

Informe del Proyecto de Programación: Parte I

Toni Forcada Sans

Gerard Casas Béjar

Abstract— En este documento se especifica el trabajo realizado en la primera parte de un proyecto de procesamiento de imagen que formado por dos sistemas y en ambos se realizan tres ejercicios: diseño, implementación y evaluación.

I. PRESENTACIÓN DEL PROBLEMA

Partiendo de la idea de que el reconocimiento de gestos tiene varias aplicaciones en la vida moderna, ya sea en el control de un ordenador, el uso de un teléfono móvil o en el mundo de los videojuegos, hemos decidido realizar una aplicación que permita reconocer gestos hechos con la mano. En nuestro caso, el problema se resume en generar un programa que permita identificar, a partir de imágenes de manos, el número de dedos que aparecen en ellas. Razonamiento y justificación de las elecciones hechas

Como la crominancia asociada a la piel es una característica bastante invariante en el ser humano, hemos podido aprovechar esto para poder detectar los píxeles de piel de las diferentes imágenes. Para ello hemos transformado la imagen del espacio de color RGB a YCbCr.

En el Alg02 hemos cogido como elemento

II. FORMULACIÓN DE LA IMPLEMENTACIÓN

Para la implementación hemos separado el problema en dos partes:

- 1) Definir, programar y evaluar un sistema que permita la identificación en imágenes de zonas de piel, representando potencialmente manos, dedos, brazos etc.
- 2) Utilizar el sistema de la primer parte para definir un nuevo sistema que permita contar el número de dedos que hay en una imagen y así reconocer el mensaje del usuario.

En este artículo nos centraremos en la 1ª parte.

Para evaluar los sistemas objetivamente, se nos han facilitado dos bases de datos Dataset I (fondos bastante uniformes) y Dataset II (fondos complejos) con imágenes de manos y sus correspondientes máscaras binarias.

En esta primera parte hemos creado el sistema de identificación de manos, dedos y brazos. Para ello hemos tenido

que desarrollar cuatro algoritmos definidos de la siguiente forma:

1) Alg01:

Se basa en un script que analice las imágenes de un conjunto de entrenamiento y genere un histograma modelo de los píxeles de piel. Para ello recorreremos cada imagen de train con su respectiva máscara ideal, i realizamos a su vez el histograma en aquellos puntos donde la máscara ideal nos indique que hay piel.

2) Alg02:

Este algoritmo permite analizar una imagen arbitraria y generar una máscara binaria según si los píxeles pertenecen a la piel o no. Para ellos definimos:

- Una matriz cuadrada de dimensiones $N \times N$ con la que recorreremos la imagen en cuestión.
- El histograma resultado del alg01 que nos da un histograma que define la piel.
- Un histograma temporal de cada una de las posiciones de la matriz anteriormente definida.
- La distancia entre el histograma ideal i el temporal que nos dará un valor de similitud entre histogramas.
- Un umbral con el que procederemos a la decisión de asignar a la máscara un 1(no piel) i 0(piel) respecto al valor obtenido en el paso anterior.

Con este valor binario se creará una matriz binaria que deberá ser la

3) Alg03:

Se trata de un script que utiliza el algoritmo alg02 para generar las máscaras de piel de todas las imágenes de un conjunto y las guarda en un directorio Masks. Además creará una matriz de 3 dimensiones (m,n,k), donde k será el Identificador de cada imagen. Esta matriz nos será útil para evaluar la implementación

4) Alg04:

Este algoritmo permite calcular el valor de F-Score comparando las máscaras generadas por el sistema con las máscaras ideales que se encuentran en las bases de datos. Se cogerá como argumento la matriz de máscaras calculadas, y por cada una de las máscaras de la matriz la comparará con su respectivo ideal obtenida del directorio "masks-ideal".

III. RESULTADOS

Precedemos a realizar la implementación con dos métodos para comparar histogramas:

- a) Como: $abs(sum(hist_temp - hist_pell)) * 10^{17}$
(multiplicamos por un factor de 10^{17} para tener valores mayores que cero)
- b) Como: $abs(sum(hist_temp - hist_pell)) * 60$

En el primer caso obtenemos unas máscaras como las siguientes.



Podemos ver como la máscara calculada es buena aunque comparándola con la ideal hay muchos errores. Para otras imágenes la máscara daba un mejor resultado, pero para muchas hay muchos falsos positivos, es decir que el algoritmo considera piel zonas de la imagen que no lo son.

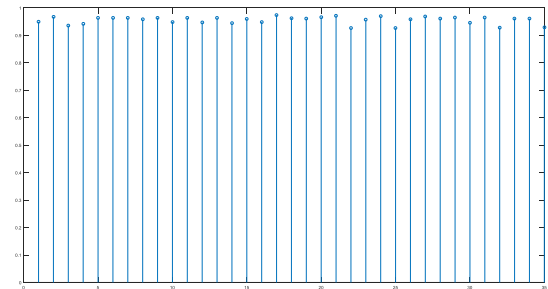
Es por eso que planteamos la opción b). Con esta obtenemos.



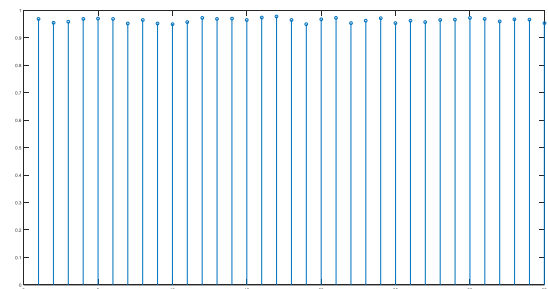
Podemos observar cómo, aunque las máscaras no son tan regulares, los falsos negativos bajan considerablemente.

Visto esto, podemos también valorar el F-score de las imágenes Test. Este F-Score se calculará dos veces, tanto para el entrenamiento con Dataset-1 i Dataset-2.

F-score1:



F-score2



Como valor medio obtenemos:

Fscore1=0.9555

Fscore2=0.9638

IV. CONCLUSIONES

Concluimos pues con hemos obtenido unas máscaras lo suficientemente buenas como para proceder a la contabilización de dedos, minimizando los falsos negativos. Aun así el tiempo de procesado de todo el conjunto de algoritmos de alto por lo que deberíamos plantear otro método. Pues acabamos analizando pixel a pixel muchas imágenes, cosa que aumenta el tiempo computacional.