

PRÁCTICA 3: DETECCIÓN AUTOMÁTICA DE OJOS

Antonio José Moya Díaz

24 de junio de 2012

INTRODUCCIÓN.- En el presente documento se expone una aproximación a la solución del problema de la detección automática de ojos en una fotografía.

Existe una diversidad de propuestas y algoritmos para la obtención de los ojos, así como de otras características biométricas tales como la boca o la cara.

En este experimento se ha optado por una detección inspirada en un modelo probabilístico basado en detección por color. La idea es que los ojos serán la zona más oscura de la cara, por lo que el proceso puede ser dividido fácilmente en 2 pasos: la delimitación de la cara y la búsqueda de 2 zonas oscuras en ese área delimitada.

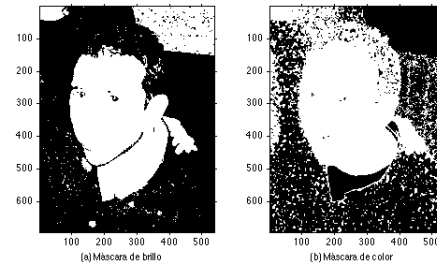


Figura 1: Máscaras para detección de piel

1. Detección de la piel

Para detectar, como anticipamos anteriormente, el área de la cara, vamos a intentar detectar la piel.

Vamos a obtener una máscara para la zona de la piel mediante la combinación de dos máscaras distintas.

Para obtener la primera de las máscaras trabajaremos en el espacio HSV. Tomaremos la banda del brillo, V, y la umbralizaremos. Podemos tomar un valor variable para dicha umbralización pero se propone un valor de 90 sobre 255. Con esta umbralización obtenemos una máscara en la que separaremos todas las áreas oscuras de la foto. En una foto aproximadamente bien iluminada, la piel será una zona más bien brillante, así que esta máscara nos servirá para descartar de una primera y sencilla forma la mayor parte de la imagen, referentes al fondo u otros elementos oscuros de la misma.

Para la segunda máscara trabajamos en RGB

usando unos umbrales propuestos en [1], que se definen como siguen:

$$90 < norm_r < 145 \quad (1)$$

$$60 < norm_g < 100 \quad (2)$$

dónde:

$$norm_r = \frac{r}{r + g + b} 255 \quad (3)$$

$$norm_g = \frac{g}{r + g + b} 255 \quad (4)$$

Podemos observar ambas máscaras en la figura 1. Así pues podemos obtener una primera aproximación a la máscara de piel aplicando la operación AND entre ambas máscaras.

Como podemos observar en la figura 2 (a) han quedado zonas oscuras dentro de la zona de la cara que nos interesa eliminar para tener una máscara de la cara sin agujeros. Para eliminarlos hemos usado herramientas de morfología matemática. Además, resulta que la máscara de la piel

así definida puede estar creada por exceso. Esto es, en los bordes puede que zonas oscuras, tales como el pelo, hayan caído dentro de nuestra máscara. Zonas que pueden alterar la posterior detección de los ojos. Para evitar ésto, volvemos a usar morfología para erosionar nuestra máscara. Así obtendremos una máscara que detectará la piel por defecto, y que podemos observar en la figura 2 (b).

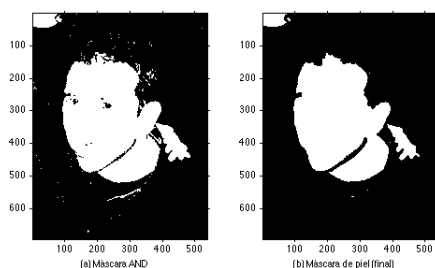


Figura 2: Máscaras para detección de piel

2. Detección de los ojos

Aplicando la máscara anterior a la imagen original tenemos el área de piel separada del fondo (3). Para obtener el área de los ojos umbralizamos ahora sin miedo, además a un nivel bastante bajo, entre el 20 % y el 30 %, sabiendo que los ojos serán ahora la zona más oscura de toda la imagen.

Tras la primera umbralización (figura 4 (a)) la máscara nos puede quedar sucia, es decir, con diversos puntos que han pasado el umbral pero que son indeseables. Además, los ojos suelen presentar un punto pequeño brillante en el centro del ojo, derivado del flash, que no pasará por nuestro umbral. Esto generará un agujero en el ojo que hay que arreglar ya que, para eliminar los puntos no deseados, vamos a aplicar un proceso de apertura, y el agujero en el ojo podría provocar que el ojo se perdiera tras el proceso de apertura. Por ello, se propone primero realizar un rellenado de áreas para que los ojos queden de forma uniforme (figura 4 (b)).

Una vez enmascarados los ojos, solo nos queda

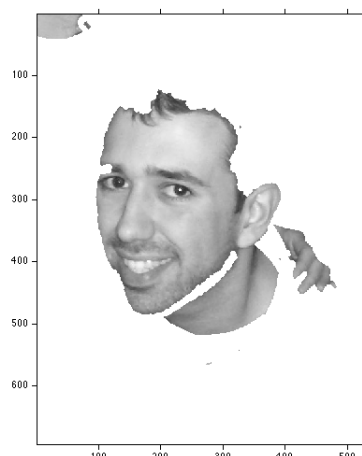


Figura 3: Área de la piel

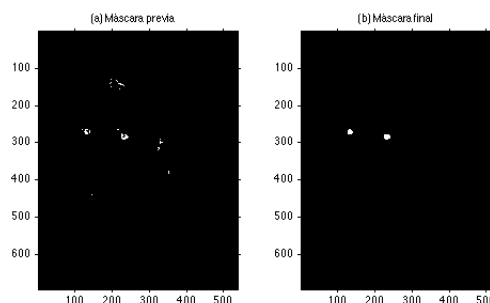


Figura 4: Máscaras de los ojos

extraer sus coordenadas. Para ello hemos usado las funciones de MATLAB *bwlabel* y *regionprops*. Además, hemos aprovechado una característica de la primera para intentar automatizar aún más el proceso.

Con los rangos y el procedimiento seguido se obtiene, para un cierto rango de fotos, un máximo de 2 regiones en el rango de umbralización anteriormente comentado. Esto nos permite iterar desde el límite inferior umbralización hasta el límite superior esperando encontrar 2, y solo 2, regiones que supondremos corresponderán con nuestros ojos. Una vez seleccionada la máscara con solo 2 regiones

pertenecientes a los ojos, la función *regionprops* nos permitirá calcular fácilmente los centroides de esas áreas, dándonos así unas coordenadas para los ojos buscados, figura 5.

algunas condiciones como por ejemplo, de separación mínima, altura con respecto a la cara, o ángulo formado con la horizontal por una línea imaginaria que cruce por ambos ojos.

3. Limitaciones y problemas del algoritmo

La principal limitación del algoritmo es que solo es válido para imágenes en color. Como se dijo anteriormente, la piel se detecta usando características del color, por lo que no funcionaría con imágenes en escala de grises. Además la imagen debe estar en color real. Imágenes en falso color son imposibles de detectar.

Las condiciones de iluminación son también importantes. Si el fondo es más claro que la piel ésta se detectará erróneamente haciendo, por tanto, imposible localizar los ojos con el método propuesto. Este efecto se hace más patente en el caso de fotografías de personas de raza negra.

Las operaciones que se realizan con morfología matemática también hacen los ojos fuertemente susceptibles de elementos estructurales en la fotografía. Pelos o bordes de gafas que estén pegando al ojo falsearán los resultados o, directamente, harán imposible la detección.

También resulta imposible determinar la posición de los ojos si hay más de una persona en la imagen. En el mejor se detectarían los ojos de solo una de las dos personas.

Posibles mejoras

Tras el estudio realizado durante este trabajo llego a la conclusión de que no es la detección basada en color la más indicada para la detección de ojos. No obstante, el presente algoritmo se podría mejorar por algunos puntos.

El quizá más significativo es la limitación de posibles resultados a sólo 2 áreas. Este decisión fue tomada, entre otras, debido al escaso tiempo del que he dispuesto para realizar el trabajo. Esta limitación podría haber sido solventada tomando algunas medidas geométricas. Una vez extraídos un conjunto de posibles candidatos se podrían haber ido comparando 2 a 2 en busca de que cumpliesen

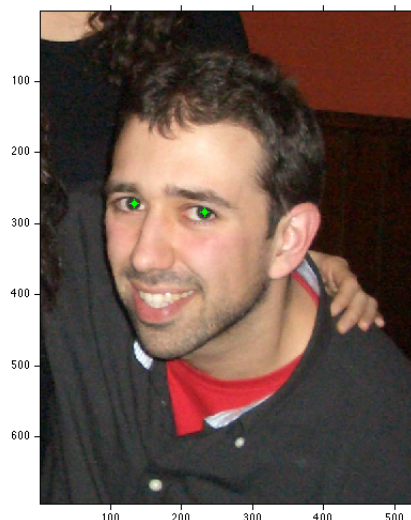


Figura 5: Ojos localizados

Referencias

- [1] R. Thilak Kumar, S. Kumar Raja and G. Ramakrishana *Eye detection using color cues and projection functions*. Department of Electrical Engineering. Indian Institute of Science.