

Detección de candidatos a pólipo en Colonoscopia Virtual mediante análisis de curvatura en CUDA

M. Poó, E. Buysse, R. Carmona, E. Coto, H. Navarro
 {malenapoo | elchuck81}@gmail.com ; {rhadames.carmona | ernesto.coto | hector.navarro}@ciens.ucv.ve

Centro de Computación Gráfica, Escuela de Computación, Universidad Central de Venezuela

RESUMEN

En la actualidad, la Colonoscopia Virtual (CV) es una importante alternativa no-invasiva para el estudio del colon. Es mucha la investigación que se le ha dedicado a este método, siendo uno de los mayores retos producir resultados precisos, en poco tiempo. Una de las fases más importantes de la CV es la detección de candidatos a pólipos, en la que se buscan automáticamente posibles lesiones en las paredes del colon. Frecuentemente, esta etapa requiere de cómputos intensivos por lo que es importante el desarrollo de nuevas técnicas para reducir su tiempo de ejecución. Este trabajo presenta una técnica de detección automática de candidatos a pólipos basada en el análisis de curvatura que reduce el tiempo de respuesta utilizando programación paralela en CUDA. Adicionalmente, se introduce una nueva técnica para descartar falsos positivos, basándose en la silueta del candidato en un corte planar. En los resultados obtenidos se evidencia una notoria reducción del tiempo de detección con respecto a una implementación en CPU y una baja tasa de falsos positivos.

ABSTRACT

Nowadays, Virtual Colonoscopy (VC) is an important non-invasive alternative for the study of the colon. Substantial research efforts have been dedicated to this method, and one of the major challenges has always been producing accurate results in a short period of time. One of the most crucial phases of VC is the detection of polyp candidates, where possible lesions on the colon walls are automatically detected. Frequently, this stage requires intensive computations and therefore it is important to develop new techniques for reducing its execution time. This paper presents a technique for automatic detection of polyp candidates based on curvature analysis that reduces the execution time using parallel programming in CUDA. Additionally, we introduce a novel technique for discarding false positive detections based on the shape of a candidate in a planar cut. The obtained results show a remarkable reduction in execution time with respect to a CPU implementation as well as a low rate of false positives.

Palabras claves: Colonoscopia virtual, detección de pólipos, análisis de curvatura, CUDA.

1. Introducción

Los pólipos colorrectales son un crecimiento anormal en el tejido a partir de la membrana mucosa que recubre la pared del colon. Si estos pólipos no son detectados y extraídos pueden degenerarse hasta convertirse en cáncer de colon, lo cual produce la muerte de cientos de personas al año en Venezuela [URL01] y miles de personas alrededor del mundo. La colonoscopia y sigmoidoscopia son los exámenes médicos clásicos que se utilizan para la detección de los pólipos, pero resultan dolorosos e incómodos para las personas, por lo que la colonoscopia virtual constituye una buena alternativa en la detección de pólipos en el colon. En la colonoscopia virtual, se captan imágenes del colon del paciente mediante una Tomografía Computarizada, y estas imágenes son luego procesadas y navegadas virtualmente en un computador en la búsqueda de candidatos a pólipos.

En la literatura es posible encontrar numerosos métodos de detección de pólipos. Entre estos se encuentran aquellos que utilizan análisis de curvatura [YN01][RZB09][HLS*10], líneas de curvatura [ZBB*06][RZB09], ajuste

de superficies [KHK*05], campos de difusión de calor [KA07] y planos de agua [BAHS05], etc. Todas estas técnicas requieren de muchos cálculos, lo que redundaría en elevados tiempos de respuesta.

En este trabajo se paraleliza el método de detección automática de pólipos basado en análisis de curvatura propuesto por Yoshida y Nappy [YN01]. Para la implementación paralela se explotan las bondades de los procesadores gráficos NVIDIA actuales, los cuales son programados mediante CUDA [URL02]. Adicionalmente, ya que el análisis de curvatura suele generar falsos positivos, en este trabajo se propone una estrategia para realizar el descarte mediante el estudio de la forma de un corte longitudinal del candidato a pólipo. Las pruebas realizadas sobre diversas tomografías muestran una excelente reducción del tiempo de respuesta, así como una eficaz disminución de los falsos positivos.

2. Trabajos Previos

En la actualidad existen numerosos métodos para la detección de pólipos en datos volumétricos. Estos métodos en

general suelen implementar dos fases. La primera es la fase de detección de candidatos a pólipos, en donde se detectan las estructuras que pueden ser pólipos. La segunda fase consiste en la eliminación de falsos positivos.

Para detectar los candidatos a pólipos, Yoshida y Nappi [YN01] proponen en el 2001 un método basado en curvaturas que extrae una región gruesa de la pared del colon a fin de eliminar la mayor cantidad de falsos positivos. Este método extrae los atributos geométricos 3D de los pólipos utilizando un índice de forma y un índice de curvatura, agrupa los candidatos a pólipo y posteriormente determina otros atributos que ayudan a reducir los falsos positivos. En el 2008, Huang et al. [HLS*10], investigaron el planteamiento del Frente de Pareto para evaluar y mejorar los algoritmos de región creciente basados en curvatura, comúnmente usados para la detección de candidatos a pólipos.

En el 2005 Kitasaka et al. [KHK*05] proponen detectar pólipos calculando la curvatura en base al ajuste de superficies en lugar de hacer análisis de curvatura basado en la aproximación de la primera y segunda derivada. Usando este método, Kitasaka et al. [KHK*05] reportan menos falsos positivos que los obtenidos aplicando el método basado en curvaturas. Ese mismo año Bitter et al. [BAHS05] plantean el método del plano de agua. El método se basa en simular que la pared del colon se llena de agua a partir de un vértice cualquiera, desde el exterior del colon hacia el interior, hasta detectar el momento a partir del cual el agua se derrama.

En el 2006 Zhao et al. [ZBB*06] introdujeron una técnica para caracterizar y visualizar las líneas de curvatura en la superficie del colon con el fin de detectar pólipos. La idea básica consiste en mostrar la superficie del colon con iluminación y mejorarla desplegando las líneas de curvatura de estas superficies. La principal virtud de esta técnica es que permite mejorar las imágenes de la colonoscopia virtual agregando las líneas de curvatura que permiten al radiólogo caracterizar pólipos fácilmente. Sin embargo, este método aún no ha sido adaptado para permitir la detección automática de los mismos.

En 2007 Konukoglu et al. [KA07] proponen el campo de difusión de calor (HDF - *Heat Diffusion Fields*). El proceso de difusión tiene un efecto de realce ya que el HDF utiliza la estructura de colisión del calor propagado que lleva a la información estructural de toda la superficie de los pólipos; en otras palabras, el HDF hace un resumen de la superficie del colon en un barrido local. El método se basa en la detección y caracterización de los hundimientos en conjunto con la distribución local de los vóxeles en torno a él. Los resultados de este método sugieren una mejora respecto al método de Yoshida [YN01], pero no se pudo hacer una comparación directa ya que trabajaron con datos distintos. A pesar de esto, su principal desventaja es el alto costo computacional requerido.

Los trabajos mencionados anteriormente demuestran que es posible detectar pólipos tanto como protuberancias en una representación explícita (superficie de malla) o como en una representación implícita (niveles de gris) de la pared del colon. Van Ravesteijn et al. [RZB09] presentan en el 2009 un patrón para el reconocimiento de pólipos, que combina los dos enfoques y revela que en cierta manera

son complementarios. Cada enfoque caracteriza diferentes aspectos de los candidatos, los cuales son detectados por medio de la deformación de una representación explícita de la superficie del colon o mediante la modificación de los datos de intensidad que contiene una representación implícita.

Qui et al. [QFY*08] implementan un algoritmo de disección y aplanamiento del colon (*virtual colon flattening*) utilizando CUDA. En nuestro caso, se utiliza CUDA para acelerar el cálculo de las derivadas de primer y segundo orden, las cuales son necesarias para calcular las curvaturas en cada vóxel de las paredes del colon.

3. Método

En este trabajo nos basamos en el trabajo de [YN01] para determinar los candidatos a pólipos. Inicialmente, se segmenta el colon del paciente. Con el colon segmentado, se obtienen los vóxeles que se encuentran sobre la superficie de la pared del colon. Estos vóxeles son clasificados según su curvatura, y agrupados para generar la lista inicial de candidatos a pólipos. Esta lista es refinada, eliminándose los posibles falsos positivos mediante el estudio de la forma del candidato a pólipo en un corte longitudinal.

A continuación se describe en detalle cada una de las etapas del método.

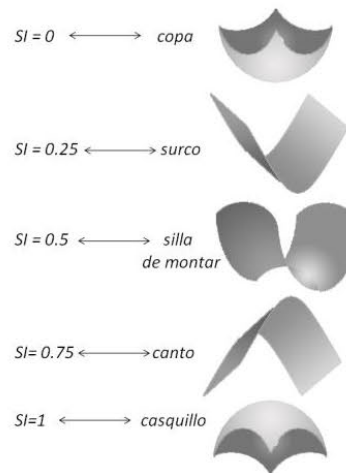


Figura 1: Relación de superficies clásicas con los valores del SI.

3.1 Segmentación del volumen

Para segmentar el volumen, se utilizó un algoritmo básico de región creciente basándose en la intensidad de los vóxeles de aire dentro de la cavidad del colon y los vóxeles de la pared de colon, este proceso se describe en detalle en [COT09]. Una vez marcados los vóxeles de aire, se dilata el subvolumen correspondiente en 2 vóxeles, para entonces obtener los vóxeles de las paredes del colon que se utilizan en las etapas siguientes.

3.2 Clasificación de los vóxeles

Los vóxeles de las paredes del colon son clasificados según el índice de curvatura o *Shape Index (SI)* acorde al trabajo propuesto por Yoshida y Nappi [YN01]. En su tra-

bajo, ellos determinaron que los vóxeles con un SI cercano a 1 podían formar parte de un pólipo debido a la similitud de la forma de un casquillo con la forma terminal de un pólipo (ver Figura 1).

Para el cálculo del SI se utiliza la siguiente fórmula:

$$SI(p) = \frac{1}{2} - \frac{1}{\pi} \arctan \frac{\kappa_1(p) + \kappa_2(p)}{\kappa_1(p) - \kappa_2(p)}$$

donde $\kappa_1(p)$ y $\kappa_2(p)$ son las curvaturas principales en el punto p .

Las curvaturas principales son las curvaturas de máximo valor y de mínimo valor, y para calcularlas utilizamos la fórmula propuesta por Monga y Benayoun [MB92]:

$$k_i = \frac{h^T H h + u^T H u \pm \sqrt{(h^T H h - u^T H u)^2 + 4(h^T H u)^2}}{2}$$

donde

$$\vec{h} = \left(\frac{f_y}{\sqrt{f_x^2 + f_y^2}}, -\frac{f_x}{\sqrt{f_x^2 + f_y^2}}, 0 \right), H = \begin{pmatrix} f_{xx} & f_{xy} & f_{xz} \\ f_{yx} & f_{yy} & f_{yz} \\ f_{zx} & f_{zy} & f_{zz} \end{pmatrix},$$

$$\vec{u} = \left(\frac{f_x f_x}{\sqrt{f_x^2 + f_y^2} \cdot \sqrt{f_x^2 + f_y^2 + f_z^2}}, \frac{f_y f_x}{\sqrt{f_x^2 + f_y^2} \cdot \sqrt{f_x^2 + f_y^2 + f_z^2}}, \frac{\sqrt{f_x^2 + f_y^2}}{\sqrt{f_x^2 + f_y^2 + f_z^2}} \right)$$

con $f_a = \frac{\partial f}{\partial a}$ y $f_{ab} = \frac{\partial^2 f}{\partial a \partial b}$ y f representa la función de la imagen de entrada.

Para obtener el cálculo aproximado de las derivadas parciales de primer y segundo orden en un punto, se debe realizar la convolución en el punto con un kernel de convolución adecuado. Para este trabajo se utilizó el método propuesto por Möller et al. [MMK*98] para el diseño de filtros. A partir del desarrollo de Taylor de la suma de convoluciones y a partir de las restricciones dadas por el grado de la derivada, el grado del error y la continuidad deseada, se obtiene un sistema de ecuaciones. Al resolver este sistema de ecuaciones se obtiene la fórmula del filtro $w(\tau)$ definido en trozos $w_k(\tau)$. Por ejemplo, el filtro de interpolación (derivada 0) con un error de grado 3 y continuidad C_2 es definido en 4 trozos de la siguiente manera:

$$w_{-2}(\tau) = -\tau^5 + \frac{5}{2}\tau^4 - \frac{3}{2}\tau^3$$

$$w_{-1}(\tau) = 3\tau^5 - \frac{15}{2}\tau^4 + \frac{9}{2}\tau^3 + \frac{1}{2}\tau^2 + \frac{1}{2}\tau$$

$$w_0(\tau) = -3\tau^5 + \frac{15}{2}\tau^4 - \frac{9}{2}\tau^3 - \tau^2 + 1$$

$$w_1(\tau) = \tau^5 - \frac{5}{2}\tau^4 + \frac{3}{2}\tau^3 + \frac{1}{2}\tau^2 - \frac{1}{2}\tau$$

Para construir el filtro $w(\tau)$ a partir de sus trozos $w_k(\tau)$, hay que considerar que el soporte de cada trozo $w_k(\tau)$ es $[0,1]$, pero representa el dominio $[-k, -k+1]$ en $w(\tau)$ acorde a (ver Figura 2):

$$w_k(\tau) = w(\tau + k)$$

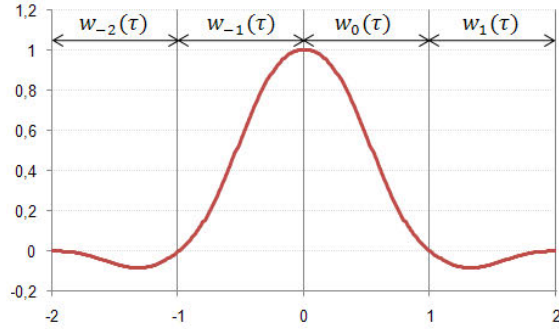


Figura 2: Filtro w definido por 4 trozos.

Para este trabajo se obtuvieron los filtros de interpolación (derivada 0), derivada 1 y derivada 2, cada uno con un grado de error 3 y continuidad C_2 .

A partir de los filtros obtenidos se forman los kernels unidimensionales. Para esto se toman muestras igualmente espaciadas en el dominio del filtro. Las muestras en los bordes del filtro no son tomadas, ya que su valor siempre es 0. Por lo tanto, si se desea obtener un número η de muestras, se debe dividir el dominio del filtro en $\eta + 1$ intervalos de igual tamaño y las muestras se tomarán en los puntos que se encuentran entre 2 intervalos. Cabe destacar que η debe ser impar.

Cada uno de los kernels de convolución tridimensionales W se obtienen a partir de la siguiente fórmula:

$$W[i, j, k] = w_x[i] \cdot w_y[j] \cdot w_z[k]$$

donde w_x , w_y y w_z son los kernels unidimensionales de las derivadas que se desean obtener en la dirección de x , y y z respectivamente. Por ejemplo, para construir el kernel que aproxime el valor de $\frac{\partial^2 f}{\partial x^2}$, w_x tendrá los valores obtenidos para el kernel unidimensional de derivada 2, mientras que w_y y w_z tendrán los valores obtenidos para el kernel unidimensional de interpolación (derivada 0). Para aproximar todas las derivadas parciales requeridas, se construyen 9 kernels de convolución: 3 para las derivadas parciales de primer orden $\frac{\partial f}{\partial x}$, $\frac{\partial f}{\partial y}$ y $\frac{\partial f}{\partial z}$, y otras 6 para las derivadas parciales de segundo orden: $\frac{\partial^2 f}{\partial x^2}$, $\frac{\partial^2 f}{\partial y^2}$, $\frac{\partial^2 f}{\partial z^2}$, $\frac{\partial^2 f}{\partial x \partial y}$, $\frac{\partial^2 f}{\partial x \partial z}$ y $\frac{\partial^2 f}{\partial y \partial z}$.

Una vez obtenidos los kernels de convolución, se obtienen los valores aproximados de las derivadas parciales de primer y segundo orden en los puntos que se encuentran sobre la superficie de la pared del colon. Sea C una arreglo que almacena dicho conjunto de puntos, asumiendo que la posición central de un filtro W es el punto $(0,0,0)$, el valor de la convolución en un punto $p \in C$ viene dado por la fórmula:

$$f^{(W)}[p] = \sum_{q \in D(W)} f[p+q] \cdot W[q]$$

donde $f^{(W)}[p]$ es el valor de la derivada parcial en p aproximado por el kernel de convolución W . Para los valores de $p+q$ que están fuera del volumen se considera que los valores en los bordes se repiten indefinidamente hacia afuera del volumen, para evitar que cortes abruptos en el borde produzcan detecciones erróneas.

Calcular la convolución para todos los puntos $p \in C$ es un proceso que requiere mucho cálculo. Por esta razón, esta

parte del método es implementada utilizando programación paralela en CUDA, mediante la cual se pueden ejecutar múltiples hilos de ejecución en múltiples núcleos de procesamiento CUDA. La idea general es tener múltiples hilos que ejecuten las mismas instrucciones para diferentes datos (ver Figura 3). Si el tiempo de ejecución de los hilos fuese irrestricto, cada hilo podría calcular las distintas convoluciones de un determinado $p \in C$. Sin embargo, existen restricciones para el tiempo de ejecución de los hilos. Por consiguiente, el cálculo asignado a cada hilo es simplemente un término de la sumatoria de una convolución para un $p \in C$. En cada activación de un hilo, se calcula un término, y actualiza la suma parcial de la sumatoria con el término calculado. Para evitar posibles problemas de consistencia, se debe realizar una sincronización de hilos antes de calcular cada término de la sumatoria. Este proceso se repite hasta que todos los términos de la convolución hayan sido calculados.

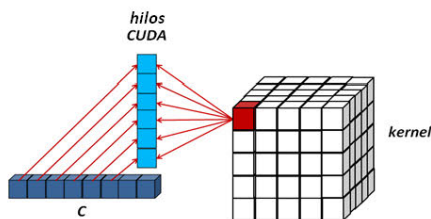


Figura 3: Cada hilo CUDA tiene un punto $p \in C$ y calcula un término de una convolución para dicho punto. Cuando todos los hilos que calculan un término de la convolución finalizan su ejecución, se inicia la ejecución de los hilos que calcularán el siguiente término.

Para reducir la cantidad de sincronizaciones, cada hilo calcula un término de cada una de las 9 convoluciones que deben hacerse en cada punto. Así, cada activación de un hilo realiza 9 cálculos de la forma:

$$\text{sumatoria} = \text{sumatoria} + f[p + q] \cdot W[q]$$

3.3 Selección y agrupación de los candidatos

Por lo general, la parte superior de un pólipo se asemeja a una cúpula o casquillo. Por lo tanto se desea seleccionar los puntos cuyo entorno se asemeje a esta forma. Como se puede apreciar en la Figura 1, si el valor del SI se acerca a 1 la superficie alrededor del punto presenta una forma similar a la deseada. Empíricamente seleccionamos que un punto puede pertenecer a un pólipo si $SI > 0.85$. Los puntos que satisfacen esta condición se almacenan en un arreglo que llamaremos S .

Es de suponer que sobre la parte superior de un pólipo existirán varios puntos seleccionados. Además, es muy probable que si varios de los puntos seleccionados se encuentran cercanos entre sí, todos estén sobre la parte superior del mismo pólipo. Por lo tanto, después de seleccionar los puntos se procede a formar grupos de puntos seleccionados que se encuentren cercanos entre sí. Para realizar esta tarea en forma eficiente en tiempo de respuesta, se creó un arreglo booleano tridimensional B con las mismas dimensiones que el volumen original y se marcan en dicho arreglo las posiciones de los puntos seleccionados.

El algoritmo para formar los grupos a partir de los puntos seleccionados es el siguiente:

1. Se crea un grupo G vacío y una cola auxiliar A vacía.
2. Se mueve un punto cualquiera de la lista S hacia la cola A , y se desmarca como seleccionado en B .
3. Se extrae un punto (que llamaremos p) de la cola A y se agrega al grupo G .
4. Se revisa en una región cúbica alrededor del p en busca de otros puntos seleccionados cercanos (marcados en B).
5. Si existen otros puntos seleccionados en la región cúbica, estos se agregan a A , se borran de S , y se desmarcan en B .
6. Si quedan puntos en A , se vuelve al paso 3
7. Si no quedan puntos en A , el grupo G constituye un candidato a pólipo, y se guarda en un arreglo de grupos.
8. Si quedan puntos en S , se vuelve al paso 1.

Cada uno de los grupos formados de esta manera será considerado como un candidato a pólipo.

3.4 Extracción del corte planar

Luego de haber formado los grupos que serán considerados como candidatos a pólipo, se desea obtener una imagen de un corte plano de cada candidato.

Para cada candidato tenemos un grupo de puntos seleccionados, los cuales suponemos que se encuentran en la parte superior de un pólipo. Si esto es así, al obtener el centro volumétrico P_0 de los puntos seleccionados pertenecientes al candidato, dicho punto central debería encontrarse en la parte interna del pólipo. Por otra parte, si obtenemos un vector N normalizado, que es el promedio de las normales en los puntos seleccionados pertenecientes al candidato, y colocamos su origen en el centro del candidato, podríamos suponer que dicho vector apunta hacia la parte superior del pólipo. Con la normal N y el punto P_0 , podemos construir la ecuación del plano $(P - P_0) \cdot N = 0$, que nos permitirá capturar un corte planar del candidato a pólipo (ver Figura 4), para posteriormente estudiar su forma.

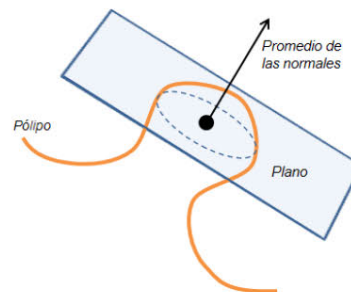


Figura 4: Corte planar del candidato a pólipo.

3.5 Análisis selectivo

Luego de haber obtenido el corte planar se realiza el análisis del mismo para seleccionar fuertes candidatos a pólipo y descartar posibles falsos positivos.

Se espera que la figura central de un candidato a pólipo sea una figura aislada, es decir, una figura de intensidades claras rodeada completamente por una región de intensidades oscuras. Si la región central no está completamente rodeada de una región de intensidades oscuras, la región central de intensidades claras toca el borde de la imagen y difícilmente sea un pólipo. En este caso el candidato es descartado (ver Figura 5). Para detectar las figuras aisladas, se utiliza un algoritmo sencillo de región creciente a partir del centro de la imagen con un umbral determinado.

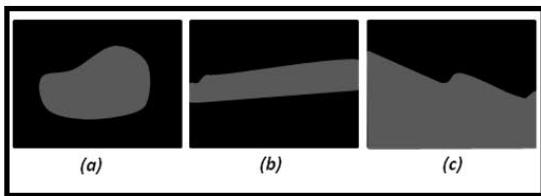


Figura 5: (a) Es fuerte candidato a pólipo. (b) y (c) Se descartan como candidatos porque la región clara toca el borde de la imagen.

Aunque se haya encontrado una figura aislada en el centro del corte, si esta figura es muy alargada, es poco probable que sea un buen candidato. Para descartar dichos candidatos se establece la siguiente relación entre el perímetro p y el área A :

$$R_{(p,A)} = \frac{p^2}{A}$$

Esta relación es un buen indicador de cuán alargada es la figura. Por ejemplo, para un círculo $R_{(p,A)} = \frac{p^2}{A} = \frac{(2\pi r)^2}{\pi r^2} = 4\pi \approx 12,56$, para un cuadrado $R_{(p,A)} = \frac{p^2}{A} = \frac{(4l)^2}{l^2} = 16$ y para un rectángulo con lado mayor $L=9$ y lado menor $l=1$, $R_{(p,A)} = \frac{p^2}{A} = \frac{(2L+2l)^2}{L \cdot l} = \frac{400}{10} = 40$. Entonces, mientras mayor sea $R_{(p,A)}$ más alargada es la figura.

Para conocer el perímetro y el área de la figura encontrada en el corte planar, simplemente se cuentan el número de píxeles en la figura (para el área) y el número de píxeles en el borde de la figura (para el perímetro). Luego, se descartan los candidatos en los cuales $R_{(p,A)}$ sea mayor a un determinado umbral. Los candidatos que no fueron descartados por este criterio son considerados fuertes candidatos a pólipo.

4. Pruebas y Resultados

Se realizaron pruebas de la aplicación con 6 volúmenes distintos, ver Tabla 1. En principio, se realizaron pruebas con diferentes tamaños η para el kernel unidimensional (ver la Sección 3.2) para determinar cuál de los tamaños es el más apropiado. Los tamaños de kernel utilizados oscilaron entre 11 y 29 muestras. Al realizar las pruebas se pudo notar que el tamaño de kernel utilizado afectaba directamente al tamaño de las protuberancias detectadas. Para detectar tanto las protuberancias pequeñas como las gran-

des se decidió combinar los resultados obtenidos con kernels de dos tamaños diferentes. Los tamaños seleccionados fueron 13 y 21.

# Volumen	Dimensiones (X x Y x Z)	Tamaño (MB)
1	512x512x442	221
2	512x512x544	272
3	512x512x548	274
4	512x512x469	234
5	512x512x463	231
6	512x512x426	213

Tabla 1: Características de los volúmenes de prueba.

Para el algoritmo de región creciente utilizado en el análisis selectivo (ver Sección 3.5) el umbral utilizado fue 475 para cinco de los volúmenes, del #2 al #6. Su utilizó 950 para el volumen #1 porque tiene un rango dinámico más amplio y con mayores intensidades que el resto de los volúmenes. También se descartaron los candidatos con un $R_{(p,A)}$ mayor a 30 para todos los volúmenes.

Sólo se contó con el diagnóstico médico para dos de los volúmenes, el volumen #5 y el volumen #6. Estos volúmenes pertenecen a un mismo paciente, uno fue tomado en posición tendida y otro en posición supina. Ambos fueron obtenidos gratuitamente, junto con sus informes médicos, de [URL03]. Para dichos volúmenes, los pólipos diagnosticados fueron detectados en su totalidad por el método propuesto. La Figura 6 muestra capturas de pantalla de los pólipos detectados. Aunque también se obtuvieron algunos falsos positivos, ver Figura 7.

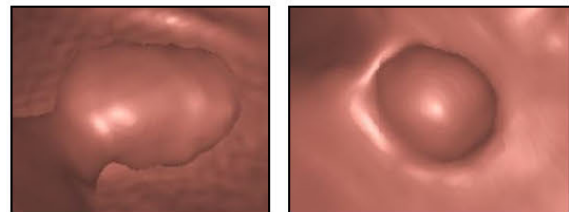


Figura 6: Pólipos exitosamente detectados en los volúmenes #5 y #6.

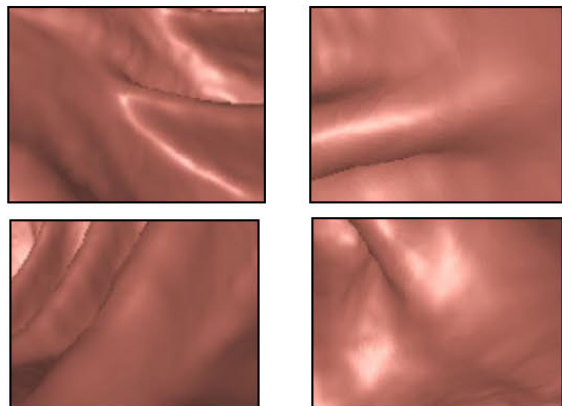


Figura 7: Falsos positivos detectados en los volúmenes #5 y #6.

Pese a que no se contó con el diagnóstico médico para el resto de los volúmenes, igualmente se visualizaron los resultados y se pudo observar que un gran número de las detecciones obtenidas presentaron la forma característica de un pólipo. La Figura 8 muestra algunos resultados.

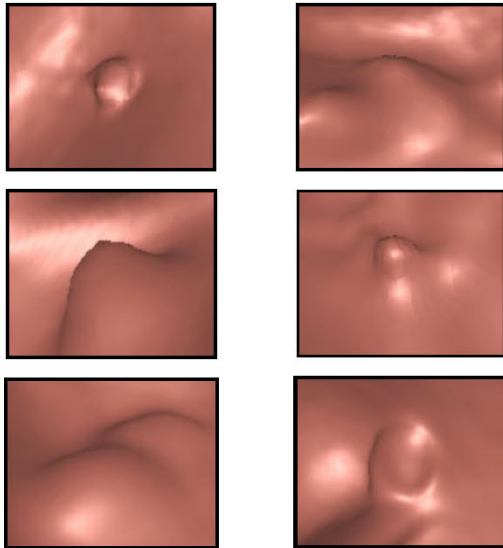


Figura 8: Algunos candidatos a pólipo detectados en los volúmenes del #1 al #4.

El número total de detecciones en cada uno de los volúmenes de prueba, para cada tamaño de kernel, se muestran en la Tabla 2. Esta tabla también muestra el número de falsos positivos detectados por el algoritmo de análisis selectivo presentado en este trabajo. Nótese la reducción dramática de candidatos a pólipos obtenida. Como era de esperarse, en muchos casos para un mismo volumen se detecta la misma región con un kernel de tamaño 13 que con uno de tamaño 21, por lo tanto se calculó el número de candidatos a pólipo combinados, eliminando las repeticiones.

# Volumen	Tamaño del kernel	Detecciones totales	Falsos positivos detectados	Candidatos a pólipo	Candidatos a pólipo (combinados)
1	13	266	266	0	0
	21	268	268	0	
2	13	334	334	0	2
	21	296	294	2	
3	13	760	748	12	18
	21	734	722	12	
4	13	302	300	2	2
	21	249	249	0	
5	13	388	387	1	5
	21	387	382	5	
6	13	404	402	2	3
	21	383	380	3	

Tabla 2: Características de los PCs de prueba.

También se implementó una versión secuencial del algoritmo que se ejecuta en el CPU en su totalidad para comparar su tiempo de ejecución con la versión propuesta. Ambas versiones se ejecutaron en los dos computadores descritos

en la Tabla 3. En la Tabla 4 se muestran los resultados de las pruebas en cada volumen, cada PC de prueba y cada tamaño de kernel. En el primer PC, se logró reducir el tiempo de ejecución entre un 60% y un 74,67% siendo la versión propuesta entre 2,5 y 3,94 veces más rápida. En el segundo PC se logró una reducción del tiempo entre 94,73% y 98,14%, siendo la versión propuesta entre 19 y 53 veces más rápida.

Ahora bien, la jerarquía de hilos en CUDA agrupa a los hilos en bloques, donde cada bloque tiene un espacio de memoria compartida de acceso rápido para los hilos pertenecientes a un mismo bloque, permitiendo así la colaboración entre dichos hilos para incrementar la velocidad de respuesta. Dado que CUDA permite al desarrollador decidir cuántos hilos pertenecerán a un bloque, se realizaron varias pruebas cambiando la cantidad de hilos por bloque. Sin embargo, no se observó un cambio significativo de los tiempos de respuesta entre las diferentes pruebas, con respecto a los reportados en la Tabla 4.

# PC	Procesador	RAM	Sistema Operativo	Tarjeta Gráfica (TG)	# Cores TG	Ancho de Banda TG	Memoria TG
1	Intel Core 2 Quad 2.40 Ghz	3 GB	Windows XP 32 bits	GeForce 8800 GTS	96	64 GB/s	640 MB
2	Intel Core i3 3.07 Ghz	4 GB	Windows 7 64 bits	GeForce GTX 470	448	133.9 GB/s	1280 MB

Tabla 3: Características de los PCs de prueba.

# Volumen	# PC	Tamaño del kernel	Tiempo (segundos)	
			CPU	GPU
1	1	13	110	34
		21	537	136
	2	13	89	3
		21	484	10
2	1	13	102	37
		21	470	123
	2	13	78	4
		21	401	10
3	1	13	247	93
		21	1195	372
	2	13	200	7
		21	1077	20
4	1	13	85	34
		21	409	126
	2	13	69	3
		21	369	10
5	1	13	94	39
		21	484	169
	2	13	76	4
		21	436	11
6	1	13	89	36
		21	436	140
	2	13	72	4
		21	393	10

Tabla 4: Tiempos de ejecución del algoritmo secuencial (CPU) y el algoritmo paralelo en CUDA (GPU), por volumen, PC de prueba y tamaño de kernel.

5. Conclusiones y Trabajos Futuros

En este trabajo se aceleró un algoritmo para la detección automática de candidatos a pólipos aprovechando el paralelismo que proveen las tarjetas gráficas. En específico, la

detección se realizó mediante el análisis de curvatura, en el que se requiere de los cálculos de las derivadas de primer y segundo orden de los vóxeles del volumen, los cuales fueron implementados en CUDA. Los resultados muestran una reducción en el tiempo de cálculo de hasta un 98% al comparar la versión implementada en CUDA con la versión secuencial que corre en procesadores convencionales. También se realizaron varias pruebas del algoritmo cambiando la cantidad de hilos de CUDA por bloque, pero no se observó un cambio significativo del tiempo de respuesta entre las diferentes pruebas.

Se implementó una heurística para eliminar aquellos candidatos a pólipos que no presentan forma de pólipos. La misma consistió en analizar la forma 2D de un corte planar del candidato a pólipo en función de su perímetro y área. La cantidad de falsos positivos eliminada con esta heurística es bastante significativa. Los candidatos a pólipos arrojados finalmente por el programa parecen ser, en su mayoría, fuertes candidatos a pólipos (según su forma). Sin embargo, no se pudo establecer un porcentaje de sensibilidad del programa ya que no todos los volúmenes de prueba fueron analizados por médicos especialistas.

Como trabajo futuro se propone realizar más pruebas y analizar los resultados con médicos especialistas para así determinar un porcentaje concreto de sensibilidad y una tasa de falsos positivos. También sería de interés realizar pruebas con pólipos planos para determinar si el programa los detecta.

Se sugiere probar otros métodos de segmentación del colon, así como implementar algoritmos para eliminar ciertos residuos presentes en el colon que estorban en el proceso de detección. También se podría aplicar un filtro para detección de bordes (e.g. Sobel) al corte planar antes de hacer el análisis de la forma y comparar los resultados.

Se puede aplicar el análisis de la forma del corte planar a otros métodos de detección (por ejemplo, el desarrollado por Kitasaka [KHK*05]). Además, se podría utilizar el corte planar para la segmentación de los pólipos. La idea es mover el centro del corte en la dirección de la normal en ambos sentidos y obtener varios cortes mientras la condición del análisis selectivo se cumpla y seleccionar los vóxeles correspondientes a las muestras de la figura central del corte.

Posterior a la segmentación de los pólipos, también podría realizarse un estudio para la clasificación y caracterización de los candidatos a pólipos utilizando descriptores (como por ejemplo la covarianza) que serían utilizados por una *Máquina de Soporte Vectorial (SVM – Support Vector Machine)* para una fase adicional de reducción de falsos positivos.

6. Agradecimientos

Este trabajo ha sido financiado por el Consejo de Desarrollo Científico y Humanístico de la Universidad Central de Venezuela (CDCH-UCV), bajo el proyecto PG 03-00-6515-2006.

Referencias

- [BAHS05] BITTER I., ASLAM B., HUANG A., SUMMERS R.: Candidate determination for computer aided detection of colon polyps. En *Medical Imaging 2005: Physiology, Function, and Structure from Medical Images*. AMINI A., MANDUCA A. (Eds.). pp 804–809. 2005.
- [COT09] COTO E., GRIMM S.: Improved data processing for virtual colonoscopy. *Proceedings del IV Iberoamerican Symposium in Computer Graphics (SIACG 2009)*. O. RODRIGUEZ ET AL. (Eds.). pp. 171–179. Junio, 2009.
- [HLS*10] HUANG, A., JIANG LI, RONALD M. SUMMERS, NICHOLAS PETRICK, AMY K. HARA.: Improving polyp detection algorithms for CT colonography: Pareto front approach. En *Pattern Recognition Letters*, vol. 31, num. 11, pp. 1461–1469. Estados Unidos. 2010.
- [KHK*05] KITASAKA T., HAYASHI Y., KIMURA T., MORI K., SUENADA Y., TORIWAKI J.: Detection of colonic polyps from 3D abdominal CT images by surface fitting. En *CARS 2005: Computer Assisted Radiology and Surgery*, pp. 1151–1156. 2005.
- [KA07] KONUKOGLU E., ACAR B.: HDF: Heat diffusion fields for polyp detection in CT colonography. En *Signal Processing*, vol. 87, num. 10, pp. 2407–2416, ISSN 0165-1684. 2007.
- [MMK*98] TORSTEN M., KLAUS M., YAIR K., RAGHU M., RONI Y.: Design of accurate and smooth filters for function and derivative reconstruction. En *IEEE Symposium on Volume Visualization*, pp. 143–151. Estados Unidos. 1998.
- [MB92] OLIVIER M., SERGE B.: Using partial derivatives of 3D images to extract typical surface features. *Proceedings of the Third Annual Conference of AI, Simulation and Planning in High Autonomy Systems: Integrating Perception, Planning and Action*. pp. 225-236. Francia. Julio, 1992.
- [QFY*08] QIU F., FAN Z., YIN X., KAUFMAN A., GU X. D.: Colon flattening with discrete ricci flow. *Proceedings de MICCAI 2008 workshop: Computational and Visualization Challenges in the New Era of Virtual Colonoscopy*. pp. 97–102. 2008.
- [URL01] ANUARIO DE MORTALIDAD 2008. *Ministerio del Poder Popular para la Salud*. República Bolivariana de Venezuela. Tomado (2011, Octubre 25) de http://www.bvs.org.ve/anuario/anuario_2008.pdf.

- [URL02] CUDA HOME. *NVIDIA Corporation*, 2011. Tomado (2011, Diciembre 19) de http://www.nvidia.com/object/cuda_home_new.html
- [URL03] VOLVIS.ORG. Tomado (2006, Marzo 29) de <http://www.volvis.org>.
- [YN01] YOSHIDA, H., J. NAPPI: Three-dimensional computer-aided diagnosis scheme for detection of colonic polyps. En *IEEE Transactions on Medical Imaging*, vol. 20, num. 12, pp. 1261–1274. 2001.
- [ZBB*06] ZHAO L., BOTHA C., BESCOS J., TRUYEN R., Vos F., POST F.: Lines of curvature for polyp detection in virtual colonoscopy. En *IEEE Transactions on Visualization and Computer Graphics*, vol.12, num 5, pp. 885–892. 2006.