

UN MÉTODO DE SEGMENTACIÓN DE IMÁGENES DIGITALES TRIDIMENSIONALES POR CRECIMIENTO DE REGIONES

Mariana del Fresno¹, Marcelo J. Vénere^{1,2}

¹ PLADEMA-ISISTAN, Universidad Nacional del Centro, 7000 Tandil
Tel/Fax:(02293) 442202 E-mail: mdelfres@exa.unicen.edu.ar

² también Comisión Nacional de Energía Atómica

RESUMEN

El proceso de segmentación de imágenes digitales 3D en objetos de interés es de gran importancia, especialmente en el campo de la medicina. No obstante, si bien son varios los enfoques planteados, no existen aún soluciones definitivas ni algoritmos generalmente aplicables. En este trabajo se presenta una estrategia para la segmentación de imágenes digitales tridimensionales basada en el crecimiento de regiones. Además, el algoritmo se complementa con la generación inmediata de la descripción por triangulación de las superficies obtenidas, incluyendo su suavizado y tratamiento de situaciones conflictivas. El esquema propuesto resulta en una alternativa más robusta y a la vez económica que otros métodos actualmente empleados.

Palabras clave: computación gráfica, segmentación de imágenes, visualización

1. INTRODUCCIÓN

El avance de la tecnología computacional ha transformado considerablemente las características del tratamiento de la información, en diferentes aspectos. En el área de la medicina, las alternativas de captura y análisis de datos han evolucionado notablemente hasta llegar actualmente a permitir el desarrollo de aplicaciones de tratamiento de imágenes tridimensionales, generadas a partir de la información proveniente de dispositivos de captura, tales como equipos de tomografía computada o resonancia magnética. Este tipo de herramientas pueden ofrecer asistencia a los profesionales y mejorar las características del diagnóstico, al facilitar el análisis de la información directamente sobre los datos tridimensionales, en un proceso de características no invasivas. Además, debido a la evolución producida en la performance de computadoras de bajo costo, se ha hecho posible experimentar con soluciones que antes estaban restringidas a grandes equipamientos, debido a la demanda computacional de los algoritmos involucrados.

Un aspecto de interés consiste en la habilidad de detectar determinadas estructuras de interés dentro de la imagen con características similares, mediante un proceso conocido como segmentación. Esta constituye una etapa esencial que condiciona los resultados del procesamiento de imágenes en diverso tipo de aplicaciones avanzadas. Por ejemplo, en el caso de la medicina, es factible la mejora de distinto tipo de prácticas, tales como planificación de cirugías y tratamientos de radioterapia, que pueden ser realizadas con mayor efectividad y menor riesgo [1], [2].

Habitualmente, la segmentación demanda tiempo y esfuerzo de preparación de los datos por parte de operadores especializados y esto impone limitaciones en la práctica médica. Incluso, en la actualidad, aún se suelen utilizar procesos de segmentación manuales a través de la selección cuidadosa de puntos sobre los cortes de la imagen, lo que constituye un trabajo lento, subjetivo y poco repetible [3].

Se han planteado diferentes métodos computacionales capaces de llevar a cabo la segmentación de una imagen; sin embargo, aún no existen soluciones definitivas ni algoritmos generalmente aplicables. Dentro de las distintas posibilidades, el *thresholding* [4] es uno de los algoritmos más simples y rudimentarios, propuesto inicialmente para imágenes planas, capaz de aislar objetos del fondo basándose en los valores de intensidad de los elementos de la imagen. No obstante, por sus características, esta técnica generalmente no se adapta a imágenes en las que existen zonas anatómicamente diferentes con intensidades similares, lo que frecuentemente ocurre en imágenes médicas. Basados en el concepto de *thresholding*, los algoritmos de *marching cubes* o *marching tetrahedra* [5] constituyen una opción de segmentación usada generalmente para el *rendering* de imágenes, que permiten la generación de iso-superficies a partir de un valor de umbral especificado. La idea básica es determinar si los elementos del volumen son atravesados por la superficie de nivel considerada, a partir de la evaluación del valor de los vértices de cada voxel con respecto al umbral. Estos esquemas suelen conducir a varias situaciones ambiguas que pueden generar huecos en la superficie resultante y además poseen las desventajas inherentes a los métodos basados en umbrales.

Por otro lado, los enfoques basados en la detección de bordes permiten la delimitación de fronteras entre componentes, caracterizadas por puntos de alto valor de gradiente [6]. Este tipo de métodos resultan apropiados para imágenes de características bien definidas, pero pueden producir resultados no satisfactorios debido a su sensibilidad al ruido o a la carencia de adaptabilidad local. Esto resulta decisivo en el caso de imágenes provenientes de reconstrucción tomográfica, en las que los diferentes tejidos no necesariamente poseen una intensidad uniforme y los bordes no siempre resultan claramente definidos a partir de las intensidades de los voxels.

Recientemente, se ha difundido otra opción de segmentación basada en contornos activos, conocidos como *snakes*, los cuales se basan en la evolución de una curva elástica ubicada sobre el objeto a segmentar [7]. Esta *snake* se modifica a partir de su forma y ubicación inicial como resultado de la acción de fuerzas internas y externas que, por un lado modelan su elasticidad y por otro tienden a acercarla a la imagen, respectivamente. Si bien se han reportado resultados satisfactorios de su aplicación, se trata de un esquema computacionalmente intensivo y que plantea algunos inconvenientes, según se describe en [8], principalmente cuando los objetos a segmentar poseen formas muy irregulares.

Los enfoques de segmentación basados en el crecimiento de regiones buscan delimitar zonas homogéneas de la imagen, mediante la inclusión progresiva de los elementos que cumplen determinado criterio de similitud [9], [10]. En este trabajo se propone un método de segmentación basado en un algoritmo de crecimiento de regiones, aplicado directamente sobre la imagen digital tridimensional conformada a partir de la integración de los sucesivos cortes transversales del objeto original. Además, se han establecido criterios que permiten controlar tal incorporación, de manera de prevenir el escape del llenado hacia regiones adyacentes. Por sus características, este esquema de segmentación tiene en cuenta las características locales de los elementos de la imagen y resulta bastante estable con respecto a los problemas en los que las técnicas mencionadas habitualmente fallan.

2. SEGMENTACIÓN POR CRECIMIENTO DE REGIONES

Una región consiste en una parte de la imagen que satisface un cierto criterio de uniformidad. Cada región comienza a formarse a partir de voxels considerado semillas y evoluciona, mediante un algoritmo recursivo, por incorporación de aquellos voxels vecinos a los ya integrados que satisfacen la condición establecida [9]. Comúnmente, los criterios utilizados se refieren a propiedades de proximidad y homogeneidad de los voxels considerados, considerándose que voxels cercanos con intensidades similares corresponden al mismo material. El proceso de segmentación finaliza cuando no se encuentran más voxels que cumplan la condición especificada.

Algoritmo de crecimiento

En general, la determinación del punto de partida para la segmentación suele ser un aspecto crítico del algoritmo. Se han propuesto varios métodos para esta etapa: automáticos, aleatorios [11], interactivos [12], etc. Un inconveniente de los métodos completamente automáticos es que pueden resultar imprácticos debido a la complejidad y variabilidad de las imágenes. La cantidad de información requerida para la determinación de las semillas suele ser muy grande y esto tiende a limitar severamente el alcance de su aplicabilidad. Por otro lado, la ubicación de los puntos iniciales sobre zonas ruidosas o correspondientes a bordes, en el caso de métodos interactivos o aleatorios, puede provocar una segmentación errónea de la imagen que finalmente requiere una etapa final de corrección.

Por este motivo, en este trabajo se consideró la posibilidad de que sea el usuario experto el encargado de especificar las semillas del proceso de segmentación. Para esto, se proveen facilidades de inspección de la imagen mediante un entorno de visualización, lo que permite determinar la posición de los puntos convenientes, sabiendo que ciertamente corresponden al material buscado. Entonces, el siguiente algoritmo de llenado se aplica a partir de cada uno de los voxels semilla y se propaga hacia sus vecinos.

```

ExtenderRegion( voxel v )
{
  si v es un voxel válido y no está visitado
  si v satisface el criterio de aceptación
    incluir v en la región (rotularlo como visitado-incluido)
    para cada vecino v' de v
      ExtenderRegion ( v' )
  sino
    descartar v (rotularlo como visitado-descartado)
  retornar
}

```

Fig. 1 Esquema general del algoritmo de segmentación por llenado

En cada paso, los voxels que son alcanzados por el algoritmo de llenado, se marcan como visitados para evitar su re-evaluación y aquellos que son incorporados a la región, se rotulan como incluidos. Por el contrario, los puntos que no cumplen con el criterio de aceptación, son marcados como descartados. Esta información se tiene en cuenta durante la etapa de visualización del volumen segmentado y generación de la geometría de las superficies detectadas.

Estrategia de incorporación de voxels

Los esquemas simples de incorporación de voxels [9] fácilmente pueden provocar desbordes de la región segmentada, debido a que el proceso recursivo se “escapa” por vías estrechas hacia otras zonas de la imagen que igualmente cumplen el criterio formulado, resultando en la inclusión de voxels que en realidad corresponden a un material diferente. Este problema se hace más evidente cuanto mayor es la tolerancia considerada, como puede apreciarse claramente en la Fig.2, en la cual se presenta el caso de segmentación de cerebro sobre una tomografía computada de cabeza conformada por 34 cortes.

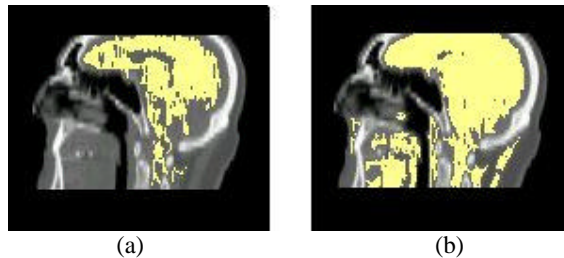


Fig. 2: Segmentación por llenado simple: (a) tolerancia 1, (b) tolerancia 3

Con el fin de restringir este tipo de situaciones que pueden conducir a una segmentación inapropiada, se consideró una estrategia más elaborada que no sólo tiene en cuenta la homogeneidad de cada voxel individual respecto de la semilla, sino también la relación con su entorno. Este esquema se complementa con un algoritmo de extensión de la frontera de la región por frente de avance, limitado por niveles.

Crecimiento según el entorno

El entorno de un voxel consiste en un conjunto de puntos que lo rodean, determinado en base a un parámetro r que indica el número de vecinos que deben ser considerados en cada dirección, según

los tres ejes cartesianos. Entonces, el criterio empleado para la incorporación de nuevos puntos consiste en verificar no sólo que la intensidad de cada voxel alcanzado por el proceso de llenado no difiera del valor de la semilla en más de un cierto umbral de tolerancia t , sino también que esta condición se satisfaga respecto del entorno.

El esquema propuesto contempla una evaluación porcentual de los puntos del entorno, en la cual no se exige que todos los puntos cumplan con el criterio establecido, sino que resulta suficiente que una proporción dada de ellos lo satisfaga para que el punto central sea anexado a la región. El porcentaje p de puntos satisfactorios puede ser establecido por el usuario al aplicar el algoritmo de llenado, junto con los demás parámetros. Esta estrategia brinda cierta flexibilidad, a la vez que reduce las oportunidades de escape del proceso de llenado por conductos delgados y el consecuente desborde hacia otras regiones de la imagen.

Crecimiento por "frente de avance" (push) por niveles

Luego de aplicar el algoritmo de segmentación, dependiendo de los parámetros elegidos, puede suceder que el llenado no alcance plenamente la frontera del material analizado. A tal fin, se ha implementado un algoritmo que trata de extender la región a partir de los límites alcanzados inicialmente, en caso de ser posible, mediante una operación denominada *push*.

El primer paso del algoritmo consiste en determinar los puntos que conforman la frontera actual de la región que se desea extender. Luego, para cada voxel en estas condiciones, la estrategia consiste en determinar un "frente de avance" en la dirección de cada vecino no incluido, compuesto por aquellos puntos que se encuentran sobre una placa de $f \times f$ voxels en torno al voxel en cuestión, sobre un plano ortogonal a la dirección de avance (Fig. 3).

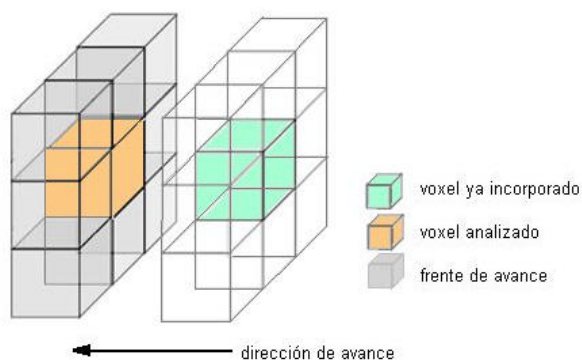


Fig.3: Esquema de *PUSH* por frente de avance

De manera similar al esquema de crecimiento presentado anteriormente, se evalúa la condición especificada por el usuario sobre los puntos que integran el frente, teniendo en cuenta el límite de tolerancia admitido t y el porcentaje p de puntos del entorno que deben cumplir el criterio de aceptación (estos dos parámetros pueden ser diferentes a los utilizados en la primera etapa de llenado). En caso de que la condición se satisfaga, se considera que el avance es válido por lo que el voxel central se agrega a la región y se aplica el mismo procedimiento a sus vecinos en forma recursiva, mientras existan voxels en condiciones de ser incorporados.

Para tratar de controlar el problema de escape del llenado hacia otras regiones, se ha establecido un crecimiento limitado por niveles, de manera que la recursión se aplica a los vecinos de cada voxel pero restringiendo el nivel de profundidad a explorar, de acuerdo a un valor n especificado.

Los resultados obtenidos por esta estrategia han sido satisfactorios. Por ejemplo, en la Fig. 4 se presentan tres cortes axiales diferentes de una resonancia magnética de cabeza conformada por 85 capas, en la cual se ha aplicado el método descrito para llevar a cabo la segmentación de masa encefálica. Inicialmente, se aplicó un llenado mediante evaluación porcentual del entorno con $t=25$, $r=2$ y $p=0.7$; el cual se completó con una operación *push* según un frente de avance de 5×5 voxels, con los parámetros anteriores y $n=2$. Como puede apreciarse, el llenado de la región ha logrado alcanzar casi completamente las fronteras del material.

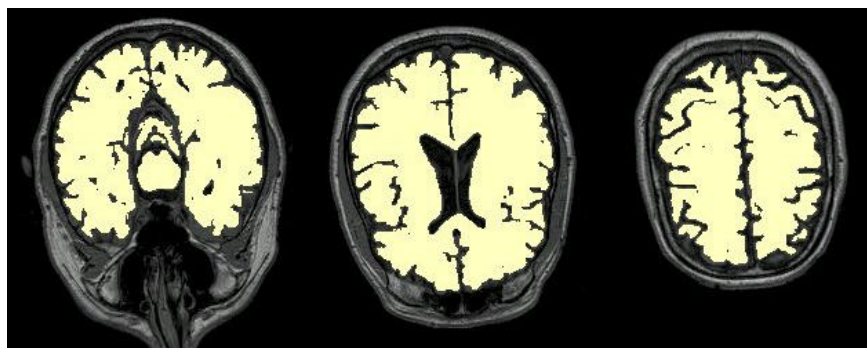


Fig. 4: Llenado porcentual sobre MRI de cabeza, con *push* por niveles

Tratamiento de puntos aislados

Ocasionalmente, el proceso de llenado pueden provocar que algunos voxels del interior de las regiones segmentadas no resulten incorporados, aún cuando se puede observar que naturalmente corresponden a la región en cuestión (Fig.5-a). Esto se debe a que, por las características del algoritmo de segmentación, la presencia de un punto cuya intensidad varíe demasiado con respecto a su entorno (punto aislado) puede hacer fallar la condición de aceptación y afectar también la inclusión de sus vecinos. Esto resulta aún más evidente cuanto mayor es el valor de p , ya que la presencia de un voxel con diferencia de intensidad mayor que el umbral de tolerancia considerado, puede producir el rechazo de todos los voxels del entorno.

Con el fin de tratar este inconveniente, se incorporó un filtro de pintado de puntos aislados, el cual se puede aplicar a la imagen generada por el proceso de segmentación con el fin de mejorar los resultados obtenidos. Este algoritmo efectúa un recorrido sobre la imagen para detectar la presencia de voxels que fueron rechazados por el algoritmo de llenado y que se encuentran rodeados de puntos que sí han sido incluidos.

Igualmente que en los casos de llenado, se considera un parámetro p referido a la proporción de voxels vecinos a verificar y además la distancia d a la que éstos se encuentran con respecto al voxel analizado. Entonces, en caso de que el porcentaje de vecinos, a distancia d , incorporados a una misma región sea igual o superior a p , el punto central se integra también a la región. De esta manera, no sólo es posible pintar voxels aislados sino también grupos de puntos que han quedado excluidos de las regiones segmentadas, de acuerdo al valor d considerado.

La Fig.5-b presenta el resultado de aplicar un pintado porcentual de puntos aislados, considerando $d= 2$ y $p= 0.7$, a partir de la imagen segmentada por llenado porcentual de regiones. Como puede apreciarse, los puntos que se encontraban aislados han sido integrados al volumen segmentado.

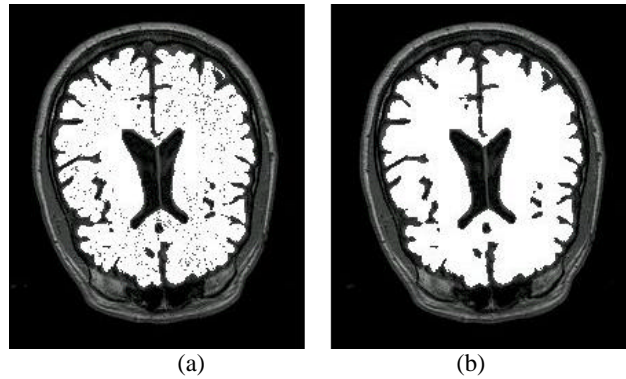


Fig.5: Tratamiento de puntos aislados: (a) segmentación por llenado, (b) resultado del pintado de puntos aislados

Este post-procesamiento es de valiosa ayuda para el proceso de generación de la geometría de las superficies detectadas, el cual se describe a continuación, ya que tiende a mejorar la calidad de las imágenes y reducir el costo de su visualización al disminuir la cantidad de triángulos obtenidos.

3. GENERACIÓN DEL MODELO DE SUPERFICIE

La herramienta desarrollada permite observar los resultados del proceso de segmentación mediante un módulo visualizador en el cual se pueden inspeccionar los diferentes cortes sobre los ejes cartesianos y especificar proyecciones arbitrarias del volumen a través de un algoritmo de *volume rendering* [13]. Además de estas facilidades, se ha incluido la posibilidad de generar los datos necesarios para la visualización de las superficies de las regiones segmentadas dentro del volumen, mediante opciones de *surface rendering* [14].

Una de las representaciones comúnmente utilizadas para la descripción de superficies consiste en la determinación de una malla de polígonos, generalmente triángulos. El algoritmo implementado para su generación se compone fundamentalmente de dos etapas. En un primer paso, se determina una triangulación inicial a partir de la información sobre los voxels que conforman la frontera de cada región. Posteriormente, esta triangulación se procesa, mediante filtros convenientes, con el objeto de suavizarla y remover imperfecciones.

Triangulación inicial

Cada región segmentada provee un marco de referencia para la generación de la descripción poligonal de su superficie. Mediante un recorrido a través de la imagen tridimensional, se obtienen las caras de los voxels que corresponden al contorno de la región de interés; es decir, aquellas que se encuentran entre un voxel que ha sido incluido en la región y otro que no pertenece a la misma. A partir de estas caras “frontera” se puede generar una triangulación inicial, transformando cada una de ellas en dos triángulos, a partir del trazado de una diagonal.

Cada esquina de un voxel constituye un vértice de la malla regular en que se convierte la imagen y entonces cada triángulo se puede describir por la tripla de vértices en el espacio que lo determinan. Para obtener superficies cerradas y coherentes, se debe tener en cuenta que el vector normal a cada triángulo debe apuntar hacia el interior de la región segmentada. Por lo tanto, si se considera un

punto encerrado en la superficie a visualizar, el sentido de los vértices de todos los triángulos debe ser antihorario.

Suavizado de superficies

Si bien el dominio generado a partir del conjunto de triángulos que describe la superficie de cada región segmentada corresponde al conjunto de celdas en estudio, la imagen resultante puede ofrecer un aspecto escalonado que entorpece la visualización, debido a que los triángulos son generados a partir de las caras de los voxels (Fig.6-a). Una manera de solucionar este inconveniente es aplicar un procedimiento de suavizado, de manera de lograr superficies más semejantes a las formas originales de los materiales especificados (Fig.6-b).

El suavizado se realiza mediante la aplicación de un filtro promedio. Para ello, se consideran –para cada vértice de la malla generada que se desea suavizar– todos los triángulos que poseen un vértice coincidente con él y se calcula su nueva posición como el promedio de las posiciones de los demás vértices de los triángulos involucrados. El proceso se puede aplicar en forma iterativa, hasta obtener los resultados deseados. Al tratar con superficies de forma convexa, este procesamiento puede provocar una reducción del volumen del cuerpo, ya que los puntos tenderán a moverse hacia el interior de la región, siendo mayor el efecto según la cantidad de iteraciones efectuadas. Por lo tanto, debe ser aplicado moderadamente y sólo a efectos de visualización.

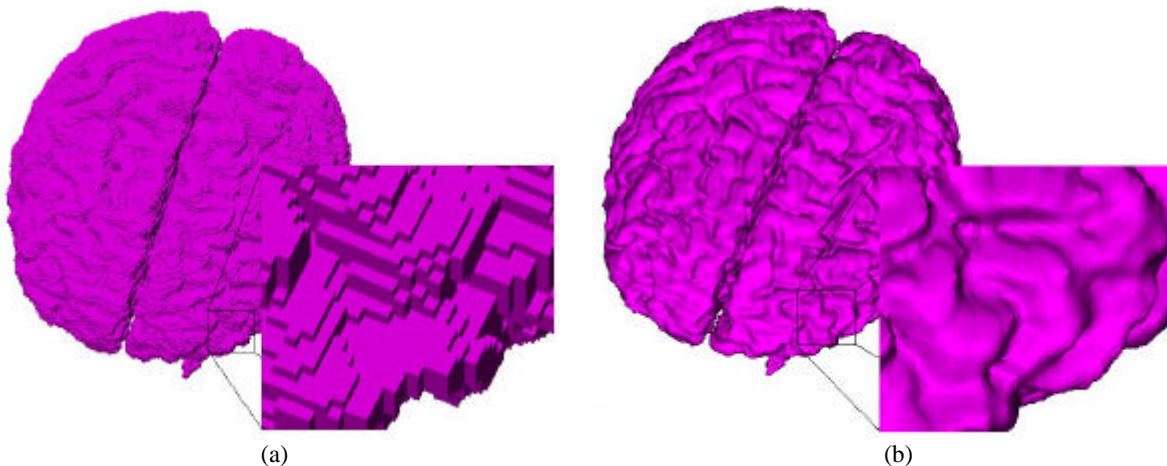


Fig.6: Descripción poligonal de superficies: (a) triangulación inicial – (b) superficie suavizada

Detección de situaciones conflictivas

Ocasionalmente, la triangulación obtenida puede poseer problemas en determinados conjuntos de triángulos, lo que perjudica la visualización, ya que el algoritmo anterior no logra suavizar efectivamente la superficie correspondiente.

Estos problemas se originan cuando dos voxels segmentados están unidos sólo por una arista común a ambos. Normalmente, en la descripción poligonal de una superficie, si se produce una situación como ésta, los vértices v_1 y v_2 mostrados en la Fig.7 aparecen sólo una vez en la lista de vértices, asociados a la ubicación que cada uno de ellos posee en el espacio considerado. Entonces, el algoritmo de suavizado no logra su efecto ya que los triángulos no pueden ser despegados.

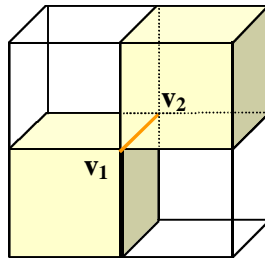


Fig.7: Situación de arista conflictiva

La solución adoptada incluye una etapa de detección de aristas conflictivas durante el proceso de generación de la descripción poligonal de la superficie, tratando de determinar si se produce uno o más de los casos posibles, mientras se realiza la evaluación para cada uno de los voxels segmentados por el algoritmo de llenado.

Sin embargo, cuando un voxel posee varias aristas conflictivas, el algoritmo de suavizado tampoco logra completamente su objetivo a pesar de que se reenumeren los vértices involucrados. Esto puede apreciarse en la Fig.8-b, en la que se presenta la superficie suavizada a partir de la triangulación inicial, cuyo detalle se muestra en la Fig.8-a, correspondiente a la componente piel de la tomografía de 34 capas anteriormente nombrada.

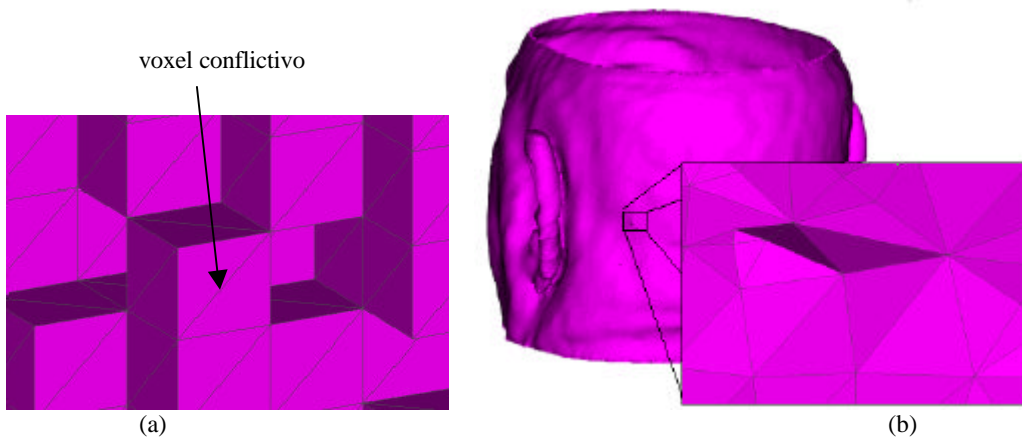


Fig.8: Tratamiento de voxels con aristas conflictivas: (a) voxel con 3 aristas conflictivas, (b) detalle de la superficie suavizada

Por este motivo, el algoritmo de generación de la superficie incluye además la opción de remover los voxels que poseen más de una cierta cantidad de aristas conflictivas. Este procesamiento de la superficie se efectúa mediante un ciclo que se encarga de contar la cantidad de aristas conflictivas para cada voxel, seguido de un paso de eliminación de los voxels que poseen el número mayor de aristas encontradas durante la iteración actual. Esto se realiza hasta que la mayor cantidad de aristas conflictivas para un voxel coincida con la especificada inicialmente, que por defecto es 1. Entonces, el paso de depuración se detiene.

En caso de existir aún alguna situación problemática, se registra cuáles son las aristas conflictivas y con qué voxel vecino se asocia cada una. Entonces, al considerar la inclusión de un triángulo para ese voxel, se comprueba si éste posee alguna arista señalada como conflictiva. Si ocurre esta situación, sus vértices son reenumerados. De esta manera, puede suceder que para una misma ubicación en el espacio, se especifique más de un vértice asociado, en el archivo que describe la superficie. El mecanismo propuesto permite que, posteriormente, al aplicar el algoritmo de

suavizado, éste trate a estos puntos como vértices diferentes y pueda despegarlos. Así, por ejemplo, en la Fig.9-a se muestra el resultado del suavizado sobre la triangulación inicial sin procesamiento de aristas, mientras que en la Fig.9-b, se aplicó el esquema de detección de aristas conflictivas y eliminación de voxels con dos o más aristas conflictivas, previo a la generación de la triangulación. El resto de los vértices correspondientes a aristas que aún son detectadas como conflictivas se han reenumerado, logrando que sean despegadas al aplicar el algoritmo de suavizado.

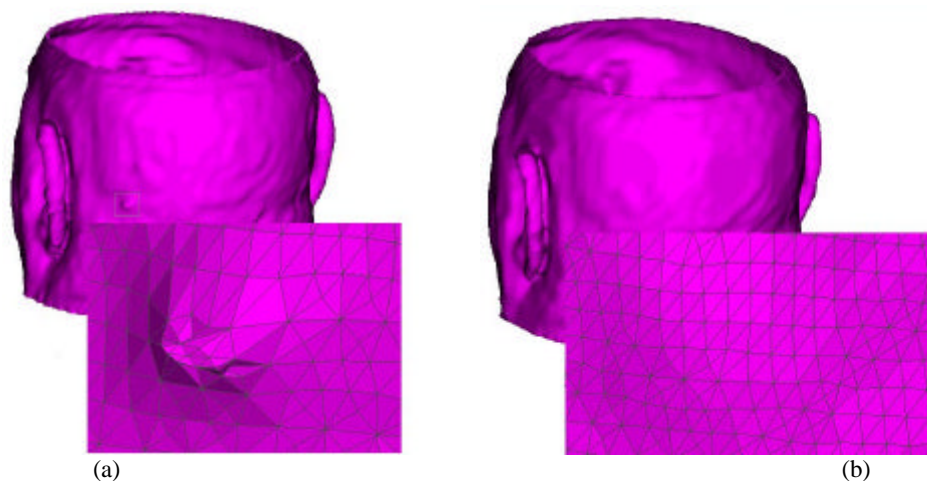


Fig.9: Comparación de superficies: (a) generación sin procesamiento de aristas conflictivas, (b) remoción de voxels con 2 o más aristas conflictivas y despegado del resto

Es interesante observar que la superficie generada mediante el algoritmo propuesto en este trabajo posee muy buena calidad, la cual supera a las obtenidas mediante otros algoritmos. Además, el método empleado es simple, robusto y de un costo computacional relativamente bajo. En la Fig.10 se presentan otras visualizaciones obtenidas por *surface rendering* de las triangulaciones generadas a partir del esquema de segmentación propuesto, postprocesadas con el filtro de suavizado. En este caso, las imágenes muestran visualizaciones de la segmentación del cerebro (Fig.10-a) a partir de la resonancia magnética, y del cráneo (Fig.10-b) y piel (Fig.10-c) correspondientes a la tomografía computada, anteriormente mencionadas.

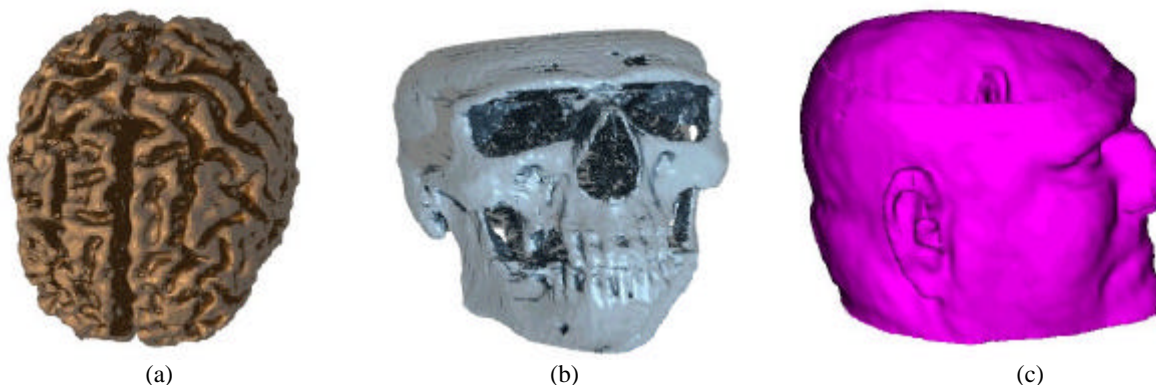


Fig.10: Visualización de superficies generadas mediante el algoritmo de llenado con suavizado: (a) cerebro de MRI, (b) cráneo y (c) piel de tomografía computada

4. CONCLUSIONES

En este trabajo se ha presentado un método simple y eficiente que permite segmentar imágenes digitales tridimensionales por crecimiento de regiones a partir de una o más semillas. Se han propuesto además soluciones para prevenir ciertos problemas del esquema utilizado, que mejoran los resultados de la segmentación. También se ha planteado el procedimiento utilizado para la descripción de las superficies por triangulación, incluyendo un novedoso algoritmo de tratamiento de aristas conflictivas. Las superficies generadas resultan de calidad superior a las obtenidas con otros métodos y con un costo computacional inferior.

5. REFERENCIAS

- [1] J. Duncan, N. Ayache, *Medical Image Analysis: Progress over Two Decades and Challenges Ahead*, IEEE Transactions on Pattern Recognition and Machine Intelligence, Vol.22, No.1, 2000
- [2] D. Fernandez, B. Penesi, F. Ribeiro, M. del Fresno, M. Vénere, *Planificación Automatizada de Tratamientos de Radioterapia*, Proceedings CACIC'2000, Ushuaia, Argentina, octubre 2000
- [3] S. Hojjatoleslami, F. Kruggel, *Segmentation of Large Brain Lesions*, IEEE Transactions on Medical Imaging, Vol.20, No.7, July 2001
- [4] R. Gonzalez, R. Woods, *Digital Image Processing*, Addison-Wesley, 1992
- [5] W.Lorensen, H. Cline, *Marching Cubes a high resolution 3D surface reconstruction algorithm*, Computer Graphics, Vol 21, No.4, pp 163-169, 1987
- [6] D. Ney, E. Fishman, D. Magid, R. Debrin, *Volumetric rendering of computed tomography data: principles and techniques*, IEEE Computer Graphics and Applications, 1990, Vol. 10, No. 2
- [7] N. Davison, H. Eviatar, R. Somorjai, *Snakes simplified*, Pattern Recognition 33, pp.1651-1664, 2000.
- [8] L. Ji, H. Yan, *Loop-free snakes for highly irregular object shapes*, Pattern Recognition Letters 23, pp 579-591, 2002
- [9] K. Castleman, *Digital Image Processing*, Prentice Hall, 1996
- [10] X. Pardo, M. Carreira, A. Mosquera, D. Cabello, *A Snake for CT image segmentation integrating region and edge information*, Image and Vision Computing 19, 2001
- [11] D.Zhu, A.Yuille, *Region Competition: Unifying Snakes, Region Growing, and Bayes/MDL for Multiband Image Segmentation*, IEEE Trans.Pattern Analysis and Machine Intelligence, vol.18, no.9, pp. 884-900, 1996
- [12] R.Adams, L.Bischof, *Seeded Region Growing*, IEEE Trans.Pattern Analysis and Machine Intelligence, vol.16, pp. 641-647, 1994
- [13] J. Foley, A. Van Dam, S. Feiner, J. Hughes, *Introduction to Computer Graphics*, Addison-Wesley, 1996
- [14] A. Watt, *3D Computer Graphics*, Addison-Wesley, 2nd. ed., 1993