

Capítulo 7

CONCLUSIONES Y LÍNEAS FUTURAS

Este proyecto ha presentado, como versa el título, la implementación en MATLAB de un algoritmo de segmentación de imágenes a color basándose en las propiedades de discontinuidad, esto es, detectando los contornos de la misma. Como ya se ha dicho, la segmentación de imágenes es actualmente una de las fases más importantes del procesamiento digital de imágenes. En definitiva, se trata de "aislar" la parte de la imagen que interese (objeto) para poder realizar su posterior estudio.

Una vez conocidas todas las técnicas y formas de segmentación existentes, se ha seleccionado el método Live Wire como el más adecuado, ya que al tratarse de una forma semi-automática posee las ventajas de la forma automática (en la que el ordenador se encarga de todo el procesamiento computacional), y las de la forma manual (en la que el usuario puede ir guiando al ordenador en todo momento para conseguir los resultados deseados).

A modo de **resumen** se listarán los capítulos que forman el presente documento, explicando en breves palabras la línea desarrollada en cada uno de ellos:

- **Capítulo 1.** Introducción y explicación del objetivo del proyecto.
- **Capítulo 2.** Introducción a la segmentación de imágenes, detallando los diferentes métodos, técnicas y formas existentes, y haciendo alusión, por primera vez, de las que se han considerado más idóneas para aplicar en este trabajo.
- **Capítulo 3.** Explicación teórica del método de segmentación seleccionado: método Live Wire. Formación del mapa de coste local de la imagen y transformación de éste en el mapa de coste acumulado. Presentación de los problemas surgidos y explicación de las modificaciones realizadas al método para solucionar dichas dificultades.
- **Capítulo 4.** Explicación del funcionamiento del programa diseñado en MATLAB para segmentar imágenes. Diagrama de flujo del programa completo y descripción de las funciones que componen el mismo (códigos MATLAB en Apéndice A).

- **Capítulo 5.** Análisis experimental encaminado a una óptima segmentación de imágenes de quemaduras, aplicación principal para la que se ha desarrollado este trabajo (imágenes originales en Apéndice B).
- **Capítulo 6.** Presentación de los resultados obtenidos al segmentar un grupo de imágenes de quemaduras y otro grupo con varias imágenes a color.

Los resultados obtenidos son buenos en calidad, pero mejorables en tiempo. Para imágenes de quemaduras de un tamaño medio de **300 x 220 píxeles** se obtuvo un tiempo medio de segmentación de aproximadamente **1 minuto**, utilizando alrededor de 10 puntos; mientras que para el resto de las imágenes a color que tenían un tamaño mayor (**550 x 425 píxeles**) el tiempo era de unos **3 minutos**.

De esta forma, se observa como aún habiendo realizado modificaciones al método, con la creación de subimágenes a las que se les calcula los mapas de coste local y acumulado en vez de hacerlo a la imagen completa, la mayor limitación sigue siendo el tiempo empleado en la segmentación. Esto se debe, principalmente, a que el algoritmo se ha implementado en MATLAB, que no utiliza un lenguaje propio para el desarrollo de aplicaciones. Como ya se dijo, este proyecto no tiene como objetivo el diseño de una aplicación real de cara al usuario, sino el desarrollo y estudio de un algoritmo de segmentación de imágenes que optimice la dupla calidad-tiempo. Todo ello se acompaña de una profunda investigación sobre las mejores opciones que se adaptan a las necesidades requeridas, como los estudios de detectores y pesos óptimos, y el análisis realizado para obtener resultados coherentes a partir de la información dada por los tres planos de color que componen las imágenes a color tratadas.

Una vez visto que el tiempo es aceptable pero mejorable, las **futuras líneas de investigación** deben de ir encaminadas a minimizar en todo lo posible dicho parámetro.

En primer lugar, es de esperar que el algoritmo sea implementado en lenguajes de programación más adecuados para el diseño de aplicaciones, como pueden ser C++ o Basic, o en sus versiones de creación de interfaces gráficas, Visual C y Visual Basic respectivamente, que faciliten su utilización con menús, opciones mejoradas y, en general, con un entorno en consideración con lo que hoy en día se entiende por una aplicación. La implementación en otro lenguaje minimizaría el tiempo computacional considerablemente, obteniendo así una respuesta más rápida.

Se podrían incluir más funciones que formen parte del mapa de coste local añadidas a las ya existentes, ya que a mayor número de funciones más información se tendrá sobre los contornos de los objetos. Por ejemplo funciones que detallen la intensidad dentro, fuera y en el mismo borde [3] y [8], o funciones que incluyan información sobre la textura, etc. También se podrían hacer modificaciones del método comprobando la mejora del mismo. Por ejemplo, en este proyecto el detector de contornos utilizado es el del *método de Canny*, en vez del *método de cruces por cero* aconsejado por los autores de [4] y [8], tal como evidenció el análisis de detectores de contornos realizado. De la misma manera se puede mejorar introduciendo detectores más complejos como los basados en **wavelet**.

El hecho de diseñar la herramienta en lenguajes de programación que optimicen el tiempo, hace pensar que es posible obtener una herramienta de segmentación más interactiva. Es decir, se podría diseñar dicha herramienta de manera que al pinchar el usuario un punto del objeto con el ratón y posicionar seguidamente el cursor del mismo en cualquier otro punto, se represente el segmento Live Wire correspondiente, de manera que si al usuario le interesa su creación proceda a pincharlo, y si no es así siga buscando posiciones mejores con el cursor del ratón. Esta forma de interactividad se denominada **On the fly**, y es aludida en la mayoría de los documentos consultados para la realización de este proyecto (ver [3], [4] y [8]).

En definitiva, las mejoras de este proyecto deben enfocarse a implementar el método en lenguajes más propios de una aplicación de cara al usuario que, a la vez, minimice el tiempo de segmentación, y a la adición y creación de funciones más complejas que ayuden en el proceso de detección de contornos.