

# Sistema de Visión Estereoscópico para el guiado de un Robot Quirúrgico en Operaciones de Cirugía Laparoscópica HALS

Carlos Castedo Hernández\*, Rafael Estop Remacha\*, Eusebio de la Fuente\*\*, Lidia Santos\*\*

\*Escuela Ingenierías Industriales Univ. Valladolid. Paseo Cauce 59. 47005 Valladolid,  
email: {carlos.castedo, rafael.estop}@alumnos.uva.es

\*\*ITAP Univ. Valladolid. Paseo Cauce 59. 47005 Valladolid,  
email: efuente@eii.uva.es, lidia.santos@uva.es

## Resumen

*En este artículo se presenta un sistema de visión estereoscópico destinado al guiado de un robot colaborativo en operaciones de cirugía. El objetivo del sistema de visión es que el robot quirúrgico tenga en todo momento localizada la mano del cirujano para no colisionar con ella y para cooperar en la operación si es necesario. El sistema de visión va ubicado en el interior del abdomen del paciente, ya que se ha enfocado a operaciones de cirugía laparoscópica asistida con la mano.*

**Palabras Clave:** Cirugía Mínimamente Invasiva, estereovisión, robótica quirúrgica.

## 1 INTRODUCCIÓN

A partir de los años noventa, con el enorme desarrollo en la tecnología del video y la transmisión de imágenes, se comienza a desarrollar la cirugía laparoscópica o mínimamente invasiva entre las diferentes especialidades quirúrgicas que comparten el tratamiento del abdomen.

La cirugía laparoscópica es una técnica que usa varias incisiones, llamadas puertos, de 0.5 a 1 cm. En cada puerto se inserta un instrumento tubular conocido como trócar. Durante el procedimiento, a través de los trocares, se pasan instrumentos especializados y una cámara especial llamada laparoscopio. Al iniciar el procedimiento, el abdomen se infla con gas, dióxido de carbono, para proporcionar al cirujano un espacio de trabajo y visibilidad. El laparoscopio transmite imágenes de la cavidad abdominal a los monitores de video de alta resolución del quirófano.

En ciertos casos, el cirujano puede elegir usar un tipo de puerto especial que es lo suficientemente amplio como para insertar una mano. Cuando se usa un puerto para la mano, la técnica quirúrgica se llama laparoscopia asistida con la mano (*HALS Hand Assisted Laparoscopic Surgery*) [5] [11].

Las ventajas de la cirugía laparoscópica son: una mejor visión y abordaje de la cavidad abdominal, disminución del dolor postoperatorio, reducción de la posibilidad de complicaciones infecciosas, acortamiento del tiempo de estancia hospitalaria, mayor rapidez en la reincorporación a la vida social y laboral y mejor resultado estético [7].

A partir del año 2000 aparecen las primeras aplicaciones de la robótica a la cirugía laparoscópica, destinadas a soslayar algunas limitaciones de esta: pérdida de la sensación táctil, limitaciones en la maniobrabilidad, campo visual bidimensional, particularidades del instrumental, complejidad de los procedimientos y necesidad de neumoperitoneo [12].

La calidad de la imagen tridimensional intraoperatoria con sensación de profundidad, la perfecta sincronización manos-ojos, la precisión de los instrumentos, la exactitud de sus suturas y la exéresis que se realiza con mayor destreza y confort, hará probablemente que pronto la cirugía robótica sea ampliamente difundida a pesar de su elevado coste [1].

En la actualidad aparecen nuevos intentos de acceder a la zona quirúrgica con la mínima invasión y han aparecido nuevos conceptos, como la cirugía endoscópica transluminal por orificios naturales (*NOTES, Natural Orifice Transluminal Endoscopic Surgery*), la cirugía laparoscópica a través de incisión única (*SILS, Single Incision Laparoscopic Surgery*) y la cirugía transanal a través de puerto único (*TAMIS, Transanal Minimally Invasive Surgery*) [4].

Existe una tendencia clara a avanzar cada vez más hacia técnicas menos invasivas, intentando lograr una cirugía más segura para el paciente y más cómoda para el cirujano.

### 1.1 CIRUGÍA HALS

Las técnicas laparoscópicas presentan ciertas limitaciones como:

- Capacidad restringida de manipulación del espécimen enfermo.
- La reducida retroalimentación táctil asociada con un abordaje totalmente laparoscópico.

- La imposibilidad de una visualización completa del campo quirúrgico.
- La larga duración de las operaciones de cirugía laparoscópica.

La cirugía laparoscópica asistida manualmente (HALS), es una técnica algo diferente de la tradicional cirugía laparoscópica. Con HALS, se utiliza un aparato para mantener el neumoperitoneo mientras se inserta la mano del cirujano a través de una pequeña incisión en el abdomen. Como en la cirugía laparoscópica estándar, el cirujano visualiza la zona mediante un monitor, pero además tiene la ventaja que implica el uso de su mano. La mano posee 27 grados de libertad que proporcionan una gran versatilidad para llevar a cabo todo tipo de operaciones. Debido a que la HALS permite el mantenimiento de la sensación táctil y promueve un grado de coordinación mano-ojo, esta variación en la cirugía laparoscópica ha sido más fácil de dominar para los cirujanos entrenados exclusivamente en la cirugía abierta.

Siguiendo el enfoque de la cirugía laparoscópica, el lugar de la mano debe ser considerado como un puerto de la operación y triangulado con el otro puerto de la operación laparoscópica de manera que ambos puertos formen ángulos de azimut iguales con el puerto de visualización laparoscópica. El posicionamiento de los puertos de la operación en ángulos iguales permite al cirujano dirigirse ergonómicamente al órgano objetivo.

Debido a que la tecnología HALS está en sus comienzos, existe la necesidad urgente de desarrollar estudios adecuados para explorar su potencia. Por último, se necesitan nuevos métodos para promover la educación quirúrgica y difundir el conocimiento de las capacidades de la cirugía HALS [2] [8].

## 2 SISTEMA DE VISION PARA EL GUIADO DEL ROBOT QUIRÚRGICO

Con objeto de robotizar el proceso de cirugía HALS, presentaremos el sistema de visión estereoscópico que tiene por objeto identificar la mano del cirujano en el interior del abdomen. La información obtenida por este sistema se empleará para el guiado de un robot colaborativo que cooperará con el cirujano en aquellas maniobras que este lo requiera.

La obtención de la posición de la mano se llevará a cabo por dos cámaras de pequeño tamaño introducidas en el abdomen del paciente que, a partir de las parejas de imágenes estereoscópicas, permitirán la reconstrucción tridimensional de la mano del cirujano,

evitando así que el robot colisione con esta en su avance.

La identificación de la mano del cirujano dentro de la cavidad abdominal del paciente resulta compleja, debido a la gran cantidad de grados de libertad que esta tiene y a que en muchas situaciones esta aparece parcialmente oculta. Para plantear una solución inicial se ha empleado una identificación empleando el color.

### 2.1 CORRESPONDENCIA ESTEREOSCOPICA

La correspondencia estereoscópica consiste en la asociación de las proyecciones de un punto de la escena en las imágenes del par estereoscópico.

El proceso de correspondencia es el paso más complejo en la visión estereoscópica. El hecho de tener dos imágenes tomadas desde dos cámaras y desde posiciones o ángulos diferentes hace que la iluminación pueda ser diferente en las imágenes, incluso con la aparición de distintos reflejos. Además, existe la posibilidad de que se produzcan oclusiones quedando ocultas partes de la escena en una imagen y visibles en la otra. Por otro lado, aunque las imágenes sean realizadas con cámaras con las mismas características, el comportamiento de los componentes electrónicos y ópticos puede ser distinto.

Existen varias técnicas de correspondencia estereoscópica que difieren en los elementos identificativos utilizados para llevar a cabo la correspondencia:

- Las **técnicas basadas en área** que utilizan patrones de intensidad en los que engloban a la vecindad de un píxel de una de las imágenes del par estereoscópico y los comparan con los patrones de intensidad correspondientes con un píxel de la otra imagen del par estereoscópico. La principal ventaja de este tipo de técnicas es la creación de mapas de profundidad muy densos, esto se debe a que la correspondencia se realiza píxel a píxel, obteniendo un valor de disparidad para cada uno. También presentan una serie de desventajas como: el uso de la intensidad en cada píxel hace que sean más sensibles a modificaciones en la iluminación, y la posibilidad de obtener falsas correspondencias.
- Las **técnicas basadas en las características** que emplean representaciones simbólicas obtenidas a partir de la imagen de intensidad. Algunas de estas representaciones simbólicas son: los puntos de borde aislados, cadenas de puntos de bordes, regiones delimitadas por bordes. Algunas de las principales ventajas que presentan estas técnicas

son: estabilidad ante cambios en la iluminación, permiten realizar comparaciones entre atributos, mayor rapidez debida al procesamiento de menos puntos. La principal pega es su falta de densidad en los mapas de profundidad.

## 2.2 CÁLCULO DE LA PROFUNDIDAD

Para la determinación de las distancias a la que se encuentra la mano del cirujano en nuestra escena se ha de realizar la correspondencia entre las imágenes izquierda y derecha. La forma de obtener las distancias está ligada con la geometría del sistema estereoscópico. Debido a la restricción epipolar [6], el cálculo de las distancias es una simple triangulación. Estando las proyecciones de un objeto en la misma posición en el eje vertical.

Las imágenes del par estereoscópico presentan un desplazamiento horizontal que hace que las proyecciones de los objetos de la escena tengan posiciones en el eje horizontal ligeramente distintas. Esta diferencia en el posicionamiento en el eje horizontal es lo que se conoce como disparidad.

Aplicando la semejanza de triángulos para ambas proyecciones, se obtiene los valores de  $x_I$ , ecuación 1 y de  $x_D$ , ecuación 2. Sabiendo que la diferencia de estos valores se corresponde con la disparidad se obtiene la coordenada z del punto de la escena.

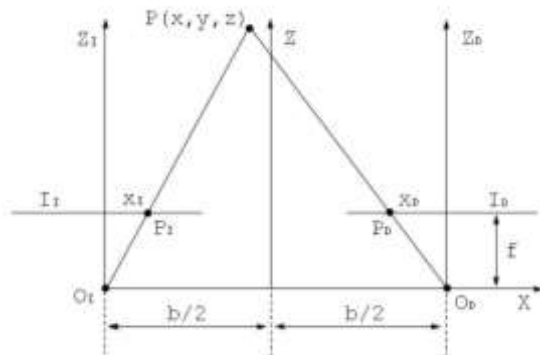


Figura 1 Geometría del estereoscopio

$$x_I = \frac{f}{z} \left( x + \frac{b}{2} \right) \quad (1)$$

$$x_D = \frac{f}{z} \left( x - \frac{b}{2} \right) \quad (2)$$

$$d = x_I - x_D \rightarrow z = \frac{f}{d} b \quad (3)$$

Como se puede observar en la ecuación 3, la distancia es inversamente proporcional a la disparidad, es decir, cuanto más cerca se encuentre un objeto en la escena con mayor disparidad aparecerá en las imágenes.

Como ya se comentó anteriormente dependiendo de la aplicación del sistema estereoscópico se necesitan mapas de profundidad más o menos densos. Dependiendo de la técnica utilizada para la creación del mapa de disparidad será necesario llevar a cabo la interpolación de este.

Tras el cálculo de disparidades cabe la posibilidad de encontrarnos con errores en los mapas de profundidad. Estos se pueden deber a errores en el cálculo de los valores de disparidad. Para solucionar estos problemas se aplican una serie de filtros sobre el mapa de disparidades.

El tratamiento de los mapas de disparidad se realiza basándose en la idea de que un píxel propio de un objeto de la escena tendrá un valor de disparidad parecido al resto de píxeles pertenecientes al objeto. Los filtros más utilizados son el filtro de la mediana y el filtro de la media.

## 3 IMPLANTACIÓN

Para la programación se ha empleado la biblioteca de procesamiento de imágenes OpenCV [9], que tiene una serie de funciones que permiten la obtención de correspondencias mediante emparejamiento por áreas (*Block Matching* en OpenCV).

### 3.1 PUESTA EN CORRESPONDENCIA E INTERPOLACIÓN

La implementación de algoritmos basados en áreas en OpenCV se realiza mediante los siguientes pasos:

- Un prefiltrado de la imagen para normalizar o intensificar la intensidad de las imágenes.
- Búsqueda de correspondencias entre ambas imágenes.
- Filtrado de los resultados para eliminar posibles ruidos presentes en el mapa de disparidades.

Todos los algoritmos basados en áreas para obtener las correspondencias estereoscópicas comparan el valor tanto de los píxeles evaluados como el de sus vecindades, el conjunto formado por el píxel y su vecindad es conocido como ventana.

En el **prefiltrado** de las imágenes de entrada se puede llevar a cabo una normalización de las imágenes o la intensificación de los bordes. El normalizado de la imagen se realiza mediante la resta del valor medio de intensidad de los píxeles de su ventana, estableciendo un máximo y un mínimo. Con el normalizado se consigue reducir el ruido presente en las imágenes. La intensificación de los bordes se realiza asignando pesos a los píxeles de una ventana, siendo mayor el peso de los píxeles centrales que el de los extremos, consiguiendo resaltar los bordes de los objetos.

La **búsqueda de correspondencias** en las funciones de OpenCV utilizan la restricción epipolar y buscan la proyección de un píxel en una imagen solo a lo largo de la línea epipolar en la otra. El algoritmo selecciona una ventana en una de las imágenes y realiza el sumatorio de las intensidades de los píxeles en dicha ventana. A continuación, recorre los píxeles de la línea epipolar de la otra imagen, estableciendo ventanas del mismo tamaño y calculando el sumatorio de las intensidades. La correspondencia se establece entre el píxel de la primera imagen y el píxel de la segunda cuyas ventanas tengan la menor diferencia entre los sumatorios de intensidad. El valor almacenado para esa posición se corresponde con el desplazamiento entre ambos píxeles.

En el **postprocesado** se realiza un filtrado de los resultados obtenidos mediante la búsqueda de correspondencias. Este filtrado es necesario para la eliminación de falsos positivos. Estos son debidos a que el algoritmo selecciona el mejor de los resultados, pero se puede dar el caso que todos los resultados obtenidos son malos, y el algoritmo selecciona el menos malo. Otro aspecto que trata el postprocesado son las zonas correspondientes a bordes de objetos y el nivel de textura en las ventanas.

Los diferentes tipos de algoritmos para la obtención de las correspondencias estereoscópicas implementados en una función presente en la biblioteca de OpenCV son:

- *Graph-Cuts*, correspondencia densa global. Los algoritmos de correspondencia globales intentan encontrar similitudes entre intensidades de píxeles a nivel de imagen completa, creando áreas de píxeles con el mismo nivel de similitud. Afectando la correspondencia de un píxel a sus vecinos (Mujika, 2010). Este algoritmo agrupa los píxeles en función de la profundidad y la similitud, tratando de minimizar la función de energía. Permitiendo identificar formas y bordes, y reduciendo los problemas causados por las oclusiones, iluminación y textura. El problema es que esto supone un coste de procesamiento muy alto, inhabilitando el uso de este algoritmo en sistemas en tiempo real.
- *Block Matching*, correspondencia densa local. Este algoritmo busca las correspondencias para cada píxel, estableciendo las disparidades mediante la comparación de las intensidades de los píxeles en las ventanas. La principal ventaja de este algoritmo es su bajo coste computacional, soportando la carga de trabajo de sistemas de visión en tiempo real. Las principales desventajas son dos. La primera de ellas es el resultado obtenido en los bordes de las imágenes. El hecho de que las imágenes estén tomadas desde diferentes posiciones hace que los bordes de los objetos en las imágenes no coincidan. Cuando se

implementan algoritmos de Block Matching se obtienen resultados muy malos debido a las grandes diferencias de intensidad entre las ventanas situadas en estos bordes. Una segunda desventaja es la posibilidad de encontrarse con texturas muy similares, esto impide que el algoritmo encuentre correspondencias para esas zonas.

- *Semi-Global Block Matching*, correspondencia densa híbrida. Este tipo de algoritmos se encuentran situados entre los algoritmos locales y globales. Es decir, intentan obtener unos resultados cercanos a los de los algoritmos globales, pero a un coste computacional tan bajo como en los locales. Este tipo de algoritmo lleva a cabo la correspondencia estereoscópica mediante la minimización de una función de energía de en las ventanas en las que previamente divide la imagen. Respecto a la correspondencia densa local, aumenta el coste computacional del algoritmo, pero mejora significativamente la calidad de los resultados

Comentadas las características principales de cada uno de los algoritmos se ha de elegir uno para implementarse en el sistema de visión estereoscópica en desarrollo. El sistema va a trabajar en tiempo real, esto obliga a desechar la utilización del algoritmo *Graph-Cuts*.

El algoritmo de correspondencia densa local, *Block Matching*, esta implementado en OpenCV por la función *stereoBM*. Esta función presenta una serie de parámetros de entrada que permiten modificar el proceso de obtención de correspondencias estereoscópicas, alterando el mapa de disparidad que muestra la función como resultado de su ejecución.



Figura 2. Imagen original. Corresponde a la imagen capturada por una de las cámaras del sistema de visión estereoscópico. La mano aparece cubierta por un guante sensorizado que proporcionará datos sobre la flexión y movimiento de los dedos.



Figura 3. Mapa de disparidad obtenido para el par correspondiente a la imagen de la figura 1. Las zonas más claras indican una mayor proximidad a la cámara.



Figura 4. Imagen en la que mediante un filtro de color, se han eliminado todos aquellos elementos que no correspondan a los dedos. Esto permite disminuir significativamente el número elementos en la nube de puntos y trabajar con componentes conexas que aproximaremos a cilindros.

### 3.2 PROCESAMIENTO DE LA NUBE DE PUNTOS

Con el mapa de profundidades obtenido ya se puede establecer de forma muy aproximada dónde se ubica la mano del cirujano dentro del abdomen. Esta información es suficiente para evitar la colisión del robot quirúrgico con la mano. No obstante, para lograr que el robot asista al cirujano en su quehacer es preciso procesar el mapa de profundidades para extraer las componentes conexas que corresponden a cada dedo. Más concretamente, será preciso identificar las puntas de estos ya que será ahí donde se llevará a cabo la cooperación.

Para ello, partiendo de la matriz de datos que contiene las coordenadas XYZ de los puntos pertenecientes a los dedos del cirujano convertiremos esta información a un formato en el cual podamos trabajar

eficientemente en espacio tridimensional. Para conseguir esto se van a utilizar las librerías *Point Cloud Library (PCL)* [10], un software libre destinado a la obtención y tratamiento de nubes de puntos en el espacio.

Antes de continuar, es importante analizar qué tipo de nube de puntos vamos a utilizar para almacenar la información. PCL dispone de varios tipos haciendo distinción principalmente en el tipo de punto que la nube va a almacenar. Se puede hacer uso de puntos XY los cuales solo tienen 2 coordenadas en el espacio, de puntos XYZ que tienen 3, los XYZI tienen, a parte de las tres dimensiones, un parámetro más que hace referencia a la intensidad pudiendo así crear una nube de puntos en escala de grises. También están los puntos XYZRGB que al igual que los anteriores ubican puntos en el espacio tridimensional pero además con información cromática.

Dado que en el proceso para conseguir el estéreo se ha convertido la imagen a escala de grises y que más tarde la función que nos proporcionaba las coordenadas de los puntos de la imagen no nos da la información del color, lo más oportuno es utilizar un tipo de punto XYZ. Esto, además, resulta lógico ya que como veremos más adelante el uso del color en la nube de puntos no va a ser necesario para el proceso planteado. Añadirlo solo provocaría un aumento del uso de memoria y tiempo de ejecución.

Una vez completado el volcado de la información a una nube de puntos en PCL las librerías de OpenCV ya no serán necesarias y podremos trabajar libremente con la librería PCL. El resultado de la conversión se puede ver en la Figura 5.

### 3.3 SEGMENTACIÓN POR CILINDROS

Ahora que se tiene una nube de puntos en el formato deseado podemos proceder a la búsqueda de cilindros en ésta con objeto de obtener cada dedo por separado.

La búsqueda se realiza mediante la función de PCL *sac\_segmentation* la cual se basa en la segmentación de una nube de puntos a partir de un modelo dado.

Para un correcto funcionamiento de esta función debemos indicarle diferentes parámetros, comenzando con la nube de puntos de entrada y de salida, éstas pueden ser la misma o no. Además, la función necesita que se le indiquen los valores de las normales a cada punto, es decir, debemos obtener previamente la dirección normal de cada punto de la nube. Este proceso se realiza mediante la función *ExtractIndices*.

Otro dato que debemos introducir es el algoritmo de búsqueda que debe utilizar la función. De entre todas las opciones destaca el método de RANSAC [3].

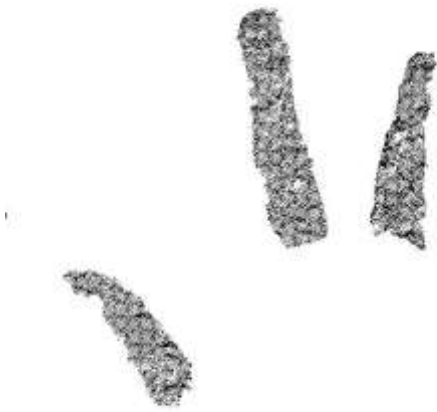


Figura 5. Nube de puntos correspondientes a los tres dedos

Continuamos indicando el modelo que deseamos buscar en nuestra nube de puntos. Estos modelos están predefinidos en PCL y tenemos diversas opciones. Entre ellas hay planos, círculos, esferas o cilindros. Dependiendo del tipo de modelo que se escoja, después se deberá indicar unos parámetros u otros para definir las características del mismo. Si por ejemplo escogemos un plano no tendremos que definir más parámetros. En el caso de un cilindro tendremos que indicar un rango en el cual puede estar el diámetro del mismo.

Por último, se debe indicar a la función valores como el número máximo de iteraciones en la búsqueda, la tolerancia para considerar que un punto forma parte o no del modelo o el peso que tendrá el valor de las normales.

Una vez se han definido correctamente todos estos valores la función puede ejecutarse dándole 2 matrices de datos. En la primera se indicarán los puntos de la nube que conforman el modelo encontrado en caso de que lo hubiera. En la segunda se guardarán los parámetros del modelo encontrado, es decir que en función del tipo de modelo que haya buscado, nos devolverá los parámetros que definen a ese modelo. Para el caso del plano devuelve 4 valores que corresponden a los cuatro parámetros de la ecuación 4:

$$ax + by + cz + d = 0 \quad (4)$$

Para el caso del cilindro se necesitan 7 parámetros: los 3 primeros dan las coordenadas XYZ de un punto en el espacio, los 3 siguientes aportan la dirección de la recta que pasa por dicho punto y que será el eje del

cilindro, por último, el séptimo valor nos proporciona el radio del cilindro.

En el caso particular de nuestra aplicación, disponemos de una nube de puntos que representa tres dedos de una mano, por lo tanto, si queremos separar dichos dedos para estudiarlos uno por uno tendremos que hacer la búsqueda de cilindros tres veces consecutivas para encontrar cada uno de los dedos. Es importante que tras cada búsqueda se eliminen de la nube original aquellos puntos que ya han sido considerados como pertenecientes a un dedo. Si no se eliminan, todas las búsquedas encontrarán siempre el mismo dedo, aquel que mejor se aproxime a un cilindro o, empleando la nomenclatura de RANSAC, el dedo que más consenso presente.

Por lo tanto, el proceso consiste en realizar la búsqueda, extraer aquellos puntos que pertenezcan al cilindro encontrado y volcarlos en una nueva nube de puntos. Además de almacenar los 7 parámetros que definen a cada cilindro para poder después encontrar fácilmente cada dedo. El resultado obtenido tras este proceso se puede visualizar en la Figura 6.

### 3.4 DETECCIÓN DE LOS EXTREMOS DE LOS DEDOS

Para identificar la posición de las puntas de los dedos existen diversas opciones, pero todas ellas han de enfrentarse al mismo problema, determinar en la nube de puntos segmentadas por cilindros qué extremo corresponde a la falange distal del dedo y cuál a la proximal.

Este problema existe debido a que se han cogido únicamente los puntos pertenecientes a los dedos, eliminando así los del resto de la mano. De este modo, perdemos la información de dónde está la mano y por lo tanto su orientación.

Para solucionar este problema, se ha recurrido a asumir que la orientación de la mano solo puede estar entre unos determinados valores. Esta suposición puede parecer muy restrictiva pero dado que en nuestra aplicación real el cirujano deberá introducir su mano por un puerto en el abdomen del paciente, la movilidad que tendrá esta será limitada y por tanto esta suposición resulta perfectamente válida.

Asumida esta premisa, el determinar la punta de los dedos se reduce a encontrar el punto más alejado dentro de la nube de puntos correspondiente a cada dedo. Para ello se analiza la coordenada Y de cada punto para después almacenar la memoria el de mayor valor. Este sistema de detección resulta realmente rápido y eficaz. Se puede ver el resultado de la operación en la Figura 7.

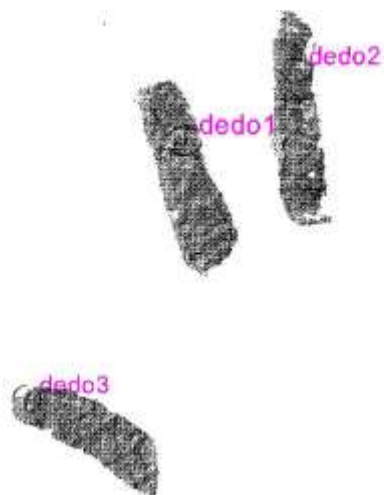


Figura 6. Cada componente conexa correspondiente a un dedo se ha aproximado por un cilindro.

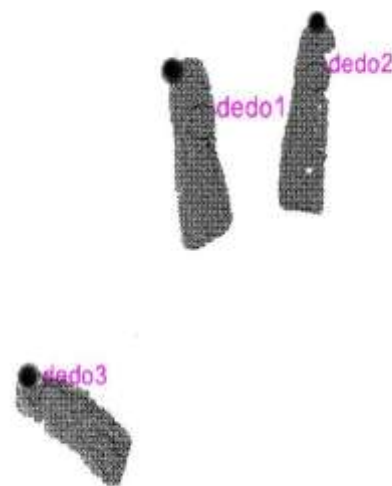


Figura 7. El círculo representado en cada dedo indica cuál es el extremo del dedo.

#### 4 ANÁLISIS DE RESULTADOS

Los resultados obtenidos sobre esta primera versión de la aplicación tanto en el emparejamiento, Figura 3, como la obtención de primitivas 3D/cilindros Figura 7 son satisfactorios.

El algoritmo ha sido probado en un ordenador con unas prestaciones medias. Se ha podido realizar el procesamiento estéreo de imágenes con resolución 640x480 estéreo a 5 y 6 frames por segundo. Este procesamiento permite obtener el mapa de profundidades válido para la evitación de colisiones por parte del robot quirúrgico. Sin embargo, el procesamiento de la nube de puntos para establecer la posición y orientación de cada dedo, necesario para la cooperación del robot con el cirujano, tarda alrededor de 1 s. Este tiempo es muy elevado para un funcionamiento en una aplicación real. En la actualidad se está trabajando en la reducción de este tiempo de ejecución disminuyendo la resolución de la nube de puntos y paralelizando el procesamiento estéreo con el de la nube.

En cuanto a la precisión lograda en el mapa de profundidad, los errores relativos obtenidos en el cálculo de las distancias nunca han sido superiores a un 2% y son por tanto perfectamente admisibles para el guiado del robot quirúrgico.

#### 5 CONCLUSIONES

El sistema de visión que se presenta en este artículo se encuentra enmarcado dentro de un proyecto de creación de una plataforma robotizada capaz de reconocer y predecir los pasos quirúrgicos y la anatomía específica del paciente para colaborar de forma automática con el cirujano.

En este artículo se presentan los resultados obtenidos en el desarrollo del sistema de visión para el guiado de este robot quirúrgico. Los resultados muestran la viabilidad del sistema diseñado para el guiado automático del robot, siendo posible su implantación en tiempo real. La identificación de los dedos del cirujano, objetivo necesario para la cooperación del robot con este, requiere de un procesamiento computacionalmente más costoso y en el que debe avanzarse más para que sus tiempos de ejecución sean acordes a los que requiere la plataforma robotizada.

#### Agradecimientos

Este trabajo ha sido financiado por el Ministerio de Economía y Competitividad, a través del proyecto: "Sensorized HALS. Entorno sensorizado para cirugía laparoscópica asistida por la mano". Subproyecto dentro del proyecto coordinado "Sistema robotizado colaborativo para cirugía laparoscópica asistida por la mano". DPI2013-47196-C3-3-R.

#### Referencias

- [1] Amodeo, A. y Linares, A., 2009. Robotic laparoscopic surgery: cost and training. *Minerva Urol Nefrol.*, 61(2), pp. 121.
- [2] Ding J., Xia Y., Liao G., Z. Zhang, S. Liu, Y. Zhang, and Z. Yan, "Hand-assisted laparoscopic surgery versus open surgery for colorectal disease: a systematic review and meta-analysis," *The American Journal of Surgery*, vol. 207, no. 1, pp. 109–119, 2014.
- [3] Fisher, M. A. y Bolles, R. C. "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, n°6, pp. 381-395, 1981.

- [4] Flora, E., Wilson, T.G. y Martin, I.J., 2008. A review of natural orifice transluminal endoscopic surgery (NOTES) for intraabdominal surgery: experimental models, techniques and applicability to the clinical setting. *Annals of surgery*, 247, pp. 583-602.
- [5] Gorey, T.F., Tierney, S., Buckley, D., O'Riordain, M., Fitzpatrick, J.M., "Video-assisted Nissen's fundoplication using a hand-access port", *Minimally Invasive Therapy and Allied Technologies*, 5 (4), pp. 364-366, 1996.
- [6] Hartley, R. and Zisserman, A. Multiple View Geometry in Computer Vision. Second Edition. Cambridge University Press, March 2004.
- [7] Iturralde, A.R., González, T. y Castillo, M., 2010. Capítulo II. Principales acontecimientos históricos de la cirugía urológica. En: M. Frank y W. Castro López, eds. *Cirugía Urológica de mínimo acceso*. pp. 54.
- [8] Leraas H., Ong C.T., Sun Z., Adam M.A, Kim J., Gilmore B.F, Ezekian B., Nag U.S., Mantyh C.R., Migaly J. "Hand-Assisted Laparoscopic Colectomy Improves Perioperative Outcomes Without Increasing Operative Time Compared to the Open Approach: a National Analysis of 8791 Patients", *J Gastrointest Surg*;21(4):684-69, April 2017
- [9] "OpenCV" Available: <http://opencv.org/> .
- [10] "Point Cloud Library" Available: <http://pointclouds.org/> .
- [11] Targarona, E.M., Gracia, E., Garriga, J., Martínez-Bru, C., Cortés, M., Boluda, R., Lerma, L., Trías, M., "Prospective randomized trial comparing conventional laparoscopic colectomy with hand-assisted laparoscopic colectomy: Applicability, immediate clinical outcome, inflammatory response, and cost", *Surgical Endoscopy and Other Interventional Techniques*, 16:2, 2002, pp. 234-239, 2002.
- [12] Yoshino, I., Hashizume, M. y Shimada, M., 2001. Thoracoscopic thymectomy with the Da Vinci computer-enhanced surgical system. *Journal of Thoracic & Cardiovascular Surgery*, 122(4), pp. 783-785.