

Práctica de detección de movimiento de ojos

Inicialmente se plantearon 3 diferentes iteraciones:

Segmentar el ojo en tres partes y contar píxeles blancos

Detectar bordes mediante sobel u otro operador

Detectar círculos con Hough

Con la aproximación con la que se obtuvieron mejores resultados fue con la primera de contar los píxeles. Sobel detectaba bastante bien los contornos del ojo pero luego se complicaba la aproximación, y Hough resultó demasiado lento para poder usarlo de forma eficiente en tiempo real.

Así pues se optó por continuar desarrollando la primera iteración, dividiéndola en dos partes (Inicialización y Procesado):

Inicialización:

Asumiendo que el usuario de la aplicación es colaborativo, no va a mover la cabeza más que unos pocos centímetros a los lados.

De modo que en el primer frame capturado se aplicará un detector en cascada para detectar la cara, y recortando con un pequeño margen, despreciaremos de ahora en adelante el resto de la imagen.

Además, una vez recortada la cara, detectaremos en ese mismo frame el ojo, para tener una referencia de donde deberían estar y de qué tamaño aproximadamente serán los ojos a partir de ese momento.

Procesado:

Esta etapa irá cargando los frames que obtiene de la ruta especificada (vídeo o webcam).

Cada frame será recortado en función de la cara detectada en el primer frame y se aplicará un detector en cascada para detectar el ojo en ese trozo de imagen. Cabe destacar la rapidez del detector debido al pequeño tamaño de la submatriz. Una vez detectado el ojo, se reducirá su tamaño en función de unos parámetros calculados de forma empírica y nos quedaremos con una región lo mas ajustada posible al ojo, sobre todo al borde inferior, donde ha resultado que es más preciso el método.

Una vez tenemos el pedazo de ojo deseado, le aplicaremos un umbral, para diferenciar la esclerótica del iris. Este umbral será calculado en función de la media de los niveles de gris del ojo, para evitar problemas de iluminación y, en la medida de lo posible, del color del iris.

Una vez umbralizado, procederemos a segmentar el ojo en tres partes en función de unos parámetros calculados de forma empírica. La primera parte del ojo así pues será desde el

extremo izquierdo hasta un cuarto de la longitud del ojo, y la tercera parte, desde 3/4 de la longitud del ojo hasta el final. Dejando pues, un espacio de la mitad de la longitud del ojo para el iris (cuando está mirando al frente).

Una vez segmentado y umbralizado el ojo, se procederá a contar el número de píxeles blancos en la primera y en la tercera parte del ojo. Considerando un umbral T (La mitad de la suma de las dos partes), una vez más calculado de forma empírica, para el cual la dirección a la que está mirando se calculará en función de lo siguiente:

SI $ABS(NumBlancosPrimeraParte - NumBlancosSegundaParte) < T$

ENTONCES

dirección = frente

SINO

SI $NumBlancosPrimeraParte > NumBlancosSegundaParte$

ENTONCES

dirección = derecha

SINO

dirección = izquierda

Además, se controla de forma exitosa el parpadeo. Esto se realiza llevando un control sobre la media de intensidad del ojo en el instante anterior. Si esa intensidad cambia de forma brusca, superando un porcentaje(para paliar problemas de luminosidad) consideramos que es que ha cerrado el ojo, y por consiguiente, parpadeado. Este umbral se mantiene hasta que finaliza el parpadeo, momento en el cual vuelve a empezar el ciclo de actualización de umbrales. Para prevenir errores, consideramos solo parpadeos de un cierto número de frames consecutivos, para prevenir cambios de iluminación o posibles problemas. Además este umbral se puede ajustar en función de la prioridad que se le quiera dar al parpadeo con respecto al movimiento de los ojos.

Como conclusión, el sistema funciona casi a la perfección en los videos de muestra. En la webcam no funciona tan bien, debido a que es un entorno con más ruido, por lo que el ojo no se recorta de forma tan precisa y le cuesta más seguirlo. Aun así tiene una precisión aceptable.

Daniel Ruiz Pérez