos digitales de futbol, estos datos corresponden al uso de estrategias y tácticas por cada uno de los equipos de futbol. El sistema ACE, actualmente desarrollado en el PRIS-Lab www.pris.eie.ucr.ac.cr, de la Universidad de Costa Rica tiene por objetivo analizar de manera automática videos de futbol. Sus etapas van desde la identificación de escenas con información útil en el video, la segmentación y rastreo de jugadores, hasta el análisis automático de los recorridos y comportamiento de los jugadores. En el presente proyecto, se propone el desarrollo de la etapa de segmentación de jugadores, el cual se detalla a continuación.

2 Análisis del problema

Primeramente para este proyecto debimos investigar sobre la arquitectura de las imagenes, donde se debía saber que cada capa era representada por una matriz y a la vez tenia un significado de color en RGB (Red - Green - Blue), al cambiar las capas a HSV (Hue - Saturation - Value) cambiaba su significado cada capa y la función al momento de representar la imagen, al realizar este cambio se debía saber que ya la matriz no estaba en valores de 0-255, sino de 0-1, en forma de porcentajes. Por otra parte se debía estudiar el lenguaje, para conocer su manejo, paquetes y funciones que poseía para la ayuda en cuanto a manipulación de imagenes.

3 Diseño de la Solución

La solución se realiza mediante un algoritmo sugerido por el profesor, el cual cuenta con los siguientes pasos:

3.1 Algoritmo de segmentación en detalle:

Detección del campo de juego o construcción de la «máscara del campo de juego» y la selección de los posibles jugadores.

1. Detección del campo de juego: Convertir la imagen de entrada U_t a una imagen de cromaticidad H_t tomando la capa de cromaticidad H después de usar la función `rgb2hsv`.
2. Encontrar los valores x_{\min} y x_{\max} que definan un rango de «verdosidad» para umbralizar la imagen en ese rango y obtener una máscara binaria de píxeles verdosos G a partir de P .
3. Rellenar los hoyos en la máscara G , posiblemente correspondiente a los jugadores, usando la función `infill` (en Octave `bwfill`).
4. Eliminar las «regiones espurias» pequeñas, usando la función `bwareaopen`.
5. Por último se debe eliminar el logo del partido.

3.2 Detección de regiones candidatas a jugadores

1. Convertir la imagen de entrada U_t a una imagen de cromaticidad H_t tomando la capa de cromaticidad H después de usar la función `rgb2hsv`. Se aconseja normalizar de 0 a 255 la imagen.
2. Calcular la imagen de varianza local St usando como entrada la imagen H_t .
3. Calcular el umbral óptimo T para la imagen St , usando el algoritmo de Otsu o Kittler de umbralización por máxima verosimilitud, implementando en la función `graythresh` de MATLAB.

4 Pruebas

El video completo está disponible en el siguiente enlace <https://youtu.be/WDebB55ubk4>



Figure 1: Imagen original

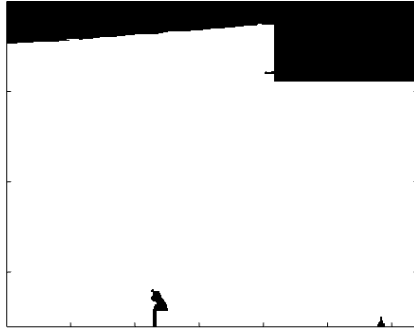


Figure 2: Máscara: detección terreno de juego

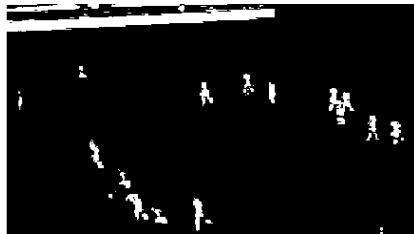


Figure 3: Máscara P



Figure 4: Resultado Final Frame 1

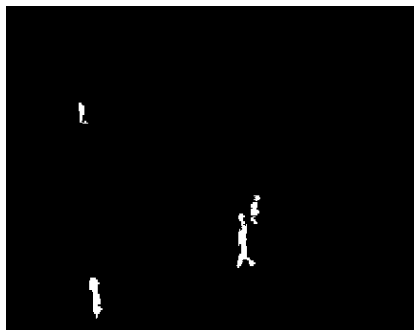


Figure 5: Resultado final Frame 50

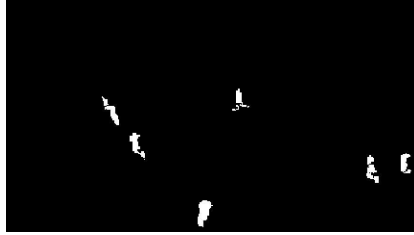


Figure 6: Resultado final Frame 100

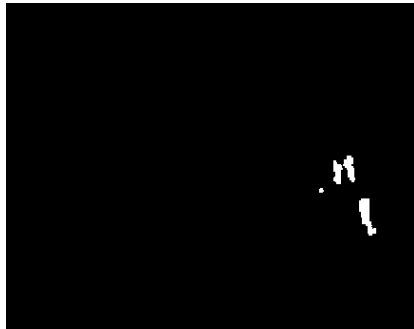


Figure 7: Resultado final Frame 150

5 Conclusiones

Durante la realización de este proyecto logramos apreciar el gran potencial de Octave, el cual brinda muchas herramientas para la manipulación de matrices e imágenes, que otros lenguajes de programación no brindan un soporte con mucho potencial para la realización entre operaciones de de matrices e imagenes. Entre las mayores dificultades que encontramos, vale la pena resaltar lo poco documentado que se encuentra el manejo de imagenes en Octave, se puede encontrar algunos sitios que dan algunos ejemplo, pero no son claros. Otra dificultad es que si se busca algo en Octave, los resultados se encuentran pero en MATLAB. En cuanto a dificultades de hardware, el algoritmo es bastante potente o el poder de procesamiento que necesita es de un alto nivel, por lo que se necesita una computadora con buenas características para poder compilar el programa.