

Herramienta de segmentación semiautomática de imágenes mediante visión por computador

Jorge Francisco Ortiz

Resumen—El proyecto propone una herramienta de segmentación semiautomática de imágenes usando visión por computador. El objetivo es poder segmentar imágenes que contienen gestos realizados con las manos para su posterior interpretación como parte de una aplicación posterior. Para llevar a cabo esta herramienta primeramente se realiza un estudio del estado del arte para poder analizar las capacidades de las herramientas actuales que ya realizan esta función, tratando de aprovechar lo que éstas nos ofrecen. Posteriormente al estudio se explica el desarrollo de la nueva herramienta, que consta de tres fases: preprocesamiento de la imagen, generación semiautomática de marcadores mediante el método *connected labeled components*, y la utilización del algoritmo *Watershed* que finalmente se encarga de hacer la segmentación final. Los resultados preliminares mostrados en este documento indican un gran potencial de la técnica desarrollada, ofreciendo resultados prometedores de segmentación sin requerir del usuario final más que marcar un punto en la imagen.

Palabras clave—Segmentación, Marcadores, Foreground, Preprocesado, Watersheds

Abstract— This project proposes a semi-automatic computer vision tool for segmenting images. The final objective is to segment images which contains persons performing a series of gestures; our methodology aims at segmenting both hands and the head as part of a bigger pipeline which would include the recognition of the actions performed. To develop our tool we have first carried out a research on the State-of-the-art on semi-automatic segmentation in order to explore the capabilities of existint approaches in order to obtain knowledge to create our tool. After this, we propose our segmentation methodology which consists of three stages: image preprocessing, semi-automatic foreground marker generation using *connected labeled components* and finally, the use of *Watersheds* to obtain the final segmentation. Preliminary results show the potential of our tool to be used for segmenting the proposed structures; our method is able to obtain promising segmentation results with minimal user interaction.

Index Terms— Segmentation, Markers, Foreground, Preprocessing, Watersheds



1 INTRODUCCIÓN

La segmentación de imágenes es el proceso de dividir una imagen en varias regiones constituidas por un conjunto de píxeles homogéneos y que presentan características comunes. La segmentación de imágenes es una técnica usada de manera frecuente como parte de aplicaciones que incluyen la categorización del contenido de regiones. Por ejemplo, ha sido usada en el campo de la medicina para detectar cáncer de piel [1], en el ámbito de la seguridad para detectar personas [2] o en aplicaciones como la detección de huellas dactilares para acceder a un recinto [3], entre otras utilidades.

Dentro de las posibles aplicaciones de la segmentación la que ha motivado este trabajo es la segmentación de estructuras del cuerpo humano para poder averiguar qué acción está realizando el individuo. En nuestro caso concreto, el objetivo es poder realizar una segmentación de las manos y la cara de una persona con el fin de poder estudiar, en un futuro, qué gestos está realizando ésta.

Una posible aplicación es la de poder mover un volumen 3D mediante gestos manuales y la Kinect [4].

1.2 Objetivos

El objetivo de este trabajo es realizar una herramienta de segmentación semiautomática que permita extraer dentro de una imagen una o varias regiones deseadas.

Con el fin de adquirir conocimiento para desarrollar una herramienta eficiente, se ha decidido realizar una comparativa previa de diferentes herramientas de segmentación ya existentes (*Meanshift*, *Grabcut*, *Levelset* y *Watershed*), valorando ventajas e inconvenientes de cada una de ellas para ver cuáles pueden ser útiles para la nueva herramienta. Tras la comparativa se desarrollará el método de segmentación propio aprovechando el aprendizaje adquirido tras realizar el estudio comparativo.

La herramienta desarrollada será validada con un conjunto de imágenes dentro del escenario de aplicación propuesto con el fin de analizar el rendimiento del nuevo método creado.

1.3 Requisitos del proyecto

La herramienta de segmentación a desarrollar tiene que cumplir una serie de requisitos:

- E-mail de contacte: Jorge.francisco.ortiz@gmail.com
- Menció realitzada: *Computación*.
- Treball tutoritzat per: Jorge Bernal (*Ciencias de la computación*)
Coen Antens (*Ciencias de la computación*)
- Curs 2014/15

Requisitos funcionales

- Debe permitir al usuario introducir una imagen a segmentar.
- Debe permitir al usuario marcar con un único punto la imagen a segmentar o varios si se desea segmentar varios objetos.
- Debe devolver el resultado de la segmentación sobrepuesto sobre la imagen original.

Requisitos no funcionales

- Debe ser fácil de usar.
- Debe poder utilizarse en cualquier entorno de trabajo.
- Debe tener un tiempo de respuesta bajo, incluyendo el marcado de los puntos por parte del usuario.

Requisitos de sistema

Para llevar a cabo el proyecto es necesario disponer de un ordenador, sobremesa o portátil, con Matlab instalado y una serie de requisitos mínimos que se pueden ver en la siguiente tabla:

Sistema Operativo	CPU	RAM
Windows XP o superior MAC Lion o superior Ubuntu 12.04 o superior	Procesador con tecnología Intel.	1024MB

Tabla 1: Requisitos mínimos

Uno de los objetivos de desarrollo una vez depurada la aplicación es posibilitar la generación de un fichero ejecutable para que la aplicación pueda funcionar sin necesidad de tener instalado Matlab.

1.4 Metodología

Para conseguir los objetivos propuestos se ha utilizado una metodología ágil de desarrollo basada en una metodología ya existente denominada Scrum [5].

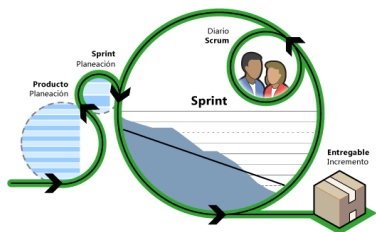


Fig.1: Metodología SCRUM

Nuestra aplicación de la metodología ha consistido en marcar objetivos a cumplir en diferentes sprints; cada semana se ha realizado una reunión y se han marcado los objetivos de cara al siguiente sprint –Fig. 1-. En estas reuniones se valoraba el trabajo realizado y se veía si continuar con los objetivos planificados o replantearlos en función de la dirección que iba tomando el proyecto.

1.5 Planificación del proyecto

Es necesario realizar una planificación inicial para el buen desarrollo del proyecto, tener un control de las tareas realizadas y poder observar si existe o no una desviación sobre la línea inicial marcada. Además de utilizar una metodología SCRUM se realizó un diagrama de Gantt (ver Apéndice A1) para marcar una serie de objetivos generales para el tiempo de desarrollo del proyecto. Dentro de este diagrama, se definieron las siguientes fases:

Fase de planificación

- Descripción del proyecto: Se especifica a grandes rasgos en que consiste el proyecto y los objetivos a conseguir.
- Elección del lenguaje de programación: Se analizan los pros y contras que proporcionan cada lenguaje de programación posible con el fin de escoger uno de ellos para el desarrollo de la herramienta.
- Preparación del entorno de desarrollo: El entorno de desarrollo se define a partir del lenguaje de programación escogido.
- Definir requisitos: Planteamiento definitivo de los requisitos.

Fase de comparativa del estado del arte

- Estudio técnicas de segmentación actual: Se realiza un estudio teórico de las diferentes herramientas de segmentación existentes para valorar ventajas e inconvenientes de las mismas. Se obtienen implementaciones de las diferentes herramientas para realizar una comparativa.
- Creación primer experimental Setup: se crea una BBDD con imágenes de prueba y se definen las métricas necesarias para comparar las herramientas del estado del arte.
- Estudios comparativos:
 - Estudio 1: Primer experimento valoración: Se realizan experimentos de validación de los diferentes algoritmos de segmentación estudiados.
 - Estudio 2: Impacto de técnicas de preprocesado de imágenes: Se aplican diferentes técnicas de preprocesado a la imagen para ver si el resultado se ve mejorado.
- Comparativa final técnicas: Se realiza una comparativa de los resultados obtenidos con cada técnica para utilizar aquello que pueda resultar útil.

Fase de desarrollo de la solución

- Desarrollo Algoritmo: Se desarrolla el nuevo algoritmo.
- Creación banco de pruebas más amplio: Se crea un banco de prueba de imágenes más amplio para realizar pruebas.
- Creación manual Ground truth: Se realiza una segmentación manual para utilizar como punto de comparación con la herramienta semiautomática de

segmentación.

- Valoración resultados: Se observan y se valoran los resultados obtenidos.

Fase de documentación

A lo largo de todo el proyecto se han ido redactando una serie de documentos para tener un control de las tareas que se iban haciendo.

1.6 Riesgos del proyecto

Como todo proyecto esta sometido a una serie de posibles contratiempos que pueden atrasar o dificultar su desarrollo. Podemos observar en la Tabla 2 los principales riesgos estimados.

Riesgo	Probabilidad	Impacto
Mala planificación	Baja	Crítico
No disponer de suficientes imágenes	Baja	Marginal
Objetivos poco realistas	Baja	Crítico
Falta de información	Media	Crítico
Herramientas de desarrollo inadecuadas	Baja	Crítico

Tabla 2: Riesgos estimados

1.7 Estructura del documento

Este documento sigue la siguiente estructura:

- **Introducción:** En este apartado se tratan los objetivos del proyecto tanto genéricos como específicos, se plantean los requisitos a cumplir la metodología a utilizar, la planificación a seguir y los posibles riesgos del proyecto.
- **Estado del arte:** En esta sección se explica de manera teórica y con ejemplos visuales los diferentes tipos de segmentación existentes.
- **Comparativa técnicas estado del arte:** Se muestran los primeros experimentos de validación, los resultados obtenidos y un análisis de los mismos.
- **Desarrollo del sistema inteligente de segmentación semiautomática:** A lo largo de esta sección se desarrolla un pipeline entero de procesado, se especifican los métodos utilizados en cada etapa, cómo afectan algunos elementos en el resultado final y la explicación de la nueva herramienta de segmentación
- **Resultados:** Se muestran los resultados de la nueva herramienta de segmentación, se hace una valoración de cómo de buena es la segmentación utilizando una serie de métricas. También se analizan las técnicas utilizadas y como han afectado al resultado final.
- **Análisis del desarrollo del método:** Se revisa y analiza el desarrollo del proyecto, si se han cumplido o no los objetivos y los problemas encontrados.
- **Conclusiones:** Finalmente en esta sección se realiza una revisión de lo aprendido y se especifica líneas futuras del proyecto.

2. ESTADO DEL ARTE

En la actualidad existen diferentes maneras y herramientas para la segmentación de regiones dentro de una imagen. En la bibliografía podemos encontrar 2 grupos de algoritmos para la segmentación de imágenes:

- Segmentación automática: Este tipo de segmentación no requiere intervención por parte del usuario, se proporciona una imagen y el resultado es la imagen segmentada.
- Segmentación semiautomática: En este caso es necesario la ayuda de el usuario para conseguir facilitar la segmentación, ya sea realizando algun tipo de aportación como por ejemplo seleccionar una zona de la región que se desea segmentar.

Teniendo en cuenta los objetivos y las restricciones temporales de nuestro proyecto, hemos decidido desarrollar una herramienta de segmentación semiautomática ya que se considera que la interacción con el usuario puede aportar ciertos beneficios tales como poder guiar a la herramienta para marcar las regiones de interés de la imagen. Actualmente existen numerosas herramientas de segmentación semiautomática, en este caso se ha decidido centrar el foco en cuatro de las más relevantes:

Watershed

Watershed[6][7] es una técnica de segmentación basada en regiones, debido a que clasifica los pixeles según su proximidad espacial y el gradiente de sus niveles de gris. Una imagen puede verse como una representación topografica de un terreno, donde a cada píxel se le asocia una altura relacionado con su nivel de gris correspondiente: las zonas más oscuras corresponden con valles y "basins" (mínimos locales) mientras que las zonas más claras se corresponden con montañas y crestas. A partir de esa imagen se inicia un algoritmo de llenado, como si empezase a llover sobre el terreno y se fuesen llenando de agua los valles hasta que las cuencas contiguas se unen, como se ve en la Fig. 2. Las líneas de unión, que representan las fronteras de regiones homogéneas consistuyen el resultado de la segmentación.

Uno de los problemas asociados al uso de watersheds radica en el exceso de mínimos locales (valles) que tenemos en una imagen, por ello es necesario marcar aquellos realmente significativos para guiar la segmentación. Este método se denomina Watershed con marcadores (WS) [8][9]. En Watershed con marcadores los marcadores pueden ser extraídos mediante operaciones morfológicas o proporcionados directamente por el usuario.

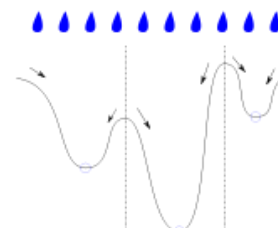


Fig.2 Ejemplo del algoritmo de inundación Watershed [10]

Grabcut

GrabCut (GC) es una técnica para la segmentación de imágenes que consiste en crear un grafo a partir de la imagen a segmentar, donde por cada píxel se genera un nodo que lo representa en el grafo. A continuación cada nodo se conecta con sus vecinos próximos a través de arcos no dirigidos [11]. El algoritmo define 2 nodos especiales en el grafo de flujo: Fuente (source) y destino (sink). El nodo fuente representa el objeto a segmentar en la imagen (foreground), y el destino representa el fondo de la imagen (background). Podemos encontrar ejemplos de implementación de GrabCut en [11,12,13].

Cada nodo se conecta a estos nodos especiales midiendo la probabilidad de que sean foreground o background. Estas probabilidades se pueden obtener a partir de similitud de colores, texturas, distancias entre el píxel y la fuente o destino entre otras.

A partir del grafo de flujo es posible calcular el algoritmo de corte mínimo [14] el cual divide el grafo en dos grafos disjuntos como se muestra en la Fig.3. Así, un grafo poseerá el nodo fuente con un grupo de píxeles, y el otro al nodo destino con el grupo de píxeles restantes.

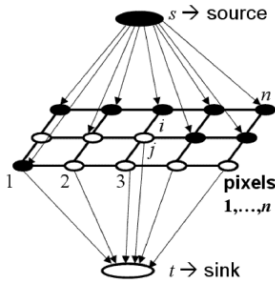


Fig.3 Grafo de flujo [15]

Meanshift

El algoritmo de segmentación Meanshift (MS) [16] representa el espacio de datos como una función de densidad. Una función de densidad es una función matemática que caracteriza el comportamiento probable de una población, especificando la probabilidad relativa de que una variable aleatoria X tome un valor cercano a x .

Regiones densas en el espacio de características corresponde a un máximo sobre la mencionada función, estos máximos son denominados modas.

Meanshift agrupa un conjunto de datos asociando a cada punto con la moda o pico más probable. Para cada punto el algoritmo calcula la moda correspondiente y la asocia a él: En primer lugar, se define un kernel radial centrado en el punto y con un radio de cobertura. A continuación se calcula la media de los puntos que caen dentro del área definida bajo el kernel. Después, el algoritmo desplaza la ventana o el kernel hacia la media y repite el proceso hasta alcanzar convergencia, es decir, hasta que el vector calculado recursivamente se mantiene constante o hasta que su variación no supera un umbral determinado [17].

En cada iteración, la ventana se desplazará hacia zonas de mayor densidad de puntos hasta que se alcance el pico o

moda, donde los datos están distribuidos de la misma manera.

Una vez se ha encontrado la moda para cada punto, se asociarán en un grupo común todas aquellas que se encuentren en un contorno determinado. Un ejemplo gráfico de MeanShift es mostrado en la Fig. 4.

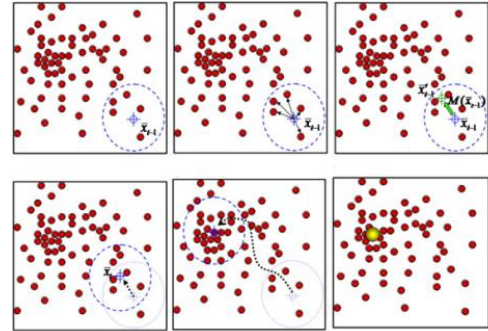


Fig.4 Proceso Meanshift[18]

Level Set

Level Sets (LS) [19][20] se basa en la aplicación de métodos numéricos para rastrear la evolución de contornos y superficies denominados snakes que son colocados inicialmente sobre la imagen y van modificándose hasta encontrar bordes y adquirir la forma de las zonas de interés. Un snake puede ser una curva o superficie que se deforma en dirección de características de interés en la imagen como líneas, bordes, y es controlado a través de una ecuación diferencial (Eq.1) que establece en valor de la función LevelSet basada en tres velocidades: velocidad de propagación, que es la responsable de la extensión del snake hacia dentro o hacia fuera; velocidad de curvatura: responsable de controlar la forma del Snake y velocidad de advección, que es la más crítica y es responsable de que el snake frene ante la presencia de bordes en las regiones.

$$\frac{d}{dt} I = -\alpha A(x)|\nabla I| - \beta P(x)|\nabla I| + \gamma Z(x)k|\nabla I| \quad (1)$$

, donde A es la velocidad de advección, es la variación de un escalar en un punto dado por efecto de un campo vectorial, P es la velocidad de propagación, Z es el modificador de la curvatura k , α , β y γ proporcionan más peso o menos a cada velocidad en cada movimiento del snake. Fig. 5 muestra un ejemplo de la evolución en cada etapa de la segmentación mediante LevelSets.

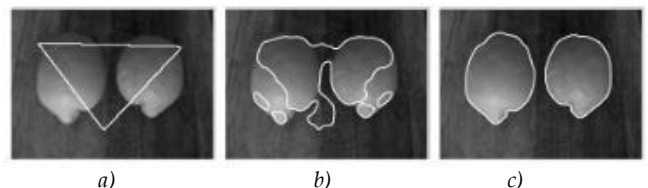


Fig.5 Proceso Levelset: a) Marcador inicial; b) Proceso de segmentación; c) Resultado final [21]

3. COMPARATIVA TÉCNICAS ESTADO DEL ARTE

3.1 Primer experimento de valoración

Definición de la base de datos

Para llevar a cabo la comparativa de las diferentes técnicas mencionadas anteriormente lo primero es disponer de una base de datos con las imágenes a utilizar.

En esta comparativa las imágenes utilizadas han sido imágenes frontales de personas realizando gestos con las manos en diferentes fondos. A continuación se muestra un ejemplo de algunas imágenes utilizadas:



Fig.6 Ejemplo de contenido de la ase de datos manos y caras

Para realizar la comparativa entre algoritmos se ha realizado un *Ground Truth* manual de las diferentes imágenes, para ver como de buena es la segmentación que estos realizan. En la fig.7 se aprecia el *Ground Truth* manual para las imágenes de ejemplo la Fig.6.



Fig.7 Ground truth manual

Métricas

Para valorar los resultados que proporcionan las diferentes técnicas es necesario definir métricas para poder evaluar la calidad de la segmentación y poder comparar con el *Ground Truth*. Se han definido una serie de conceptos a partir de los cuales se calcularan las métricas de evaluación.

- Píxeles resultantes de la segmentación manual **Asm**.
- Píxeles resultantes de la segmentación semiautomática del algoritmo **Ass**.
- Píxeles común entre la segmentación semiautomática y manual **Ac**.

Teniendo en cuenta estos factores se han utilizado 3 de métricas de evaluación de segmentación más frecuentemente empleadas [22].

Precision

$$Precision = \frac{\#Ac}{\#Asm} * 100$$

Un *Precision* del 100% representa que el algoritmo concreto ha marcado como parte del objeto todos los píxeles que han sido marcados como parte del objeto en la imagen segmentada manualmente. Esto no implica que la segmentación sea perfecta ya que el *Precision* no es capaz de detectar si se han añadido píxeles extras.

Recall

$$Recall = \frac{\#Ac}{\#Ass} * 100$$

Un *Recall* al 100% representa que en la región proporcionada no hay contenido perteneciente a otra región de la imagen. El *Recall* no es capaz de determinar si en la región final faltan píxeles de la región original.

DICE

$$DICE = 2 * \frac{\#Ac}{\#Asm + \#Ass} * 100$$

Un *DICE* del 100% nos indica que los píxeles indicados por el algoritmo son exactamente los mismos que los indicados en la segmentación manual.

Resultados

Para poder observar el resultado que proporcionan dichas técnicas se han realizado pruebas con diferentes implementaciones de las mismas descargadas de [23,24,25]. Podemos observar en Fig. 8 una comparativa visual de resultados de segmentación donde podemos observar la imagen inicial a segmentar, la imagen segmentada manualmente, el resultado de la segmentación que proporciona el algoritmo y la imagen diferencia (diferencia de píxeles entre la segmentación manual y la segmentación semiautomática) para cada una de las 4 técnicas más relevantes.

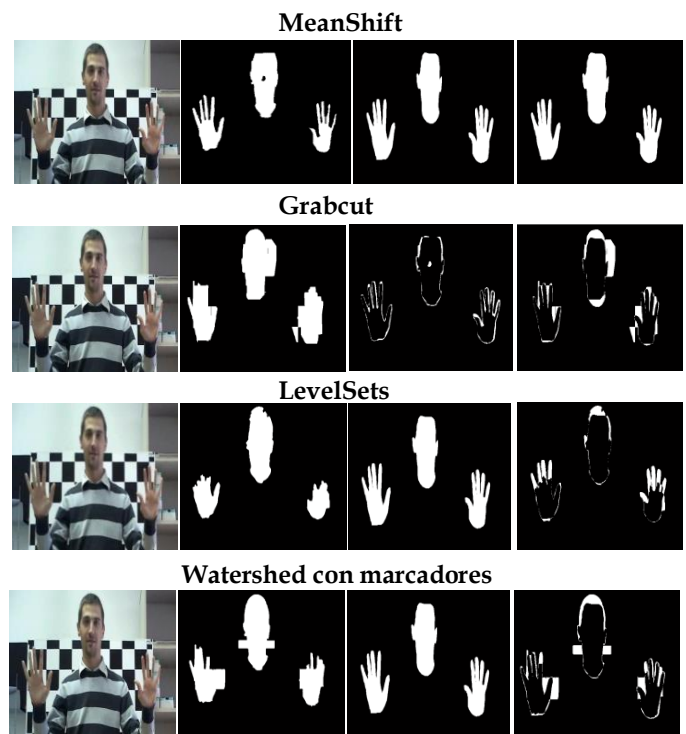


Fig.8 Comparativa visual métodos estado del arte: (primera columna) imagen original; (segunda columna) segmentación proporcionada por el método; (tercera columna) Ground truth manual; (cuarta columna) imagen diferencia.

Para hacer un análisis de la calidad del resultado a nivel cuantitativo se han utilizado las métricas comentadas anteriormente. En el apéndice A2 se pueden observar un desglose de los resultados obtenidos.

Podemos observar en la Tabla 3 una gráfica con el resultado de las métricas del estudio realizado.

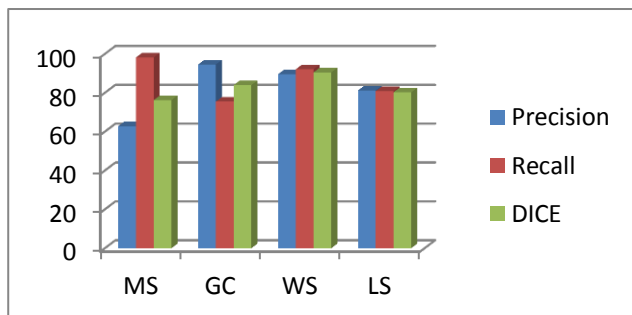


Tabla 3. Gráfica resultado precision, recall, DICE

Se puede observar que Grabcut tiene un *Precision* promedio más alto, esto implica que Grabcut es la técnica que se deja menos píxeles pertenecientes al objeto sin coger seguido muy de cerca por Watershed.

En resumen,

- Meanshift: MS proporciona muchas regiones como salida y elementos que deberían ser una única región las divide en varias (sobre segmentación).
- Grabcut: GC apenas deja píxeles del objeto sin coger pero añade muchos que no forman parte del mismo.
- LevelSet: Da un buen resultado si el objeto a segmentar es homogéneo y el contorno es similar al objeto a segmentar. También depende de la cantidad de iteraciones (fijadas en 100 para el experimento).
- Watershed con marcadores: Da un mejor resultado respecto a los demás. Es el algoritmo que proporciona menos falsos positivos.

A pesar de que Grabcut da un *Precision* un poco más elevado, y Meanshift un *Recall* más alto, si se miran las 3 medidas en conjunto Watershed nos proporciona un mejor resultado, por lo que Watershed es el algoritmo escogido para seguir trabajando.

4. SISTEMA INTELIGENTE DE SEGMENTACIÓN SEMI-AUTOMÁTICA

4.1 Pipeline de procesado

Nuestro sistema de segmentación semiautomático consta de 3 partes claramente diferenciadas, la primera de ellas consiste en realizar un preprocesado de la imagen previo a su segmentación, la segunda parte es la creación automática de marcadores de *foreground* a partir de unos marcadores definidos por el usuario y la tercera y última es la aplicación del algoritmo Watershed con marcadores.

4.1.1 Preprocesado de imagen

Realizar un preprocesado de la imagen puede ayudar a proporcionar mejores resultados en la segmentación.

El primer aspecto a tratar en cuanto al preprocesado de la imagen es el cambio de espacio de color, utilizar el espacio de color HSV nos permite detectar con mayor facilidad la similitud de colores para la creación del marcador [26]. Esto es debido a que el espacio de color HSV esta compuesto por 3 canales: H y S son los canales cromáticos y V la luminosidad. Esto permite mediante la eliminación del canal V obtener el color de la imagen sin verse afectado por la iluminación.

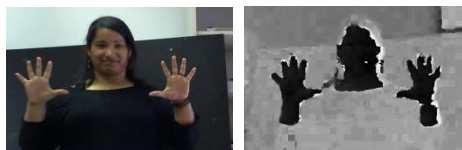


Fig.9 Imagen mostrada en espacio de color (izquierda) RGB y (derecha) H

Otra posibilidad estudiada ha consistido en probar la aplicación de un filtro gaussiano con el fin de eliminar ruido innecesario dentro de la imagen.

4.1.2 Creación marcador automático para foreground

Para evitar la sobresegmentación de Watershed es necesario proporcionar unos marcadores para indicar aquellos mínimos realmente importantes que deben guiar a la técnica. Estos marcadores son píxeles de la imagen proporcionados por el usuario que indican que esos píxeles pertenecen al *foreground*.

En este caso nuestro método proporciona una herramienta de creación automática de marcadores de *foreground* a partir de puntos definidos por el usuario. Para proporcionar estos marcadores se ha realizado una implementación del algoritmo denominado connected components labeling [27]. El algoritmo permite marcando un punto inicial en la región a segmentar obtener el marcador. Una vez se dispone del marcador este será el que se le indicará a Watershed como marcador de *foreground*.

En la Fig.10 se puede observar el proceso de la creación del marcador para esta imagen.

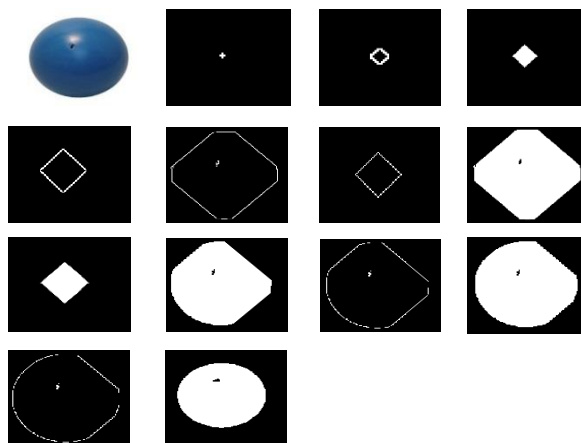


Fig.10 Proceso BFS para creación marcador

En la Fig.11 se puede observar el diagrama de flujo del proceso de creación de marcadores.

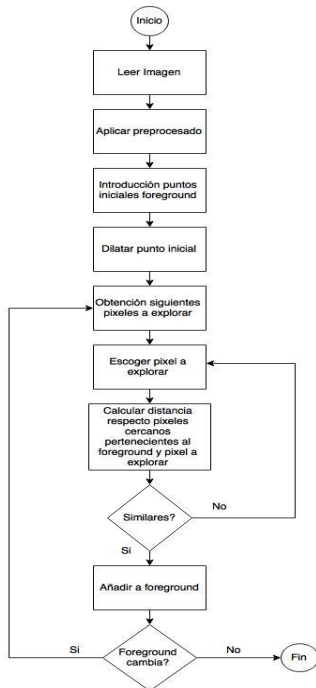


Fig.11 Diagrama de flujo algoritmo

Nuestro algoritmo lee inicialmente la imagen de la cual se desea realizar la segmentación, a la cual se le aplican técnicas de preprocesado para facilitar su tratamiento. A continuación se pide al usuario marcar un punto inicial y a partir de esta semilla, la región va creciendo al observar los píxeles de alrededor, comprobando si son o no similares basándose en un umbral de similitud. En este caso el umbral de similitud consiste en que la diferencia entre el píxel a explorar y el más cercano sea inferior a un porcentaje, fijado por defecto en 10%. Esto implica que el nuevo píxel es como mínimo un 90% similar al píxel más cercano. En caso de cumplir esa condición ese píxel pasa a ser considerado parte del *foreground*. Una vez observado todos los píxeles vecinos se compara los nuevos píxeles candidatos a *foreground* con los ya considerados *foreground* en búsqueda de similitud. Este proceso se realiza de manera iterativa hasta que los píxeles que forman el *foreground* no cambian de una iteración a otra.

4.1.3 Watershed con marcadores

Una vez definidos los marcadores del *foreground*, para poder emplear Watershed se han de definir los marcadores del fondo. Se ha probado de situar diferente cantidad de marcadores de fondo en diferentes zonas de la imagen, tanto puntos lejanos a la región a segmentar como muy próximos. Se ha observado que permite corregir zonas que el gradiente determina como *foreground* para pasar a ser consideradas *background*.

Watershed necesita unos marcadores y una función de energía para poder realizar la segmentación. Se ha probado el gradiente como función de energía. Para calcular dicho gradiente se han utilizado diferentes operadores

diferenciales como son:

- Sobel
- Prewitt
- Roberts
- Laplacian

Estos operadores permiten obtener una aproximación del gradiente para cada punto de la imagen [28]. Se ha escogido el gradiente debido a que este permite detectar la variación de nivel de gris entre píxeles consecutivos (contornos).

Tras realizar todo el *pipeline*, desde el preprocesado de la imagen, la obtención del marcador y la obtención de la función de energía el último paso es la aplicación del algoritmo Watershed al cual se le pasa como parámetros los tres elementos (marcadores *foreground*, marcadores *background* y función de energía) obtenidos en las fases anteriores.

5 RESULTADOS

5.1 Resultados globales

A continuación Fig.12 se puede observar el resultado que proporciona la herramienta después de aplicar el pipeline completo de procesado.

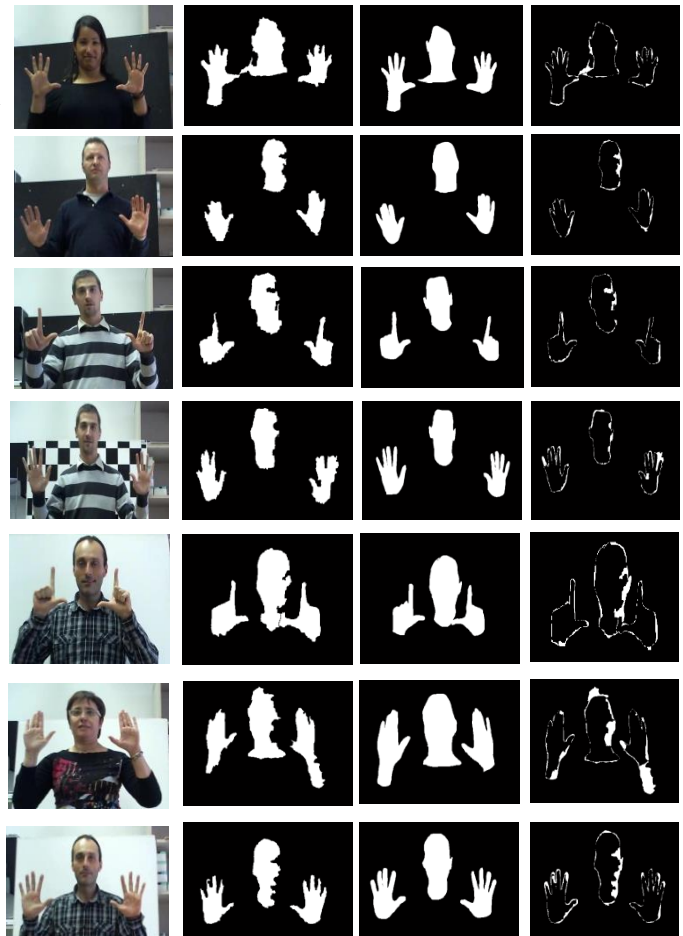


Fig.12 Resultado segmentación final (Imagen original, resultado segmentación, GT manual e imagen diferencia) (Imágenes del mismo tamaño)

Utilizando las métricas mencionadas anteriormente se obtienen los siguientes resultados contemplados en la Tabla 4:

	Precisión %	Recall %	DICE %
Imagen 1	94.3501	91.9248	93.1217
Imagen 2	91.1211	94.9592	93.0005
Imagen 3	91.6636	95.6878	93.6324
Imagen 4	89.8586	92.4851	91.1529
Imagen 5	88.6866	97.8062	93.0234
Imagen 6	91.8396	82.8709	87.1450
Imagen 7	87.1174	87.1668	87.1421
Imagen 8	85.8737	96.4842	90.8702
Promedio	90.0683	92.4231	91.1360

Tabla 4: Resultado métrica Precisión, Recall y DICE

5. 2. Experimentos de validación

En esta subsección se analiza el impacto de cada una de las etapas del pipeline en la obtención del resultado final de segmentación.

5.2. 1 Efecto del preprocesado

5.2.1.1 Cambio espacio de color

El primer aspecto a tratar en cuanto al preprocesado de la imagen es el cambio de espacio de color con el fin de observar cómo afecta al resultado el utilizar una imagen en el espacio de color RGB o HSV.

Utilizar un espacio de color u otro tiene un impacto directo en la creación del marcador y la generación de la función de energía como veremos más adelante.

5.2.1.2 Suavizado gaussiano

Suavizar la imagen con un filtro gaussiano permite eliminar ruido innecesario en la imagen. El principal problema de aplicar un suavizado gaussiano es que elimina ruido pero también suaviza los contornos. Para la segmentación el contorno tiene un papel muy importante por lo que suavizar los mismos no proporciona ningún beneficio por lo que se ha descartado su uso.

5.2.2 Generación marcadores *foreground*

La elección del punto inicial es un elemento clave para la obtención de un buen marcador. Dependiendo del punto seleccionado el marcador obtenido es uno u otro.

En la Fig.13 se puede observar como dado 2 puntos de la cara el algoritmo implementado genera marcadores distintos. Esto es debido a que el punto seleccionado tiene como valor de píxel un valor muy distinto al resto, en la imagen original no se puede apreciar pero si se observa la imagen en el espacio HSV se puede ver la diferencia.



Fig.13 Resultado marcadores cabeza

Otro factor importante que afecta a la creación del marcador es el porcentaje de similitud escogido.

Cuanto más estricto es el criterio de similitud más pequeño es el marcador generado. Es conveniente no ser muy estricto para que el marcador no sea muy pequeño ni muy permisivo para que el marcador no coja zonas ajenas a la región a segmentar. Es conveniente encontrar un punto medio que proporcione un buen marcador tanto para las manos como para la cabeza. En la Fig.14 se pueden observar diferentes marcadores dependiendo del criterio de similitud establecido.



Fig.14 Marcadores manos similitud del (a) 95%; (b) 90%; (c) 85% y (d) 80%.

Es esencial una buena combinación de ambos factores ya que puede darse el caso de que el marcador sobrepase y de una mala segmentación como se puede ver en la fig.15.



Fig.15 Ejemplo de marcador mano sobrepasado

Otro factor que afecta a la creación de los marcadores es el espacio de color escogido. En el caso de la imagen RGB el marcador tiende a sobrepasar la región a segmentar mientras que la imagen H encuentra el marcador dentro de los límites de la región a segmentar.

A continuación (Figs. 16 y 17) podemos observar el resultado de la segmentación tras la creación de los marcadores utilizando el canal RGB y el canal H con el mismo punto inicial y mismo criterio de similitud además de un análisis métrico (Tablas 5 y 6).

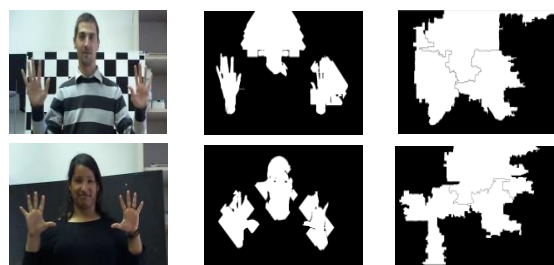


Fig.16 Resultado segmentación RGB

RGB	Precisión %	Recall %	DICE %
Imagen 1	94.5703	15.2241	26.2262
Imagen 2	99.5483	10.5773	19.1227

Tabla 5 resultados RGB

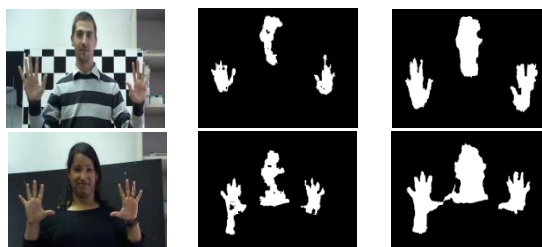


Fig.17 Resultado segmentación HSV

H	Precision %	Recall %	DICE %
Imagen 1	89.8586	92.4851	91.1529
Imagen 2	94.3501	91.9248	93.1217

Tabla 6 resultados HSV

En vista de los resultados obtenidos se ha considerado que para la creación de marcadores es mejor utilizar el espacio de color HSV.

La selección del *background* no tiene tanta importancia en la segmentación como los elementos anteriores pero sí es clave que entre los píxeles escogidos como *background* no se cojan píxeles del *foreground* o región a segmentar. También sirve para forzar que una zona que pertenece a *foreground* según gradiente pase a ser *background*.

5.2.3 Función de energía

Finalmente otro elemento que tiene gran peso en el resultado final de la segmentación es la función de energía empleada para guiar los Watersheds. En este caso se ha empleado el gradiente obtenido mediante Sobel como función de energía el cual permite detectar zonas en las que el nivel de gris cambia bruscamente (contornos). El espacio de color escogido afecta a la función de energía obtenida y al resultado final.

A continuación (Figs. 18 y 19) se puede observar el gradiente obtenido para imágenes en espacio de color RGB, HSV, el resultado final de la segmentación y un análisis métrico de los resultados (Tablas 7 y 8).

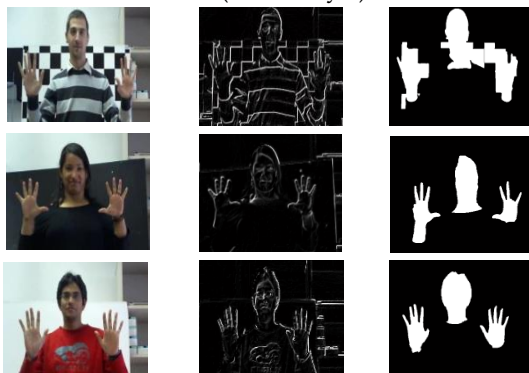


Fig.18 Resultado segmentación RGB

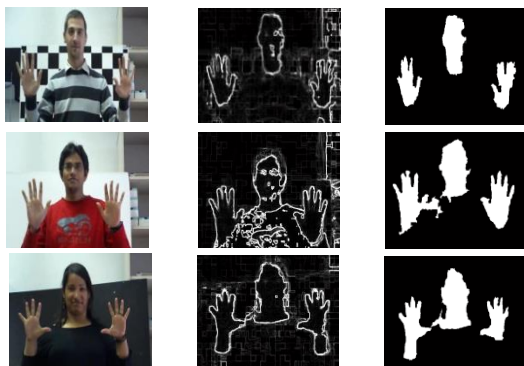


Fig.19 Resultado segmentación HSV

RGB	Precisión %	Recall %	DICE %
Imagen 1	96.1890	56.5959	71.2623
Imagen 2	84.2858	99.6400	91.3220
Imagen 3	89.7103	86.5054	88.0787

Tabla 7 Resultado segmentación RGB

H	Precisión %	Recall %	DICE %
Imagen 1	89.8586	92.4851	91.1529
Imagen 2	94.3501	91.9248	93.1217
Imagen 3	88.2243	94.9003	91.4406

Tabla 8 Resultado segmentación HSV

Por lo que respecta al espacio de color RGB se puede observar los 3 tipos de resultados posibles, uno bueno como es la imagen 3 en la que la segmentación es muy buena pero coge el pelo como elemento de la cara, otro algo menos bueno como es la imagen 2 en la que desaparecen algunos elementos como son algunos dedos y un resultado muy malo como es el caso de la imagen 1 donde coge elementos del fondo como parte de la región a segmentar. En la imagen en espacio de color H se ve como no se omiten elementos, aunque hace un poco más ancha la región a segmentar.

En vista de los resultados se puede decir que el gradiente de una imagen en espacio de color H proporciona mejores resultados que la imagen en espacio de color RGB para este tipo de imágenes.

Esto se debe a al hecho de que el gradiente en las imágenes RGB es menos pronunciado que en las imágenes H al existir menos diferencia de nivel de gris entre las diferentes regiones.

El gradiente afecta más a las caras debido a que existen diferentes elementos dentro de ella que generan un cambio de nivel de gris, como pueden ser la boca, la nariz o los ojos como se puede ver en la fig.20.

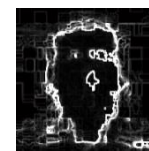


Fig.20 Gradiente imagen de cara

Otro factor a tener en cuenta es la problemática que tiene el gradiente en los casos en los que la región a segmentar se encuentra muy próxima a un elemento muy parecido (cambio de nivel de gris mínimo). En la fig.21 se puede apreciar como el color de la mano es prácticamente idéntico a la franja de la camisa provocando que el valor de gradiente sea mínimo.



Fig.21 Similitud de color generando problemas de gradiente

Hasta el momento se puede observar que es necesario que la imagen se encuentre bien iluminada (menor diferencia entre píxeles) y que el punto o puntos iniciales de marcación se encuentren en una zona similar a sus vecinos y no en una zona aislada.

6. ANÁLISIS DEL DESARROLLO DEL MÉTODO

6.1 Revisión de planificación

Una vez finalizado el proyecto se ha realizado un análisis de la planificación inicial. Se ha observado que se han cumplido la mayoría de plazos estipulados. Se ha producido un retraso en el inicio de la fase de desarrollo de la herramienta de segmentación debido a que se ha ampliado el tiempo de estudio del arte. Esto nos ha quitado tiempo para tratar de poner en práctica algunas técnicas para tratar de obtener un mejor resultado. A pesar de eso el resultado es satisfactorio.

6.2 Problemas encontrados

Inicialmente se tenía en mente realizar una herramienta que pudiera servir para segmentar cualquier tipo de imagen pero tras realizar la investigación previa y la comparativa del estado del se ha observado que era un objetivo demasiado ambicioso. Por eso se decidió centrar el foco de desarrollo de la herramienta en la segmentación de un tipo de imágenes específicas como son imágenes de manos y caras. Otro problema encontrado fue la problemática para segmentar imágenes en condiciones adversas de iluminación.

7. CONCLUSIONES

Se ha conseguido realizar en gran parte los objetivos propuestos al inicio del proyecto. Se ha presentado un estudio de los diferentes métodos de segmentación semiautomáticos actuales más relevantes como son MeanShift, LevelSet, GrabCut y Watershed. El estudio comparativo muestra que Watershed es la herramienta que permite un compromiso entre calidad de resultados obtenidos y tiempo de computación para un mismo marcador inicial. Como resultado de este proyecto, se ha conseguido realizar una herramienta de segmentación semiautomática que emplea watersheds con marcadores definidos a partir de una posición inicial marcada por el usuario. El proceso de creación de marcadores permite obtener una buena primera estimación de la zona que ocupa el objeto a segmentar. Esta estimación se ha demostrado mediante un experimento preliminar de validación que permite obtener resultados óptimos de segmentación en todas las métricas empleadas.

7.1 Revisión aprendizaje

Uno de los objetivos principales del proyecto era afianzar el conocimiento sobre procesado de imágenes. Este objetivo se ha cumplido con creces dado que se ha podido diseñar desde cero un pipeline completo de procesado. Asimismo se han adquirido capacidades relacionadas con organización y gestión de proyectos: se ha aprendido a organizar un proyecto más amplio de lo habitual desde su principio a fin, se ha aprendido a gestionar el tiempo,

gestionar momentos de crisis y a replanificar y priorizar objetivos.

7.2 Líneas futuras

En un futuro se desearía poder ampliar la funcionalidad de la herramienta para poder segmentar imágenes de cualquier tipo y bajo diferentes factores. Asimismo se plantea realizar una validación más extensiva del algoritmo desarrollado, incluyendo la evaluación de su funcionamiento en condiciones adversas de imagen (mala iluminación, similitud *foreground-background*) que no se ha podido realizar en profundidad por falta de tiempo.

8. AGRADECIMIENTOS

Primeramente me gustaría agradecer a Jorge Bernal y a Coen Antens por llevar a cabo mi proyecto y haberme guiado en el desarrollo del mismo y ayudarme a seguir en los momentos más complicados y por estar disponibles en cualquier momento en los que requería de su ayuda. También me gustaría agradecer a Debora Gil y a Samuel Esteban por permitirme utilizar la base de datos con la que realizar las pruebas y a todos los que participaron en la creación de la misma. Por último me gustaría agradecer a mis padres y amigos por soportarme en los momentos en los que no me salían las cosas y no encontraba el camino a seguir.

9. BIBLIOGRAFIA

- [1] Hance, G., Umbaugh, S. E., Moss, R. H., & Stoecker, W. V. (1996). Unsupervised color image segmentation: with application to skin tumor borders. *Engineering in Medicine and Biology Magazine, IEEE*, 15(1), 104-111.
- [2] Rougier, C., Meunier, J., St-Arnaud, A., & Rousseau, J. (2007, May). Fall detection from human shape and motion history using video surveillance. In *Advanced Information Networking and Applications Workshops, 2007, AINAW'07. 21st International Conference on* (Vol. 2, pp. 875-880). IEEE.
- [3] Hennings-Yeomans, P. H., Kumar, B. V. K. V., & Savvides, M. (2007). Palmprint classification using multiple advanced correlation filters and palm-specific segmentation. *Information Forensics and Security, IEEE Transactions on*, 2(3), 613-622.
- [4] Lopes, O., Reyes, M., Escalera, S., & Gonzalez, J. (2014). Spherical blurred shape model for 3-d object and pose recognition: Quantitative analysis and hci applications in smart environments. *Cybernetics, IEEE Transactions on*, 44(12), 2379-2390.
- [5] Schwaber, K. (1997). Scrum development process. In *Business Object Design and Implementation* (pp. 117-134). Springer London.
- [6] Mangan, A. P., & Whitaker, R. T. (1999). Partitioning 3D surface meshes using watershed segmentation. *Visualization and Computer Graphics, IEEE Transactions on*, 5(4), 308-321.
- [7] Beucher, S. (1992). The watershed transformation applied to image segmentation. *SCANNING MICROSCOPU SUPPLEMENT*, 299-299.
- [8] Lotufo, R., & Silva, W. (2002). Minimal set of markers for the watershed transform. In *Proceedings of ISMM* (Vol. 2002, pp. 359-368).

[9] Zhang, X., Jia, F., Luo, S., Liu, G., & Hu, Q. (2014). A marker-based watershed method for X-ray image segmentation. *Computer methods and programs in biomedicine*, 113(3), 894-903.

[10] https://es.wikipedia.org/wiki/Transformaci%C3%B3n_divisoria

[11] Han, S., Tao, W., Wang, D., Tai, X. C., & Wu, X. (2009). Image segmentation based on GrabCut framework integrating multiscale nonlinear structure tensor. *Image Processing, IEEE Transactions on*, 18(10), 2289-2302.

[12] Talbot, J. F., & Xu, X. (2006). Implementing grabcut. Brigham Young University.

[13] Rother, C., Kolmogorov, V., & Blake, A. (2004). Grabcut: Interactive foreground extraction using iterated graph cuts. *ACM Transactions on Graphics (TOG)*, 23(3), 309-314.

[14] Boykov, Y., & Kolmogorov, V. (2004). An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26(9), 1124-1137.

[15] <https://www.cs.auckland.ac.nz/courses/compsci773s1c/lectures/773-GG/lectBc-773.htm>

[16] Comaniciu, D., & Meer, P. (2002). Mean shift: A robust approach toward feature space analysis. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(5), 603-619.

[17] Comaniciu, D., Ramesh, V., & Meer, P. (2000). Real-time tracking of non-rigid objects using mean shift. In *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on (Vol. 2, pp. 142-149)*. IEEE.

[18] <http://www.cnblogs.com/easymind223/archive/2012/07/03/2574887.html>

[19] Sethian, J. A. (1999). *Level set methods and fast marching methods: evolving interfaces in computational geometry, fluid mechanics, computer vision, and materials science (Vol. 3)*. Cambridge university press.

[20] Gavidia, G., Soudah, E., Martín-Landrove, M., & Cerrolaza, M. (2011). Generación de modelos discretos de tejidos del ser humano a través del pre procesamiento y segmentación de imágenes médicas. *Revista Internacional de Métodos Numéricos para Cálculo y Diseño en Ingeniería*, 27(3), 200-226.

[21] <http://n0143855.servername.com/qredirect.php?doi=ajaps.2011.101.111&linkid=ref>

[22] Riaz, F., Ribeiro, M. D., & Coimbra, M. T. (2009, September). Quantitative comparison of segmentation methods for in-body images. In *Engineering in Medicine and Biology Society, 2009. EMBC 2009. Annual International Conference of the IEEE (pp. 5785-5788)*. IEEE.

[23] <http://grabcut.weebly.com/code.html> Grabcut implementation.

[24] <http://coewww.rutgers.edu/riul/research/code/EDISON/index.html>. Meanshift implementation.

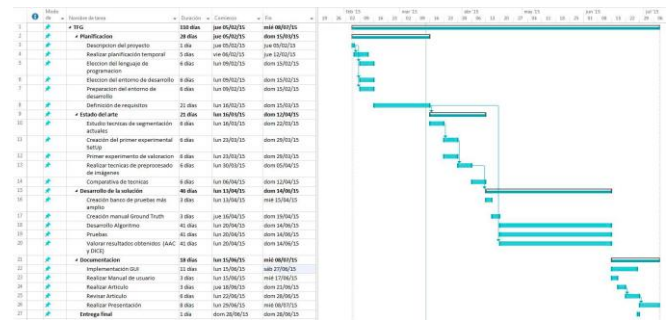
[25] <http://www.mathworks.com/matlabcentral/fileexchange/12711-level-set-for-image-segmentation>. LevelSet implementation.

[26] Sural, S., Qian, G., & Pramanik, S. (2002). Segmentation and histogram generation using the HSV color space for image retrieval. In *Image Processing. 2002. Proceedings. 2002 International Conference on (Vol. 2, pp. II-589)*. IEEE.

[27] Di Stefano, L., & Bulgarelli, A. (1999). A simple and efficient connected components labeling algorithm. In *Image Analysis and Processing, 1999. Proceedings. International Conference on (pp. 322-327)*. IEEE.

[28] Wang, W., Bergholm, F., & Yang, B. (2003). Froth delineation based on image classification. *Minerals Engineering*, 16(11), 1183-1192.

Anexo A1: Diagrama de Gantt del proyecto



Anexo A2: Resultados preliminares estudio comparativo

Meanshift



Fig.22 Imagen original, resultado segmentación, GT manual e imagen diferencia Grabcut

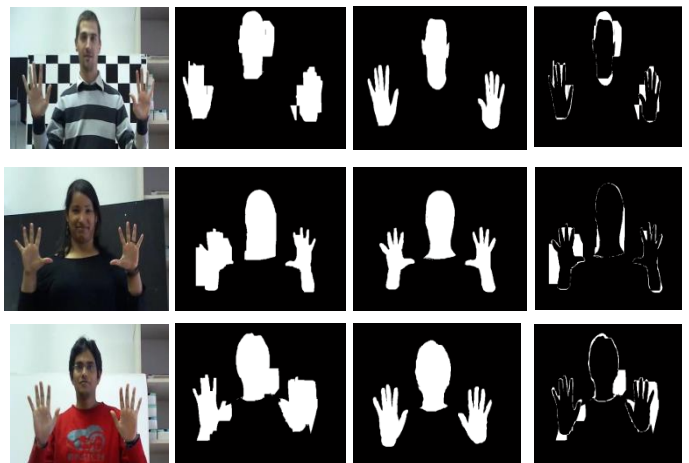


Fig.23 Imagen original, resultado segmentación, GT manual e imagen diferencia

LevelSet

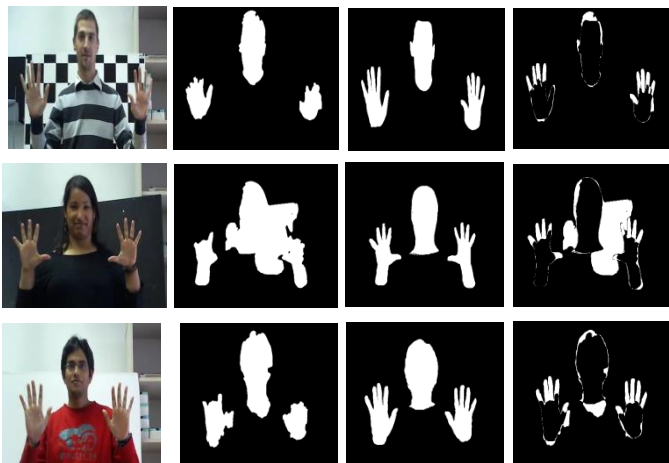


Fig.24 Imagen original, resultado segmentación, GT manual e imagen diferencia

Watershed

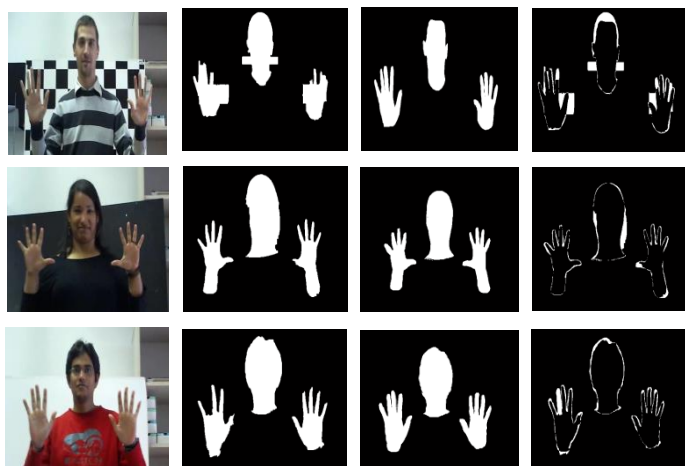


Fig.25 Imagen original, resultado segmentación, GT manual e imagen diferencia