

PROCESAMIENTO PARALELO DE IMÁGENES DE VIDEO PARA LA DETECCIÓN DE SANGRADO Y VENDAS EN OPERACIONES DE CIRUGÍA LAPAROSCOPICA

Dylan Derwael

Katholieke Hogeschool Limburg, Bélgica, 1135195@student.khlim.be

Rubén García, Eusebio de la Fuente, Félix M. Trespaderne, Juan C. Fraile, Javier P. Turiel, Lidia Santos
ITAP - Instituto de las Tecnologías Avanzadas de la Producción, Universidad de Valladolid.
ruben.garcia.sanchez@alumnos.uva.es; { efuente, trespa, jcfraile, turriel, }@eii.uva.es, lidia.santos@uva.es

Resumen

Se presenta una aplicación de procesamiento de imagen para un entorno robotizado destinado a asistir a un médico en operaciones de cirugía laparoscópica. Durante la operación el robot, a través del sistema de visión, deberá detectar cuándo aparece una situación de sangrado dentro de la cavidad abdominal con objeto de ayudar al cirujano a atajar esta situación. El sistema de procesamiento de imágenes se ha ampliado también a la detección de vendas para garantizar que se retiran todas al finalizar la operación. Con objeto de permitir un procesamiento en tiempo real de las imágenes de vídeo, el sistema desarrollado realiza un procesamiento en paralelo dividiendo las diferentes tareas entre todos los núcleos existentes en el procesador empleando la biblioteca Intel TBB.

Palabras Clave: Cirugía Laparoscópica, Procesamiento de Imagen, Procesamiento Paralelo.

1 INTRODUCCIÓN

Considerada como la única técnica que ha supuesto un cambio revolucionario en el campo quirúrgico en el último siglo [13] [10], y cuyo empleo se ha expandido de forma constante desde los años setenta [14], la laparoscopia permite examinar y actuar sobre los órganos de las cavidades abdominal, torácica y la pelvis. La tradicional premisa de que los grandes problemas quirúrgicos implican grandes incisiones [5] deja de cumplirse con la aplicación de este tipo de intervenciones mínimamente invasivas. En la actualidad la laparoscopia en sus diferentes versiones se puede encontrar en un amplio y diverso rango de especialidades: Cirugía General y del Aparato Digestivo, Ginecología y Obstetricia, Urología, Cirugía Pediátrica, Cirugía Torácica, Cirugía Vascular, Neurocirugía y Traumatología, entre otras.



Figura 1. Representación esquemática de los dispositivos de entrada típicos (trócares) del instrumental laparoscópico [11]

En la ejecución del proceso laparoscópico se realizan una o varias incisiones de 1 cm en el abdomen, permitiendo de esta forma el uso del abordaje endoscópico como instrumento quirúrgico (Figura 1).

Numerosos estudios [16] [3] [9] han revelado los mejores resultados terapéuticos, a nivel perioperatorio, mediante la aplicación de la laparoscopia frente a la laparotomía (operación quirúrgica en la cual se abre el abdomen):

- Hospitalizaciones más cortas
- Incisiones más pequeñas
- Recuperaciones funcionales más rápidas
- Menor uso de analgésicos
- Menor pérdida de sangre
- Menor dolor posoperatorio

No obstante, la estandarización de esta técnica quirúrgica ha generado cierta resistencia entre algunos miembros de algunas especialidades en la comunidad médica debido, principalmente, a los siguientes factores:

- incremento del tiempo de operación
- acceso limitado a la formación
- aumento de la complejidad de la operación
- mayores dificultades técnicas en la cirugía

Aunque estas limitaciones han influido notablemente en el incremento de su curva de aprendizaje [15], la

robótica médica puede contribuir en la mejora de la capacidad técnica humana para llevar a cabo este tipo de intervenciones [17].

La inteligencia artificial ha estado presente de forma determinante en el avance, entre otras, de la cirugía mínimamente invasiva. En la categoría de los sistemas de soporte de la cámara se pueden destacar [8] [2] *AESOP (Automated Endoscopic System for Optimal Positioning)* cuyo guiado se realiza de forma manual o a través de la voz; *EndoAssist*, dirigido mediante los movimientos de la cabeza del cirujano; *SoloAssist*, accionado con un pequeño joystick; *ViKY*, cuya cámara se dirige vía voz o con el pie; *Passist*, dispositivo de sujeción pasivo; y *LapMan*. Así mismo, de forma paralela han aparecido varios desarrollos de tipo maestro-esclavo en el área de la telepresencia laparoscópica, los sistemas *Zeus* y *da Vinci*, que han aportado tridimensionalidad aumentada de alta resolución, filtrado de movimientos parásitos del cirujano y precisión de los instrumentos al reproducir los movimientos de la mano, muñeca y dedos.

Resulta obvio que a pesar de las ventajas que presenta la laparoscopia desde el punto de vista clínico, esta técnica disminuye en el cirujano la percepción de la profundidad, la coordinación mano-ojo y la realimentación de fuerza al manipular los tejidos con microinstrumentos. Por ello, en los últimos años se está realizando un esfuerzo de investigación significativo para desarrollar prototipos de robots quirúrgicos que faciliten el trabajo al cirujano y permitan solventar los inconvenientes y disminuir las desventajas. Dentro de esta línea se encuentra este artículo en el que se presenta un sistema de procesamiento de imagen destinado a asistir a un médico en operaciones de cirugía laparoscópica. Durante la operación la plataforma robotizada, a través del sistema de visión, deberá detectar cuando aparece una situación de sangrado dentro de la cavidad abdominal con objeto de ayudar al cirujano a resolver rápidamente esta situación.

El sistema de procesamiento de imágenes también se ha desarrollado para la detección de vendas, garantizando que se retiran todas de la cavidad abdominal al finalizar la operación.

2 DETECCIÓN DE SANGRADO

En las operaciones de laparoscopia es frecuente que los cortes afecten a vasos sanguíneos del paciente y se produzcan situaciones de sangrado que deben ser atajadas rápidamente. Aunque esto es propio de cualquier operación de cirugía, en el caso de la laparoscopia estas situaciones son más complicadas de abordar por las limitaciones que tiene esta técnica

sobre la percepción y sensibilidad del cirujano y su movilidad.

Por este motivo se ha desarrollado una aplicación de procesamiento de imagen que, partiendo de la secuencia de video proporcionada por el endoscopio, sea capaz de detectar estas situaciones. Esto permitirá en un futuro asistir al cirujano ante la aparición de estas condiciones de sangrado mediante un robot que opere de forma totalmente autónoma.

Con objeto de localizar las situaciones de sangrado trataremos de detectar una presencia significativa de sangre en las imágenes. Aunque la sangre tiene un color muy característico, este no lo es tanto en un escenario de laparoscopia donde su color se confunde fácilmente con el del tejido humano.

Para de detectar las partes con presencia de sangre en cada imagen del vídeo (figura 2), hemos empleado un par de condiciones sobre el color. La primera es establecer un rango válido sobre las componentes de la señal RGB. Puesto que la sangre siempre será roja se ha determinado un amplio rango para la componente R, y mucho más limitados en las componentes G y B. De esta forma cualquier otro elemento que difiera en el color tal como grasa, vendas o instrumental es fácilmente excluido.

En la segunda condición que se ha establecido se trabaja con una relación entre dos canales de color diferentes. En [4] se presenta la segmentación de imágenes capturadas por una capsula endoscópica en la que se emplea la relación B/R y G/R. Solo aquellos píxeles que presentan ambas relaciones por debajo del 0.6 son considerados como sangre. En nuestro caso hemos bajado el valor a 0.4 puesto que las imágenes de la cavidad abdominal presentan un mayor contenido en rojo que en el interior del intestino. Serán considerados como sangre aquellos píxeles que además de la primera condición relativa al color cumplan las siguientes relaciones:

$$\begin{aligned} 0.1 < (B/R) < 0.4 \\ 0.1 < (G/R) < 0.4 \end{aligned} \quad (1)$$

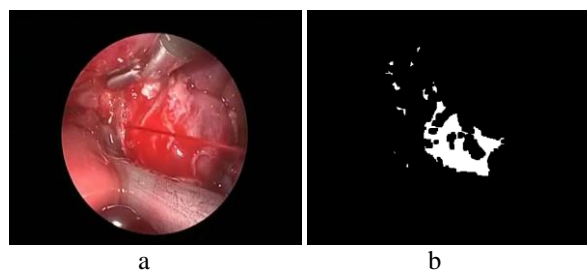


Figura 2. a) Imagen original donde aparece un proceso sangrado. b) Píxeles marcados como sangre.

3 DETECCIÓN DE LA PRESENCIA DE VENDAS EN LA CAVIDAD ABDOMINAL

El procesamiento de imágenes no se ha limitado a la detección de sangrado sino que se ha ampliado también al control de vendas en el interior de la cavidad abdominal para garantizar que se hayan retirado todas cuando la operación haya finalizado.

En la detección de vendas, la utilización del color no es del todo discriminante. La apariencia de una venda, que inicialmente presenta un color blanco muy característico (fig.3a), cambia radicalmente cuando esta se empapa de sangre (fig.6a). Sin embargo, a pesar de este cambio en el color, la venda sigue manteniendo una textura más o menos reticulada que resulta muy particular. Por ello se analizará la textura para determinar la presencia de una venda en la imagen.

El análisis de textura [7] se ha empleado en muchas aplicaciones que van desde el procesamiento de imágenes de satélite hasta imágenes médicas pasando por sistemas de inspección industrial. El análisis de textura en una imagen proporciona información acerca de la distribución espacial del color o intensidades en las regiones de la imagen. La textura se puede cuantificar calculando un conjunto de métricas. El grado de textura que describimos intuitivamente con calificativos como áspera, suave, granulada, rayada, ... puede medirse en función de la variación local del color o nivel de gris en los píxeles.

En nuestra aplicación emplearemos el análisis de textura para extraer de la imagen las áreas que presentan una mayor textura. Para ello utilizaremos parámetros estadísticos. Estos parámetros pueden caracterizar la textura porque proporcionan información sobre la variabilidad local del nivel de gris en los píxeles. Por ejemplo, calculando la varianza del nivel de gris en la vecindad de cada píxel obtenemos una medida del grado de variabilidad de los valores de los píxeles en esa región. En áreas con una textura suave, la varianza de los niveles de gris será baja, mientras que en áreas con más textura la varianza será mayor. Análogamente se pueden calcular otros parámetros estadísticos como (fig.3):

- El contraste que proporciona una medida de las frecuencias espaciales en la vecindad del píxel.
- La energía que mide la uniformidad de la textura. Alcanza altos valores cuando la distribución de niveles de gris en la ventana tiene bien un valor constante o una forma periódica.

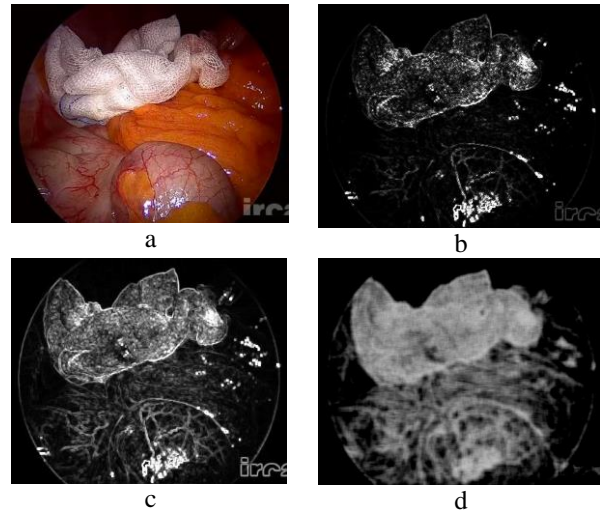


Figura 3. a) Imagen original[6]. Resultados de calcular distintos estadísticos en una ventana 7x7: b) Contraste, c) Desviación estándar y d) Entropía.

- La entropía que cuantifica el desorden en la imagen. Cuando no existe una textura uniforme la entropía es alta. La entropía está relacionada inversamente con la energía.

Veamos cómo se realizará el cómputo de la desviación estándar. Sea X una variable aleatoria con media μ :

$$E[X] = \mu \quad (2)$$

Donde el operador E es la media o valor esperado de X. La desviación estándar de X es la cantidad:

$$\sigma = \sqrt{E[(X - \mu)^2]} \quad (3)$$

Empleando las propiedades del valor esperado:

$$\begin{aligned} \sigma &= \sqrt{E[(X - \mu)^2]} \\ &= \sqrt{E[X^2] + E[(-2\mu X)] + E[\mu^2]} \\ &= \sqrt{E[X^2] - 2\mu E[X] + \mu^2} \\ &= \sqrt{E[X^2] - 2\mu^2 + \mu^2} = \sqrt{E[X^2] - \mu^2} \\ &= \sqrt{E[X^2] - (E[X])^2} \end{aligned} \quad (4)$$

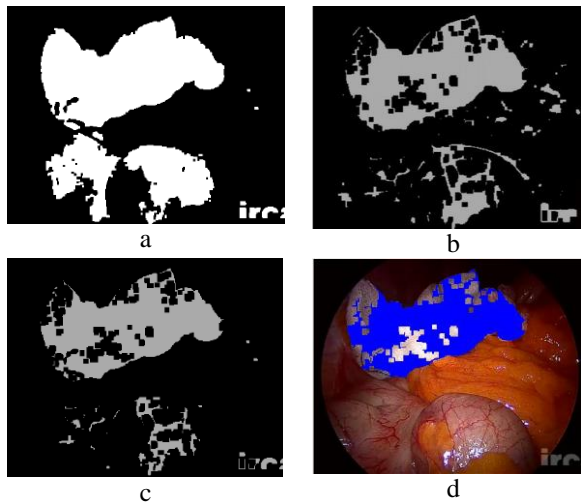


Figura 4. Resultados intermedios sobre la imagen de la fig.3a: a) segmentación por color. b) segmentación por textura (desviación estándar). c) combinación de *a* y *b* (AND lógico). d) Resultado final donde se señala la presencia de la venda sobre la imagen original.

que es la fórmula que se emplea computacionalmente para el cálculo de la desviación estándar. En el caso que nos ocupa de procesamiento de imagen el valor esperado de cada pixel $E[X]$ puede calcularse en una determinada vecindad empleando un *box filter*. Y lo mismo para el caso de $E[X^2]$ sobre la imagen al cuadrado. De esta forma la desviación estándar puede ser calculada rápidamente. Esta es una característica muy interesante cuando estamos hablando de un sistema de procesamiento de video en tiempo real.

4 PROCESAMIENTO PARALELO

Para cumplir con los requerimientos de tiempo real la aplicación para realizar un procesamiento de las imágenes en paralelo. La aplicación ha sido desarrollada con la biblioteca TBB (Intel Threading Building Blocks) y podrá ejecutarse eficientemente en procesadores con cualquier número de núcleos.

4.1 LA BIBLIOTECA INTEL TBB

La TBB es una biblioteca desarrollada por Intel que está basada en el uso plantillas y en el concepto de programación genérica de C++ [12]. Posibilita una programación paralela más directa mediante una mayor abstracción en la gestión de hilos que permite olvidarse de los detalles de la plataforma.

TBB permite especificar el paralelismo de manera más conveniente que usando los tradicionales hilos, mejorando además las prestaciones, portabilidad y escalabilidad. La escalabilidad de un programa cuantifica en qué medida aumenta su velocidad de ejecución según el número de núcleos de

procesamiento aumenta. En un contexto en el que el número de núcleos en los procesadores está aumentando constantemente, la librería hace mucho más fácil el escribir un programa que siga siendo eficiente con el aumento de la cantidad de núcleos de procesamiento.

La TBB trabaja con tareas, no con hilos, y permite que sea la biblioteca quien distribuya las tareas en hilos de una manera eficiente. Esta es una de las mayores ventajas que ofrece, que permite que la aplicación aproveche todo el potencial del hardware cuando se ejecuta en un procesador con más núcleos.

La división del programa en bloques funcionales, asignando un hilo a cada bloque, generalmente no es un buen enfoque en procesamiento de imagen porque no es escalable puesto que el número de bloques funcionales será inicialmente fijo. En la TBB emplea una técnica más flexible de programación basada en la paralelización de datos que permite a diferentes hilos trabajar en diferentes partes de la imagen. El paralelismo de datos es un paradigma muy adecuado para operaciones sobre imágenes, dado que su procesamiento consiste en aplicar la misma operación sobre cada una de sus partes.

La paralelización de datos permite reescalar el procesamiento a cualquier número de procesadores al dividir las imágenes en partes más pequeñas continuamente y recursivamente si es necesario. Esta escalabilidad hace que no se precise ningún cambio en el código fuente o los ejecutables cada vez que aparece un nuevo procesador con más núcleos.

El paralelismo de datos funciona muy bien en combinación con el equilibrado de cargas dinámico que implementa la TBB. Esta biblioteca emplea el robo de tareas (*task stealing*) para equilibrar la carga de procesamiento sobre los núcleos existentes. Al inicio la carga se distribuye equitativamente entre los distintos núcleos de procesamiento. No obstante, si en el gestor de tareas en algún momento detecta que algún núcleo está ocioso mientras otros presentan en sus colas de tareas una gran carga, el gestor redirecciona parte de la carga al núcleo desocupado. Esta capacidad de redistribución dinámica de la carga entre los núcleos desacopla la programación de la máquina y hace posible que las aplicaciones se ejecuten eficientemente cualquiera que sea el número de núcleos del procesador sin necesidad de reescribir el código.

4.2 ESTRUCTURA DE LA APLICACIÓN

La aplicación desarrollada se puede representar en un diagrama de flujo (fig. 5). En este diagrama los nodos representarían el procesamiento que se realiza en esa etapa y los arcos serían los canales de comunicación

entre esos elementos de procesamiento. Los mensajes que se pasan entre los nodos pueden contener datos o simplemente actuar como señales que indican que el nodo anterior ha finalizado.

En la biblioteca TBB existe una clase grafo y clases nodo asociadas que se emplean para expresar estas dependencias. El programador es responsable de introducir los arcos necesarios para expresar las dependencias que deben respetarse cuando los nodos se vayan a ejecutar. De esta forma se proporciona al planificador Intel TBB la flexibilidad para aprovechar el paralelismo que se especifica en la topología del grafo.

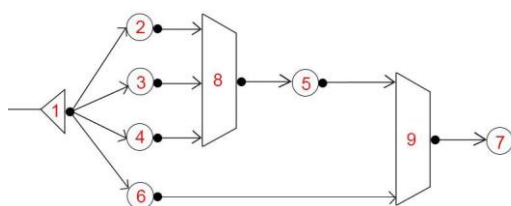


Figura 5. Estructura de la aplicación

1. Recibe cada *frame* capturada y a continuación la envía a los cuatro componentes que pueden ejecutarse en paralelo 2, 3, 4 y 6.
2. Calcula el primer sumando $E[X^2]$ para la desviación estándar (ec.4).
3. Calcula el segundo sumando $(E[X])^2$ para la desviación estándar (ec.4).
4. Obtiene imagen binaria en la que se muestran todos los píxeles cuyo color se encuentra dentro de los rangos prestablecidos para ser etiquetado como venda.
5. Cuando los componentes 2, 3 y 4 terminan se procede al cálculo de la desviación estándar a partir de los resultados proporcionados por 2 y 3. A partir de la desviación estándar se obtiene una imagen binaria que marca la presencia de texturas como la de la venda. Esta imagen binaria se combina (AND lógico) con la originada en 4 correspondiente al color proporcionando otra imagen binaria de la cual se extrae el componente de mayor área y sobrepone el resultado en la imagen en azul.
6. Este componente analiza la imagen buscando regiones de sangrado. El resultado es una imagen binaria que muestra como píxeles activos aquellas zonas que por su color se han considerado como sangre.
7. Este componente se ocupa únicamente de coleccionar todos los resultados para pasárselos al interfaz de usuario transformándolos de la estructura OpenCV Mat a formato QImage/QPixmap para el interfaz gráfico Qt.

8. Este componente espera hasta que los cálculos para obtener la desviación estándar 2 y 3 terminen así como la terminación del componente de color 4. Una vez que estos tres componentes finalizan, el componente 8 proporciona la señal para que la detección de vendas se continúe en el componente 5.
9. Este será también un componente de espera que proporcionará una señal al componente de visualización solo cuando ambas, la detección de sangre y de vendas, hayan finalizado.

5 RESULTADOS

Los resultados obtenidos sobre esta primera versión de la aplicación tanto en la detección de sangrado (fig.2) como de vendas (fig.4) son muy satisfactorios. El algoritmo funciona en tiempo real y tiene unas buenas tasas de detección. No obstante, actualmente se está trabajando para introducir mejoras en la detección de sangre introduciendo umbrales adaptativos para el color. Esto permitirá dotar de mayor fiabilidad al algoritmo cuando existan cambios significativos en la iluminación.

En cuanto a la detección de vendas esta se lleva a cabo correctamente en la mayoría de las imágenes incluso sobre vendas ensangrentadas (fig. 6). Aunque el cálculo de desviación típica parece una buena opción para un algoritmo de tiempo real por su eficiencia computacional hay que indicar que la introducción de alguna otra característica permitiría afinar más la segmentación. Esta es una línea que también se está investigando.

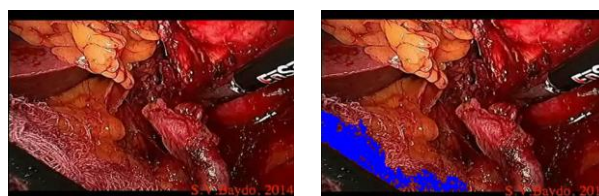


Figura 6. Detección de una venda ensangrentada en una imagen de [1].

6 CONCLUSIONES

La aplicación de procesamiento de imagen que se presenta en este artículo se encuentra enmarcada 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 una primera aplicación de procesamiento de imagen para la detección de situaciones de sangrado

y presencia de vendas. Se ha prestado especial atención a la ejecución en tiempo real lográndose una aplicación capaz de llevar a cabo la detección con buena fiabilidad. No obstante hay una serie de modificaciones previstas para mejorar la detección cuando existen variaciones significativas en la iluminación de la escena. Estas mejoras pueden introducirse sin necesidad de programar manualmente la sincronización de nuevas tareas de procesamiento gracias a la biblioteca Intel TBB. La biblioteca además permite optimizar el procesamiento en paralelo adaptándose en tiempo de ejecución al número de núcleos disponibles en el procesador.

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.

La programación de la aplicación se ha desarrollado dentro del Proyecto num.39 de Innovación Docente en asignaturas de Informática Industrial 2014-15 de la Universidad de Valladolid.

También queremos agradecer a Karlz Storz Endoscopia Ibérica la cesión para experimentación de un entorno de entrenamiento quirúrgico que incluye un *Laparoscopic Trainer*, un sistema *Tele pack X* con *Telecam camera Head PAL* y varias ópticas.

Referencias

- [1] Baydo, Sergei. <https://www.youtube.com>
- [2] Berkelman P. and Ma J. , (2009) "A Compact Modular Teleoperated Robot System for Laparoscopic Surgery," *The International journal of robotics research*.
- [3] Ding J., Xia Y., Liao G., Z. Zhang, S. Liu, Y. Zhang, and Z. Yan, (2014) "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.
- [4] Ghosh, T., Fattah, S.A. and Wahid, K.A. (2014) "Automatic Bleeding Detection in Wireless Capsule Endoscopy Based on RGB Pixel Intensity Ratio" *Conference on Electrical Eng. and Information & Communication Technology. (ICEEICT)*.
- [5] Gomez G., (2007) "Emerging Technology in surgery: informatics, electronics, robotics," in *Sabiston Textbook of Surgery*, C. M. Townsend, R. D. Beauchamp, B. M. Evers, and K. L. Mattox, Eds. Elsevier Health Sciences, pp. 464–476.
- [6] IRCAD <http://www.websurg.com/>
- [7] Haralick, R.M. ; Shanmugam, K. ; Dinstein, (1973) "Textural Features for Image Classification" *IEEE Transactions on Systems, Man and Cybernetics*, (Volume:SMC-3 , Issue: 6) Page(s): 610 - 621.
- [8] Healy D. A, S. P. Murphy, J. P. Burke, and J. C. Coffey, (2013) "Artificial interfaces ('AI') in surgery: Historic development, current status and program implementation in the public health sector," *Surgical oncology*, vol. 22, no. 2, pp. 77–85.
- [9] Heemskerk J., Zandbergen R., J. Maessen, J. Greve, and N. Bouvy, (2006) "Advantages of advanced laparoscopic systems," *Surgical Endoscopy And Other Interventional Techniques*, vol. 31, no. 9, pp. 732–737.
- [10] Kelley, W. E. (2008) "The evolution of laparoscopy and the revolution in surgery in the decade of the 1990s," *JSLs, Journal of the Society of Laparoendoscopic Surgeons*, vol. 12, no. 4, pp. 351–357.
- [11] Park, S., Bergs R. A, Eberhart R., Baker L, R. Fernandez, and J. A. Cadeddu, (2007) "Trocar-less instrumentation for laparoscopy: magnetic positioning of intra-abdominal camera and retractor," *Annals of surgery*, vol. 245, no. 3, p. 379.
- [12] Reinders, J. Intel Threading Building Blocks. (2010). O'Reilly Media. Safari Books Online. March. Pages: 336. Ebook ISBN: 978-0-596-15959-7
- [13] Rockall, T. A. and N. Demartines, (2014) "Laparoscopy in the era of enhanced recovery," *Best Practice & Research Clinical Gastroenterology*, vol. 28, no. 1, pp. 133–142.
- [14] Sanford, P. Singer, S. (2013) "Laparoscopy," *Magill's Medical Guide* (Online Edition).
- [15] Subramonian K., S. DeSylva, P. Bishai, P. Thompson, and G. Muir, (2004) "Acquiring surgical skills: a comparative study of open versus laparoscopic surgery," *European urology*, vol. 45, no. 3, pp. 346–351.
- [16] Swanstrom L. L., Kozarek R., Pasricha P. J., Gross S., Birkett D., Park P.-O., V. Saadat, R. Ewers, and P. Swain, (2005) "Development of a new access device for transgastric surgery," *Journal of gastrointestinal surgery*, vol. 9, no. 8, pp. 1129–1137.
- [17] Taylor R. H., (2006) "A perspective on medical robotics," *Proceedings of the IEEE*, vol. 94, no. 9, pp. 1652–1664.