

# Mejora en el algoritmo de construcción de esqueletos 3D y algoritmos de adelgazamiento de eje medio.

Manuel G. Forero, Camilo A. Murillo

[manuel.forero@unibague.edu.co](mailto:manuel.forero@unibague.edu.co), [camiloandresmurilloruiz@gmail.com](mailto:camiloandresmurilloruiz@gmail.com)

Semillero Lún grupo D+TEC

Universidad de Ibagué

Ibagué, Colombia

*Palabras clave— Esqueleto, esqueletonización, adelgazamiento, Topología matemática.*

**Resumen—** La esqueletonización es una de las técnicas más utilizadas en reconocimiento y seguimiento de objetos; entre ellas una de las técnicas conocidas y utilizadas es la desarrollada por Lee et al. Sin embargo, en este artículo se muestra que ésta no siempre produce esqueletos completos, encontrando las causas del error, en particular cuando el ancho del objeto es par un caso de las características de Euler permite la eliminación de los vóxeles erróneamente. Se presenta una solución que permite obtener esqueletos completos en todas las situaciones, basado en la modificación del barrido de los objetos, presentando resultados comparados entre la técnica original y la mejorada.

*Key words— Skeleton, skeletonization, thinning, mathematical topology.*

**Abstract—** Skeletonization is one of the most used techniques in object recognition and tracking; being the one developed by Lee et al. one of the most popular and used. However, this article shows that it does not always produce complete skeletons, finding the causes of error, particularly when the width of the object is even a case of Euler characteristics allows elimination of voxels incorrectly. A solution that allows complete skeletons in all situations, based on the modification of the scanning of objects, presenting comparison between the original and improved technical results is presented.

## 1. INTRODUCCIÓN

Uno de los métodos más utilizados en el área de visión artificial para el reconocimiento de objetos es la esqueletonización. Su objetivo es obtener un patrón continuo que representa el objeto original con la menor cantidad de datos posibles. El esqueleto es la representación más pequeña de un objeto mediante un conjunto de líneas con un ancho de un píxel (2D) o un voxel (3D). Para llevar a cabo este proceso existen diferentes métodos. En el caso 2D uno de los métodos más utilizados consiste en la eliminación iterativa de los píxeles del objeto que tienen contacto con el fondo

(bordes) hasta que encuentre el eje central del objeto. Este debe ser un píxel de espesor. En el caso 3D, hay dos tipos de esqueletos: El primero dado por el eje central del objeto, constituido por el número mínimo de vóxeles continuos situados a distancia equidistante de los bordes del objeto y el segundo definido por la superficie media del objeto, la cual incluye el eje central.

Otros métodos más modernos han sido desarrollados como los presentados por Palagyi pero debido a la complejidad en los cálculos que se proponen en la técnica estas no son muy utilizadas.

La esqueletonización se utiliza en diferentes aplicaciones como la animación por ordenador, visión y el reconocimiento de patrones, la robótica, la navegación y la medicina. La esqueletonización 3D es una técnica compleja que requiere un alto número de operaciones, lo que hace que sea computacionalmente lento y difícil de procesar. Algunos métodos se basan en la transformada de distancia 3D, que tiene una alta carga computacional, mientras que otros se basan en el adelgazamiento. Entre estas últimas se encuentran las técnicas desarrolladas por Lee et al. la cual es, a pesar de su antigüedad una de las más populares debido a la baja complejidad que presenta en comparación con otras técnicas, al dividir el volumen conformado por el voxel en estudio y sus vecinos (26-vecinos alrededor de cada voxel que está siendo considerado ser eliminado) en octantes, haciéndolo más eficiente [1]. Esta técnica se ha desarrollado para ITK (Insight Segmentation and Registration Toolkit) [2], ImageJ [3] y Matlab [4] con el fin de obtener el eje medial 3D. Un ejemplo se muestra en la Fig. 1. Palagyi et al. [5] desarrollaron también varias técnicas basadas en el adelgazamiento del objeto, pero son más pesadas computacionalmente.



Fig. 1. Ejemplo de esqueletonización.

A pesar de su popularidad, hemos encontrado que el método de Lee et al falla al no hallar esqueletos completos. En algunos casos, tal como se ilustra en el ejemplos de las Fig. 6). En este artículo se describen las causas del error y la solución al problema. También se incluye una optimización que permite hacer más eficiente el método.

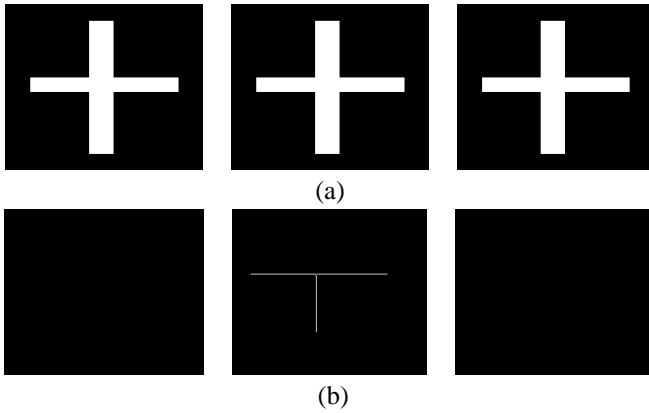


Fig. 2. (a) Imagen 3D compuesta de tres imágenes sintéticas (b) Esqueleto obtenido al aplicar el método de Lee.

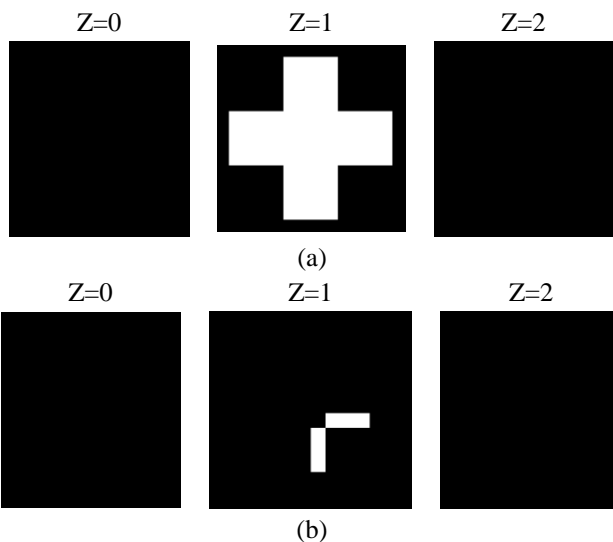


Fig. 3. (a) Imagen 3D compuesta por tres imágenes sintéticas iguales donde Z representa la coordenada de

profundidad. (b) Esqueleto obtenido al aplicar el método de Lee.

El método barre la imagen 3D en 6 direcciones (ver Fig. 4) y elimina de forma repetitiva los vóxeles que satisfacen las siguientes condiciones:

- I. Se trata de un simple punto, es decir, si se elimina el voxel el número de objetos en el interior del volumen debe permanecer constante. Este término se ha utilizado en las operaciones de adelgazamiento para preservar propiedades topológicas como el número de objetos conectados, cavidades y agujeros [7, 8].
- II. La característica de Euler se mantiene constante después de la eliminación del voxel.
- III. El número de vecinos es mayor o igual a uno.

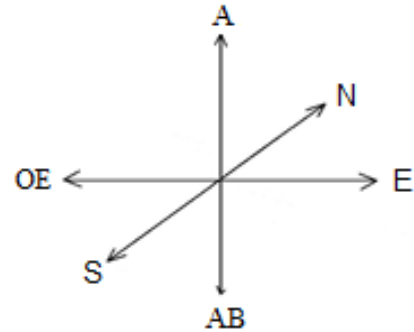


Fig. 4. Direcciones de barrido. Cada letra representa la Norte, Sur, Este, Oeste, Arriba y Abajo.

## 2. METODOLOGÍA

No se ha encontrado en la literatura ninguna mención respecto a los errores producidos por esta técnica. Por lo tanto, se hizo necesario hacer un estudio profundo de este método a fin de determinar en qué condiciones se producen esqueletos erróneos.

Para la evaluación de la técnica se crearon 132 imágenes sintéticas en 3D recreando diferentes condiciones. En particular, imágenes que producen esqueletos erróneos como se ilustra en la Fig. 2 y la Fig. 3.

Para disminuir el tiempo de procesamiento utilizado por el método, Lee et al. subdividen el volumen de compuesto por el voxel en estudio y sus 26-vecinos en 8 octantes superpuestos que comparten siempre el voxel en estudio. Para analizar si un voxel puede ser eliminado,

los autores representan cada octante  $2 \times 2 \times 2$  como un byte binario, donde cada bit representa un voxel del octante. Así, un bit es cero o uno dependiendo de si el voxel correspondiente es negro o blanco. La Fig. 5 (a) muestra los pesos dados a cada voxel. Dado que existen 256 posibles combinaciones de valores de los vóxeles y la dificultad para calcular la característica de Euler, Lee et al. construyeron una tabla con los 256 valores pre calculados de las características de Euler. Sin embargo,

dado que un voxel debe ser considerado para su eliminación únicamente cuando pertenece a un objeto, que aquí se asume de color blanco, el análisis puede simplificarse al considerar solamente los 7 vóxeles del octante restante, cuando el voxel de estudio es blanco. Así, la tabla se reduce a 128 valores y el número de operaciones para la construcción de la palabra binaria representando cada voxel se reduce de  $O(8n)$  a  $O(7n)$  (ver Fig. 5 (b)).

Por lo tanto, ahora se emplea una tabla de 128 valores (Ver Tabla 1).

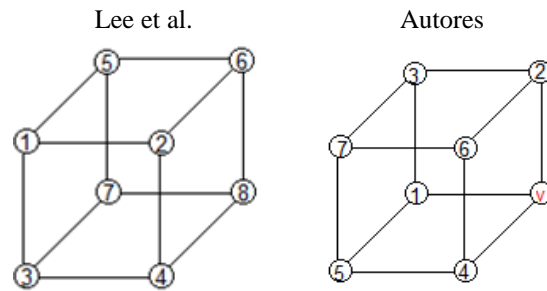


Fig. 5. Orden de los vóxeles. (a) Lee et al. (b) Autores.

**Table 1. Tabla de preservación de las características de Euler.**

$n$	$8\delta G_{26}$	$n$	$8\delta G_{26}$	$n$	$8\delta G_{26}$	$n$	$8\delta G_{26}$	$n$	$8\delta G_{26}$
0	1	26	3	52	3	78	1	104	-1
1	-7	27	3	53	3	79	1	105	-1
2	-3	28	3	54	3	80	-1	106	-1
3	-3	29	3	55	3	81	-1	107	-1
4	-3	30	3	56	1	82	3	108	-1
5	-3	31	3	57	1	83	3	109	-1
6	1	32	-1	58	1	84	-1	110	-1
7	1	33	-1	59	1	85	-1	111	-1
8	-1	34	-1	60	1	86	3	112	1
9	-1	35	-1	61	1	87	3	113	1
10	-1	36	3	62	1	88	1	114	1
11	-1	37	3	63	1	89	1	115	1
12	-1	38	3	64	-1	90	1	116	1
13	-1	39	3	65	-1	91	1	117	1
14	-1	40	1	66	3	92	1	118	1
15	-1	41	1	67	3	93	1	119	1
16	-3	42	1	68	-1	94	1	120	-1
17	-3	43	1	69	-1	95	1	121	-1
18	1	44	1	70	3	96	1	122	-1
19	1	45	1	71	3	97	1	123	-1
20	1	46	1	72	1	98	1	124	-1
21	1	47	1	73	1	99	1	125	-1
22	5	48	-1	74	1	100	1	126	-1
23	5	49	-1	75	1	101	1	127	-1
24	3	50	-1	76	1	102	1		
25	3	51	-1	77	1	103	1		

Se encontró que el método de Lee falla cuando el ancho inicial del objeto en una dirección de barrido es par como se ilustra en la Fig. 3 y 4. El error no es fácil de verse, ya que un objeto tiene diferentes anchos a lo largo de su cuerpo.

Para entender cómo se produce el problema, se emplea un objeto 2D en un espacio 3D para describir cómo el algoritmo de la Lee obtiene el esqueleto. Este objeto es un rectángulo de tamaño 4x12x1 orientado norte-sur en el plano X con capas superior e inferior negras como se muestra en la figura. 6 (a).

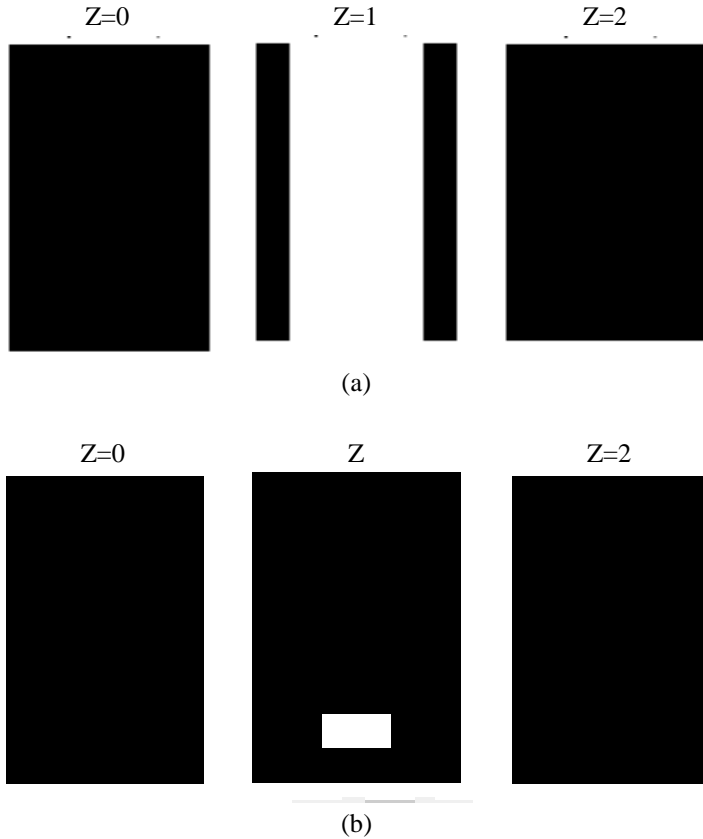


Fig. 6. Imagen sintética creada para el análisis. (a) Imagen original y (b) Resultado del método.

La Fig. 6 (b) muestra el resultado al aplicar el método de Lee, donde el esqueleto consta de sólo dos vóxeles, situados en la parte inferior del rectángulo original. Este es un esqueleto erróneo dado que el esqueleto esperado es una línea de una anchura de un pixel que va de arriba a abajo. El resultado se obtiene de la siguiente forma. En primer lugar, el objeto es barrido como se ilustra en la Fig. 7, donde los números 1, 2, 3 y 4 describen la secuencia del barrido. Inicialmente los puntos fronterizos de tipo Norte se evalúan, luego los tipo Sur, Oeste y Este, respectivamente para cada imagen. Todo el conjunto de vóxeles candidatos son eliminados, dado que cumplen las tres condiciones establecidas, anteriormente nombradas. Entonces, la imagen 3D se barre de nuevo, pero ahora desde la parte superior a la inferior, y en este caso la segunda imagen de la pila es barrida como se muestra en la Fig. 8. Ahora, el

primer voxel blanco (a), se suprime ya que satisface las tres condiciones. El siguiente voxel (b) también se elimina, ya que cumple con las tres condiciones. Los vóxeles restantes también se eliminarán de la misma forma. Sólo los dos últimos vóxeles (c, d) se conservan, sólo porque el número de vecinos es inferior a dos (Condición 4 en Lee et al.).

En el caso ideal, el voxel b no debe mantenerse. Sin embargo, debido a que es un solo voxel, el número de vecinos es mayor que uno y en particular la característica de Euler no cambia, como se muestra es la figura. 10, es decir, antes y después de la eliminación de un candidato voxel la característica de Euler permanecen constantes, y por lo tanto se elimina. Esta situación corresponde al caso 12 en la Tabla 1 (equivalente al caso 49 en la tabla de Lee). Por lo tanto, se retira el voxel cuando debería ser retenido. Este caso no aparece cuando el ancho del objeto es impar. La única solución para evitar que se presente el caso 12 cuando el objeto es par, es modificar la forma del barrido de manera que un objeto no sea barrido de forma consecutiva en alguna de sus caras.

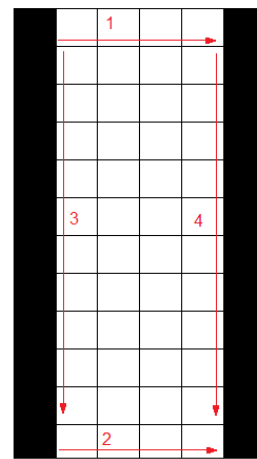


Fig. 7. Barrido de los bordes de la imagen.

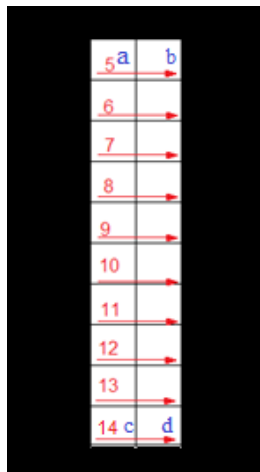


Fig. 8. Segundo barrido de la imagen.



Fig. 9. Resultado final del método de Lee

Así, en la fig. 9 el problema aparece cuando la imagen fue barrido fila por fila. Para resolver el problema, los vóxeles a ser considerados como candidatos para la eliminación deben ser barridos en las esquinas del objeto, de modo que los vóxeles seleccionados como candidatos para la eliminación son los que tienen dos planos diferentes simultáneamente llamados por nosotros bordes fronterizos. Por lo tanto, en lugar de los seis barridos original en los puntos fronterizos de adyacencia, se consideran 24 barridos dados por NE, NW, NU, NB, SE, SW, SU, SB, EN, ES, EU, EB, WN, WS, WU, WB, UN, US, UE, UW, BN, BS, BE, BW.

Así, la imagen eliminara continuamente desde el exterior hacia el centro del objeto.

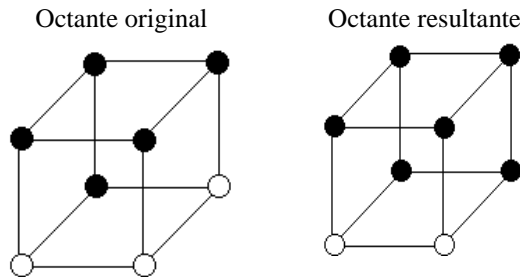
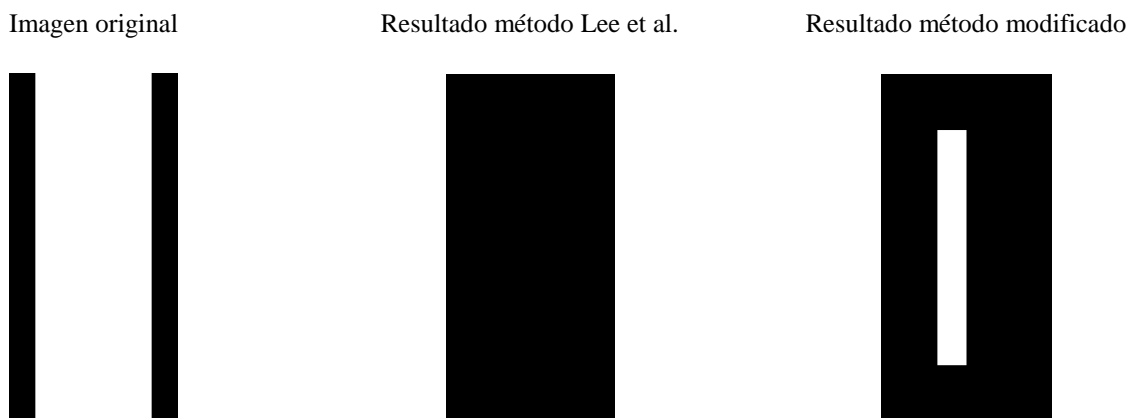


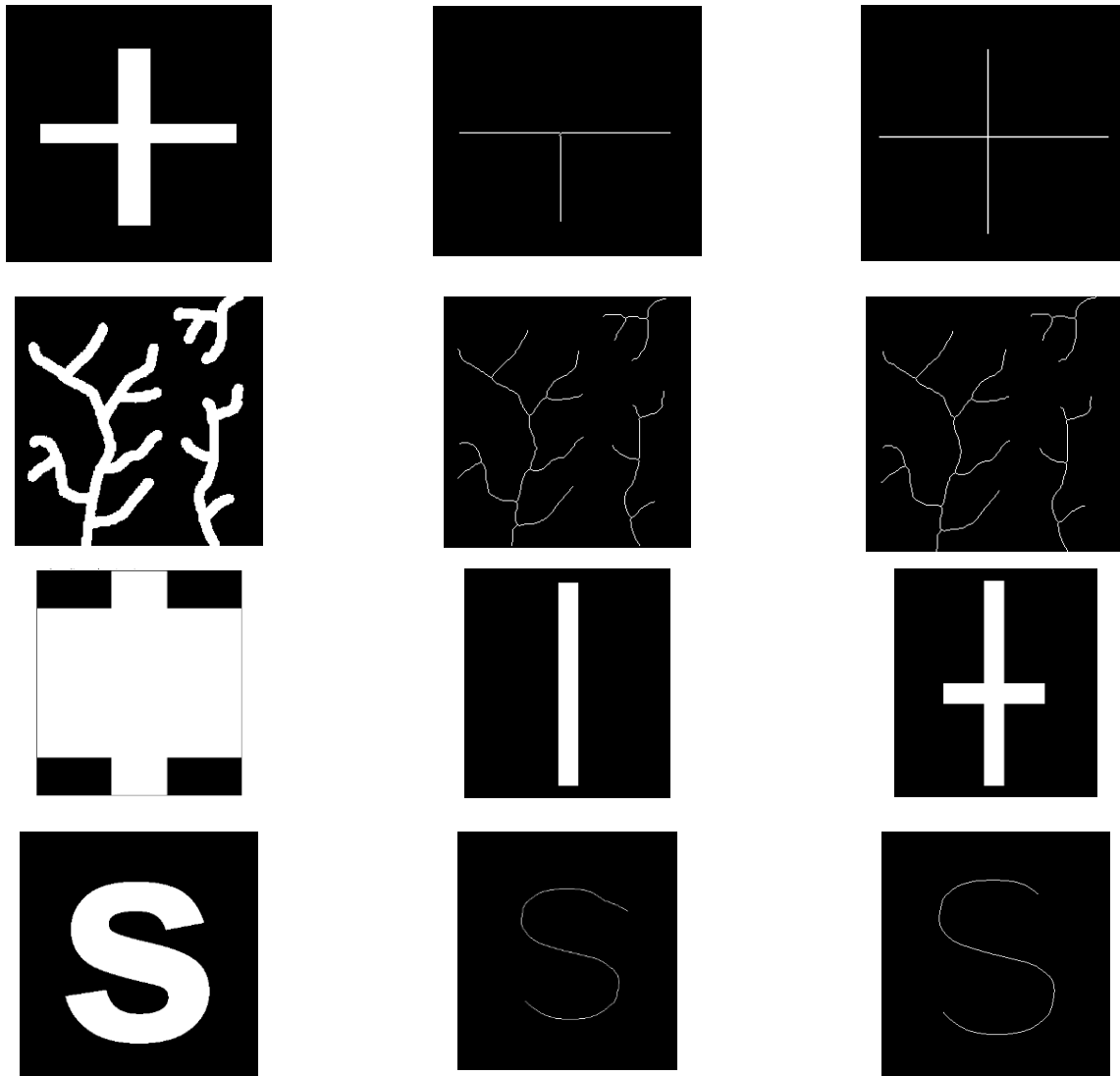
Fig. 10. Caso 12 que produce errores en los esqueletos resultantes.

### 3. RESULTADOS

La tabla 2 muestra los resultados obtenidos con algunas imágenes sintéticas, utilizando el método de Lee et al y el modificado por los autores. Todo el conjunto de imágenes se compone de tres capas idénticas.

**Tabla 2. Resultados.**





Como se puede ver el método modificado produce esqueletos más precisos, se soluciona el problema encontrado en el método original y se mantienen los mismos esqueletos cuando el método original funcionaba de manera correcta.

#### 4. CONCLUSIONES

Las causas de los errores producidos por las irregularidades en los esqueletos obtenidos con el método de Lee et al. se explicaron. Se encontró que a pesar de las condiciones básicas del método son correctas, producen errores debido a la forma en que las imágenes son barridas. Se presenta una solución, que permite resolver el problema. Se expone una nueva tabla de las características de Euler en donde se reduce el tamaño de la tabla original 256-128 valores, lo que permite reducir el tiempo de procesamiento. Además se propone un nuevo barrido de las imágenes para poder obtener esqueletos precisos de imágenes de tal forma que sean invariantes a la rotación del objeto 3D.

#### 5. REFERENCIAS

- [1] Lee T. C., R. L. Kashyap, "Building skeletons models via 3D medial surface/axis thinning algorithms", Knowledge based laboratory, Vol 56, pp. 462-478, 1994.
- [2] Implementación del método de Lee et. al. para ITK, por Hanno Homann, <http://www.insight-journal.org/browse/publication/181>.
- [3] Implementación del método de Lee et. al. para ImageJ, por Ignacio Arganda, [http://fiji.sc/Skeletonize3D#General\\_Description](http://fiji.sc/Skeletonize3D#General_Description).
- [4] Implementación del método de Lee et. al. para Matlab, por Philip Kollmannsberger, <http://www.mathworks.com/matlabcentral/fileexchange/43400-skeleton3d>.
- [5] Palágyi K., "Topology Preserving Parallel 3D", Institute of Informatics, University of Szeged, Szeged, Hungary, 2012.
- [6] Skeletonize 3D [Online]. Disponible: (<http://imagejdocu.tudor.lu/doku.php?id=plugin:morphology:skeletonize3d:start>), Nov 2011, Oct 2014.

[7] D. G. Morgenthaler, "Three Dimensional Simple Points: Serial Erosion, Parallel Thinning, and Skeletonization", Technical Report TR-1005, Computer Vision Laboratory, Computer Science Center, University of Maryland, 1981.

[8] S. Lobregt, W. Verbeek, and F. C. A. Groen, "Three dimensional skeletonization: Principle and algorithm", IEEE Trans, Pattern Anal. March Intell. 2, 75-77, 1989.

[9] D. G. Morgenthaler, "Three Dimensional Digital Topology: The genus, Technical Report" TR-980, Computer Vision Laboratory, Computer Science Center, University of Maryland, 1980.

[10] T. Y. Kong and A. Rosenfeld, Digital topology: Introduction and survey, Comput. Vision Graphics Image Process. 48(3). 357-393, 1989.