

Informe: Reconocimiento de Gestos

David Afonso Dorta

Javier Gómez de Vera

Christian Jesús Pérez Hernández

- Descripción del método de sustracción de fondo que hemos visto en prácticas, así como las ventajas e inconvenientes que han encontrado.

Para realizar la sustracción de fondo se siguen las siguientes fases:

1. Se inicializa la cámara y se dibujan en la pantalla varios rectángulos. Estos indican las regiones de interés que el usuario tiene que cubrir con la mano para extraer los colores.
2. Se convierte el espacio de color de RGB a HSL. Utilizamos este espacio de color porque así la componente Luz queda aislada, utilizando solo el “color real” de la mano. Esto debería permitir que el programa siga detectando bien la mano incluso cuando hay ligeros cambios en la iluminación.
3. A partir de la imagen HSL, obtenemos muestras de las regiones de interés de la mano (los rectángulos verdes), y calculamos la media de cada una de ellas. Esto es necesario porque algunas zonas de la mano pueden tener un color ligeramente distinto.
4. El último paso es obtener una máscara binaria por cada una de las medias calculadas. Para ello, se utiliza la función `cv::inRange`. Dicha función devolverá 1 si el color está dentro del rango `low - high` que se le especifique, o 0 en otro caso. Para calcular los límites `low` y `high`, le hemos restado (o sumado) a cada componente HSL un valor (`h_low`, `h_high`, `l_low`, `l_high`...) modificable a partir de las trackbars de la ventana. Después de hacer el cálculo, hay que limitar los valores para que `low` no sea menor que 0, y `high` no sea mayor que 255. Finalmente, estas máscaras binarias se sumarán, creando una única máscara binaria como resultado.

Este método tiene una desventaja principal, y es que si es en las regiones de interés se coge parte del color del fondo, la máscara generada tendrá mucho ruido (ya que los colores del fondo también se marcarán como blanco). Es por eso que la mano siempre debe cubrir completamente todos los rectángulos de la imagen. Además, este sistema de sustracción de fondo no sirve cuando mano y fondo tienen el mismo color, ya que no hay forma de distinguirlos, al igual que detectará la cara y otras zonas del cuerpo que estén al descubierto como parte de la máscara.

Sin embargo, el método tiene sus ventajas, como que funciona muy bien cuando el color del fondo es muy diferente al de la mano, es sencillo, y permite que la cámara o el fondo puedan moverse sin que eso afecte a la detección.

- ¿Cómo funciona el filtro de la mediana y por qué es interesante para reducir el ruido? ¿Cómo funcionan las operaciones morfológicas de dilatación y erosión y cómo se pueden combinar para reducir el ruido?

Filtro de la mediana:

Es uno de los métodos utilizados a la hora de filtrar el ruido en la máscara generada. Básicamente, por cada píxel comprueba los colores que hay en su vecindad, y reemplaza el valor del píxel por el valor de la mediana. Por ejemplo, si en la máscara binaria hay algún píxel blanco en una zona con mucho negro, este se pintará de negro, y viceversa. Además, a diferencia de otros filtros como la media, este no hace que la imagen original se vea más borrosa, y tampoco añade colores nuevos a la imagen, por lo que es ideal para la práctica.

Sin embargo, hay que tener en cuenta que es una operación computacionalmente cara. Por eso, se puede ajustar el tamaño de la vecindad mediante un parámetro de la función.

Operaciones morfológicas:

Las operaciones morfológicas son operaciones de transformación que se aplican a cada píxel de una imagen binaria para modificar la forma. Para realizar las operaciones morfológicas también es necesario que haya un elemento estructurante

(o kernel), que defina la forma que tiene la vecindad (rectángulo, elipse...). En la práctica hemos visto las dos más básicas, dilatación y erosión:

- **Dilatación:** Si hay al menos un píxel del kernel que esté marcado como 1 (blanco), el resto se pondrá también a 1, “dilatando” las zonas blancas de la imagen.
- **Erosión:** Si hay al menos un píxel del kernel que esté marcado como 0 (negro), el resto se pondrá también a 0, “erosionando” las zonas blancas de la imagen (o “dilatando” las zonas negras, por eso se dice que dilatación y erosión son operaciones complementarias).

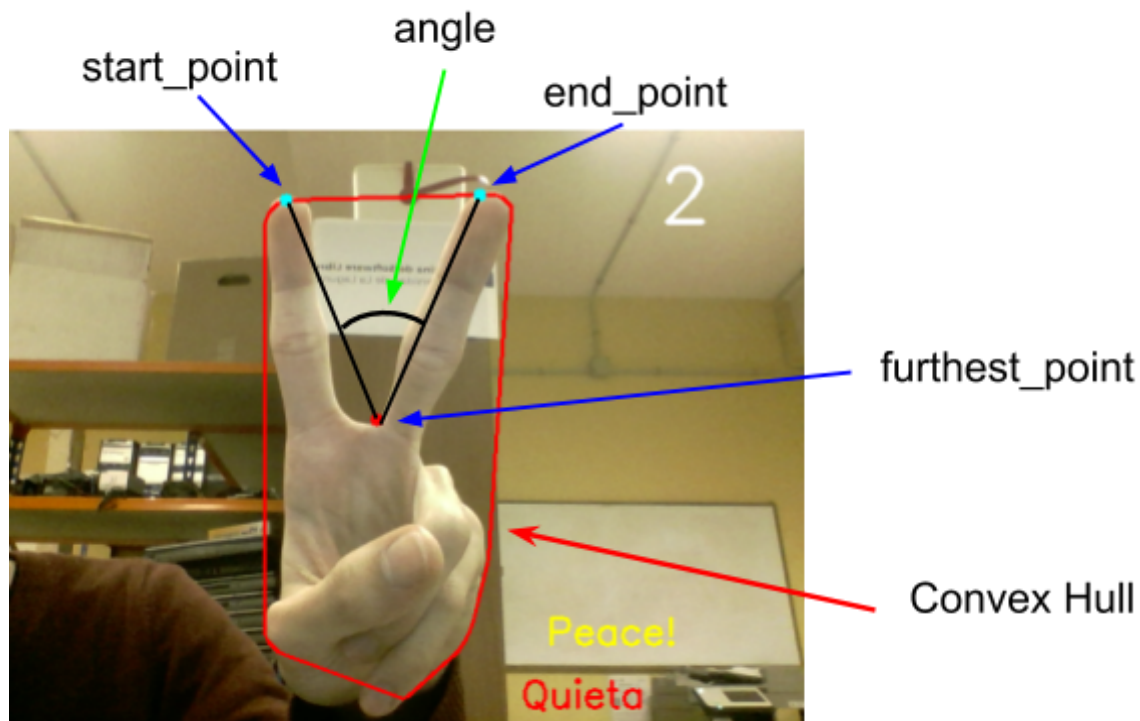
La idea para reducir el ruido es combinar las operaciones morfológicas. Por ejemplo, se puede utilizar primero la erosión para eliminar las islas blancas que puedan haber en el fondo de la máscara, y luego utilizar la dilatación para eliminar los huecos negros que pueda tener la máscara de la mano.



- Explica qué es una malla convexa (convex hull) y qué son los defectos de convexidad (convexity defects). ¿Qué criterio has usado para filtrar los defectos de convexidad que ocurren en la unión de los dedos? ¿Qué aplicación final basada en reconocimiento de gestos has implementado?

Una malla convexa es una especie de “malla elástica”, generada a partir de los extremos del contorno de la mano. Tiene la característica de que vista desde dentro, ninguno de los ángulos que la forman tiene más de 180 grados. Es a partir de esta malla convexa de donde se sacan los defectos de convexidad. Básicamente, un defecto de convexidad marca las zonas de la malla convexa donde el contorno “se aleja” del borde.

Cada uno de los defectos de convexidad está formado por cuatro componentes. Las tres primeras (`start_index`, `end_index` y `furthest_point`) son índices del contorno de la mano, correspondientes a los extremos y al punto más alejado del borde. La cuarta componente es la distancia entre el borde y el `furthest_point`, que se puede obtener dividiendo la componente entre 256. Además, a partir de estos tres puntos se puede obtener el ángulo que forman.



Con esta información (distancia al borde y ángulo) podemos filtrar los defectos de la mano, con el fin de solo quedarnos con los que se encuentren en la unión de los dedos. Los dos criterios utilizados para filtrar los defectos son:

1. La distancia al borde debe ser menor que el 35% del `bounding rect`. (El "bounding rect" es un rectángulo que engloba toda la malla convexa, y que se utiliza como referencia para las operaciones, haciendo que estas sean invariantes a la escala).
2. El ángulo que forma el defecto no puede ser ni muy pequeño ($< 15^\circ$) ni muy grande ($> 120^\circ$).

Finalmente, se han implementado las siguientes funcionalidades en la aplicación:

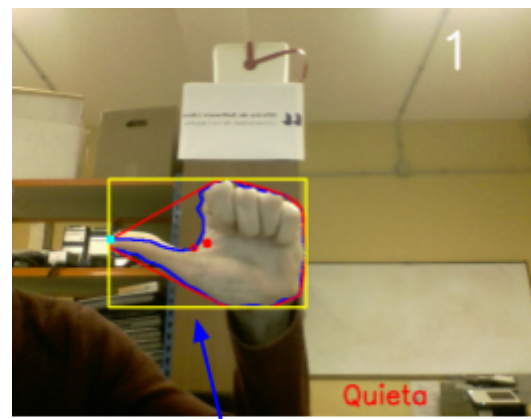
Contador de dedos:

Para contar los dedos, se cuenta el número de defectos de convexidad (es decir, de uniones entre los dedos) y se le suma uno. Sin embargo, hay que contemplar los siguientes casos:

- **La mano está cerrada:** Cuando ocurre no hay que sumar uno extra. Para detectarlo, se comprueba que no haya ningún defecto de convexidad y que la diferencia entre el alto y el ancho del rectángulo sea menor a un umbral (pareciéndose más a un cuadrado).
- **Se levanta el dedo índice:** En este caso, aunque se detecte un dedo, no se forma ningún defecto de convexidad. Sin embargo, se diferencia del caso de la mano cerrada en la proporción del rectángulo.
- **Se abre el dedo pulgar (hacia la izquierda o la derecha):** En este último caso, el pulgar crea un defecto de convexidad, pero no hay que sumarle uno extra. Este caso se detecta cuando solo hay un defecto de convexidad y además el rectángulo es más ancho que alto.





Las proporciones del rectángulo se parecen más a un cuadrado





Es más ancho que alto

Detección de gestos estáticos:

La aplicación puede detectar varios gestos sencillos. Para ello, comprueba una serie de condiciones en la mano, basándose en la cantidad de dedos que cuente y en los defectos de convexidad. Las condiciones están basadas en una mano derecha, con los defectos de convexidad generados en el sentido de las agujas del reloj. Los gestos detectados son:

Gesto	Condiciones
Loser	<ul style="list-style-type: none">- 2 dedos- El ángulo formado por el primer defecto de convexidad (pulgar - índice) tiene que estar entre 80° y 120° 
Peace	<ul style="list-style-type: none">- 2 dedos- El ángulo formado por el primer defecto de convexidad (índice - anular) tiene que estar entre 20° y 60° 

<p>Rock</p>	<ul style="list-style-type: none"> - 3 dedos - La distancia entre el comienzo (dedo índice) y el final (dedo meñique) del segundo defecto de convexidad debe ser mayor que el 56% del bounding rect. 
<p>OK</p>	<ul style="list-style-type: none"> - 4 dedos - El ángulo formado por el primer defecto de convexidad (índice - anular) debe ser mayor que 50° - La distancia vertical entre el comienzo y el final del primer defecto de convexidad debe ser mayor que el 25% del bounding rect. - La distancia entre los furthest_point del primer y segundo defecto de convexidad debe ser menor que el 20% del bounding rect. 

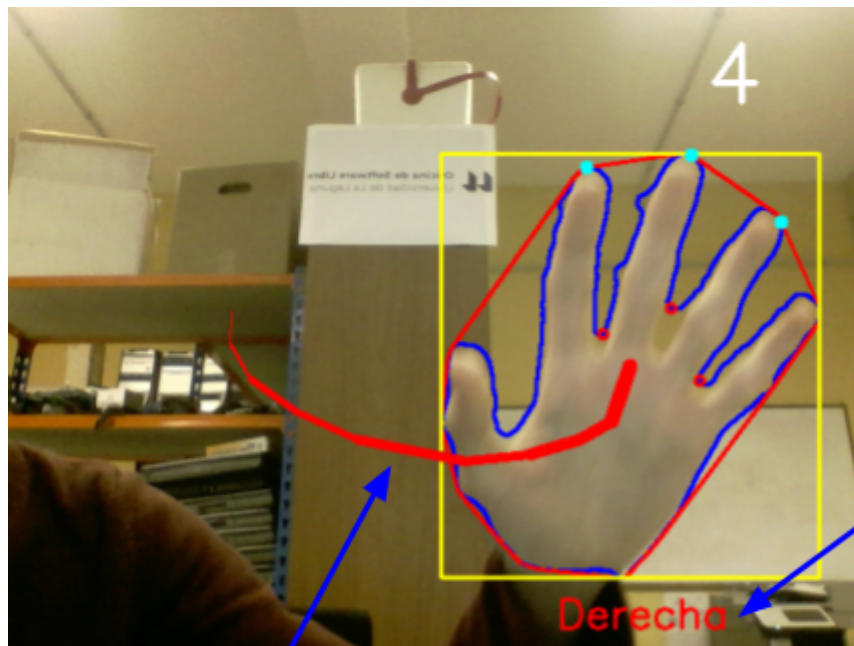
Hay una tecla para hacer que los defectos de convexidad se creen en el sentido contrario a las agujas del reloj, pudiendo así detectar los gestos utilizando la mano izquierda.

Detección de la dirección de movimiento de la mano:

La aplicación puede detectar la dirección hacia la que se esté desplazando la mano respecto a la pantalla (arriba, abajo, izquierda, derecha y diagonales).

Para ello, por cada frame la aplicación inserta el centro del bounding rect (que es, aproximadamente, el centro de la mano) en una cola de 10 elementos. De esta forma, siempre tendremos las últimas 10 posiciones de la mano.

En base a esta cola de posiciones, se calcula la diferencia entre la primera y la última posición de la mano, en la componente x y la componente y. Si esta diferencia es mayor a un cierto umbral, podemos decir que la mano se ha desplazado en ese eje, y mediante el signo podemos distinguir la dirección hacia la que se ha desplazado.

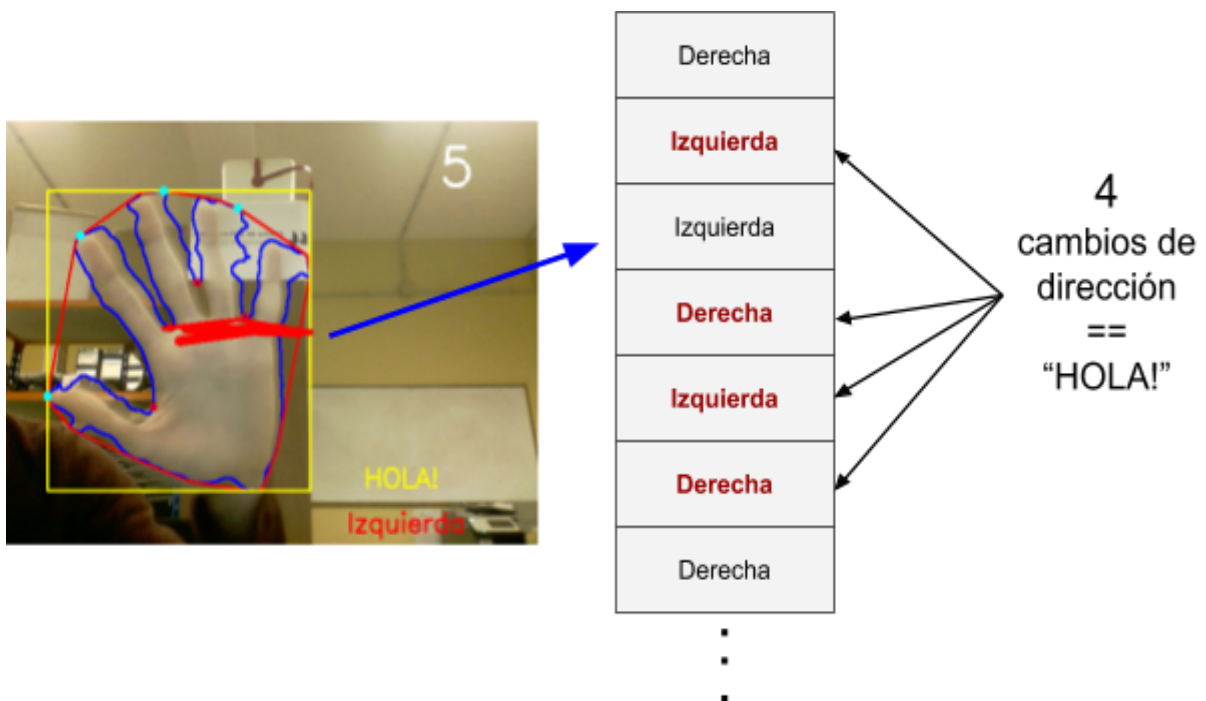


La diferencia entre el principio y el final de la estela marca la dirección de movimiento

La estela representa las últimas 10 posiciones de la imagen. Cuanto más pequeño, más antiguo es el punto

Detección de gestos dinámicos:

La aplicación también puede detectar un sencillo gesto basado en el movimiento: abanicar la mano de derecha a izquierda rápidamente, simulando el gesto de “hola”. Se basa también en el concepto de una cola, pero guarda las últimas 10 direcciones de movimiento de la mano ([Izquierda, Izquierda, Derecha, Derecha, Arriba...]). Para detectar el gesto, simplemente va a contar la cantidad de veces que la mano cambia de dirección (de Derecha a Izquierda y viceversa). Si ve que en los últimos 10 frames la mano ha cambiado de dirección más de 4 veces, entonces mostrará el mensaje de “Hola”. Además, no detectará el gesto si en algún momento la mano se ha desplazado hacia arriba o hacia abajo.



Dibujar:

La aplicación también permite utilizar las puntas de los dedos para dibujar líneas en la pantalla. Cuando detecte 2 dedos, irá guardando en un vector (`current_line`) la posición del `start_index` del defecto del convexidad. Si detecta más o menos dedos, se cortará la línea actual y se guardará en un vector (`drawn_lines`) dicha línea junto al color.

Finalmente, por cada frame se dibujan tanto la línea actual (`current_line`) como todas las líneas anteriores (`drawn_lines`), utilizando la función `cv::polylines`. En la pantalla también se muestra una interfaz sencilla con la que elegir los colores, simplemente poniendo un dedo encima del rectángulo. Si se seleccionan varios colores a la vez (con varios dedos), estos se combinarán para formar un color nuevo.

