

# Informe del Proyecto de Programación: Parte II

Toni Forcada Sans

Gerard Casas Béjar

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

## I. PRESENTACIÓN DEL PROBLEMA

Al igual que en la primera parte, el objetivo que se desea lograr con este proyecto es implementar un sistema capaz de detectar y contar, de forma correcta, el número de dedos de una mano que aparecen en una imagen.

Partiendo del contenido de la primera parte del proyecto, donde obteníamos unas máscaras de piel más o menos correctas (en algunos casos mejores y en otros peores), el próximo paso era diseñar el sistema de detección de dedos a partir de dichas máscaras binarias. Teniendo en cuenta la calidad de estas, seremos más o menos capaces de contar los dedos de cada imagen y por tanto, generar unos resultados mejores.

## II. FORMULACIÓN DE LA IMPLEMENTACIÓN

Al final de la primera parte del proyecto, obtuvimos unas máscaras de piel que se asemejaban de forma coherente con las máscaras ideales. Pero para poder trabajar de forma más cómoda con unos datos mejores, nos dispusimos a editar el algoritmo 3 para que las máscaras obtenidas fueran aun mejores. Para ello utilizamos la función de MATLAB *bwareaopen(BW,P)* que nos permite eliminar objetos de menos de P píxeles de una imagen binaria, BW. De esta forma, de la base de datos DataSet I obtuvimos máscaras binarias como esta:



Fig 1: Máscara de 3 dedos de Dataset I después de aplicar *bwareaopen(BW,P)*.

Una vez teníamos las máscaras con una calidad significativa, nos dispusimos a implementar el sistema de detección de dedos.

Para este sistema decidimos utilizar una implementación del sistema *Middle Axis* utilizado por Georgiana Simions, Vasile Gui y Marius Otesteanu. Su sistema utilizaba el histograma de color y el contorno de la mano para detectar los dedos.

En nuestro caso, hemos utilizado un algoritmo que realiza un barrido horizontal y vertical en toda la máscara, calculando el punto medio entre los puntos del contorno para determinar que parte pertenece a la mano y que pertenece a los dedos. Para ello establecemos un umbral de decisión. Al sobrepasar este umbral, entiende que se trata de la mano. Por el contrario, si el punto medio está por debajo del umbral, entiende pues, que se trata de un dedo. En la siguiente imagen podemos ver el resultado obtenido a través del algoritmo *middle\_axis* utilizando una imagen con el fondo complejo y podemos ver que el algoritmo detecta de forma correcta la posición del dedo en esta imagen.

Para eliminar aquellos puntos que queden entre los dedos (en el caso de más de 1 dedo), deberemos superponer el resultado a la máscara para así eliminar dichos puntos.

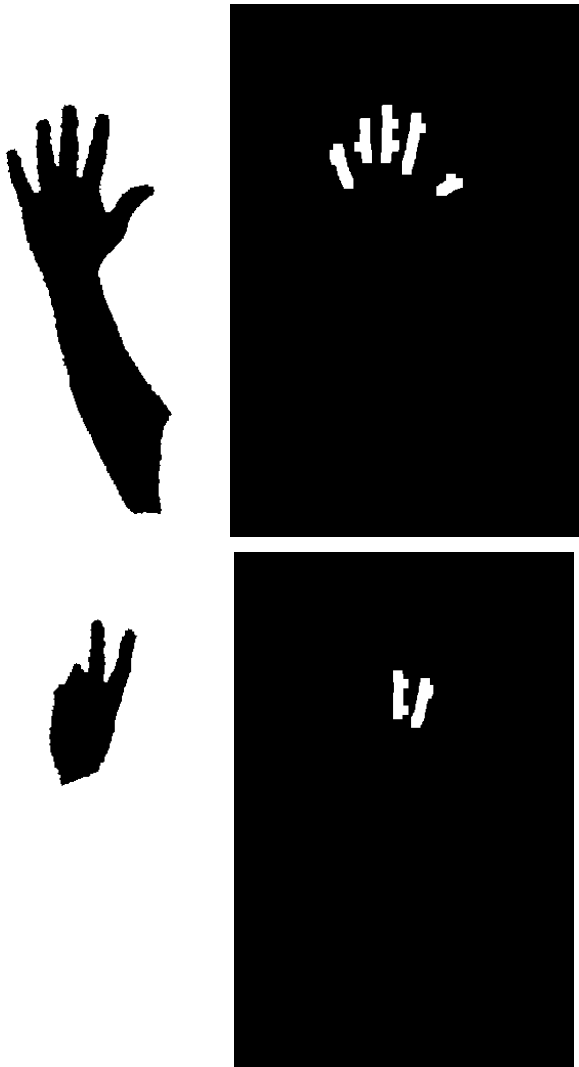
Para poder realizar el recuento deberemos dilatar para tener regiones más consistentes (como salida del “*middle\_axis*”).

Para obtener de forma adecuada las regiones pertenecientes a dedos, hemos utilizado posteriormente el mismo método que para tratar las máscaras, *bwareaopen* (elimina pequeñas regiones que no pertenecen a los dedos). De esta forma hemos obtenido unas regiones de dedos mucho más claras.

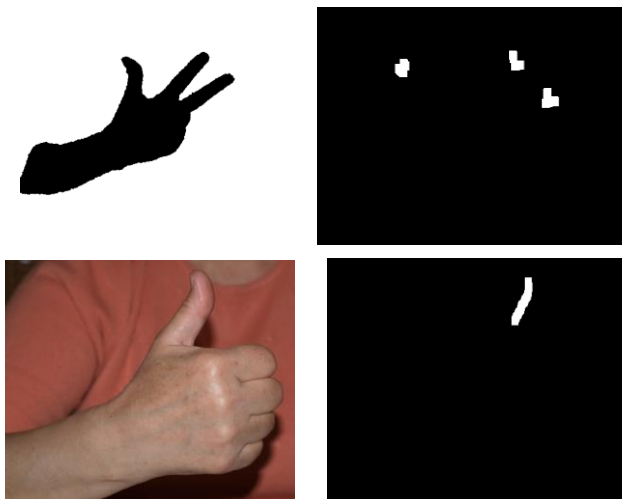
Por último, para el recuento del número de dedos de cada máscara hemos utilizado la función de MATLAB *bwlabel(BW)* que para la imagen BW nos devuelve la matriz de las regiones etiquetadas de los objetos de la imagen con conectividad 8. Para utilizar correctamente esta función, tan solo cogemos el segundo resultado de esta función, que es el número de regiones reconocidas.

Pondremos algunos ejemplos en cada uno de los Dataset:

En Dataset-I:



Dataset-II:



### III. RESULTADOS

Para comprobar el cálculo y la implementación correcta de nuestro sistema, al igual que en la primera parte del proyecto, hemos utilizado la medida f-score para evaluar la calidad del sistema. Para esta medida, aprovechamos el formato del nombre de las imágenes donde en primer lugar aparecía el número de dedos de la imagen. Aprovechando esto, calculamos el f-score, pero en este caso, el resultado para cada imagen solo podrá ser 0 o 1 dependiendo de si el recuento de dedos es correcto o no. Con las diferentes bases de datos hemos obtenido unos resultados más o menos adecuados:

- **Con Dataset I:** F-Score = 0.68
- **Con Dataset II:** F-Score = 0.3

Para las imágenes test obtenemos los siguientes resultados:

- **Para DS-II:**  
Score = 0.771429
- **Para DS-II:**  
Score = 0.116667

### IV. CONCLUSIONES

Por último, nos gustaría comentar los resultados obtenidos. Por empezar la implementación del Sistema ha sido mucho más eficiente que la de la parte 1 del Proyecto, donde el coste computacional fue mucho más elevado. En este caso, el tiempo de procesado es mucho más bajo. Por otra parte, los resultados obtenidos no terminan de ser buenos cuando los fondos de las imágenes son irregulares dando muchos más errores que resultados correctos. Este se debe a que las máscaras que se obtienen no son satisfactorias.

Otro motivo del mal resultado para Dataset II es el hecho que las manos se encuentran a diferentes distancias de la cámara, por lo que las regiones de los dedos son más grandes. Por lo tanto, los umbrales deberían cambiar. Una solución sería tener en cuenta el tamaño de la región para definir los umbrales, así para manos más acercadas a la cámara el algoritmo será más permisivo.

Por otro lado, en las imágenes con fondos uniformes, los resultados son bastante mas satisfactorios, dando un resultado correcto casi en el 70 % de las imágenes testeadas.

### V. REFERENCIAS

- [1] G. Simion, V. Gui, and M. Ottesteanu, "Finger Detection Based on Hand Contour and Colour Information". May 2011

