ppi 201502ZU4659
Esta publicación científica en formato digital es continuidad de la revista impresa
ISSN 0254-0770 / Depósito legal pp 197802ZU38

Hacia los 130 años de creación de la Universidad del Zulia

"Buscar la verdad y afianzar los valores transcendentales", misión de las universidades en su artículo primero, inspirado en los principios humanísticos. Ley de Universidades 8 de septiembre de 1970.

REVISTA TÉCNICA
DE LA FACULTAD DE INGENIERÍA

Una Revista Internacional Arbitrada que está indizada en las publicaciones de referencia y comentarios:

- SCOPUS
- Compendex
- Chemical Abstracts
- Metal Abstracts
- World Aluminium Abstracts
- Mathematical Reviews
- Petroleum Abstracts
- Current Mathematical Publications
- MathSci
- Revencyt
- Materials Information
- Periódica
- Actualidad Iberoamericana

UNIVERSIDAD DEL ZULIA

REVISTA TÉCNICA
DE LA FACULTAD DE INGENIERÍA

VOLUMEN ESPECIAL 2020 No.1

Parallel segmentation of DICOM X-Ray images with mobile agents

Washington Ramírez Montalvan[1,2]*, Anabely Fernández Toapanta[2], Fausto Freire Carrera[3], Jesús Arámburo Lizárraga[1]

[1]Universidad de Guadalajara,C.P. 45100, Zapopan, Jalisco, México.
[2]Departamiento de Ciencias de la Computación, Universidad Politécnica Salesiana,C.P. 70606, Quito, Ecuador.
[3]Facultad Ciencias de la Ingeniería e Industrias, Universidad UTE, Quito, Ecuador
*Autor de Contacto:wramirez@ups.edu.ec
https://doi.org/10.22209/rt.ve2020a06
Recepción: 31/10/2019 | Aceptación: 31/01/2020 | Publicación: 01/03/2020

Abstract

The development of technology has transformed the clinical practice, and nowadays, it is common the use of applications on mobile devices for healthcare; in the detection of breast cancer medical digital images are a powerful tool for diagnosis, however, their size drawbacks their display resolution on mobile devices, to improve the performance of the devices parallelization and specialization of hardware is used. In this work, we present a proposal to resolve this problem through the implementation of a parallel application on CPUs and GPUs, which uses agent`s technology and was developed in JADE for Android platforms, the assessment of the application was carried out with 53 INbreas images, of which 25 images were of masses and 28 images of microcalcifications, after the operation of pre-and post-processing the performance improved in 6.81%.

Keywords: digital X-Ray image; mobile devices; agents; mathematical morphology; watershed.

Segmentación paralelizada de imágenes por Rayos X DICOM conagentes móviles

Resumen

El desarrollo de la tecnología ha transformado la práctica clínica, y hoy en día, es común el uso de aplicaciones en dispositivos móviles para la salud; en la detección del cáncer de mama en medicina las imágenes digitales son una herramienta poderosa para el diagnóstico, sin embargo, el tamaño de las mismas genera inconvenientes al momento de ser ostradas en dispositivos móviles, para mejorar el rendimiento de los dispositivos se utiliza la paralelización y especialización de hardware. En este trabajo, se presenta una propuesta para resolver este problema mediante la implementación de una aplicación paralela en CPU y GPU, que utiliza la tecnología de agentes y se desarrolló en JADE para la plataforma Android, la evaluación de la aplicación se realizó con 53 imágenes INbreas, de las cuales 25 imágenes de masas y 28 imágenes de microcalcificaciones, al realizar las operaciones de pre y post procesamiento el rendimiento mejoró en 6,81%.

Palabras clave: imagendeRayos X digital;Dispositivos móviles;Agentes; Morfología matemática;Watershed.

Introducción

El desarrollo de la tecnología ha transformado la práctica clínica, y en la actualidad es común el uso de aplicaciones de salud en dispositivos móviles. Visualizar las imágenes digitales médicas con estos dispositivos, se han convertido en una potente herramienta de ayuda, para el diagnóstico médico del cáncer de mama, pero las imágenes son de gran tamaño y gestionar grandes cantidades de datos, utilizando métodos tradicionales [1] presenta varios inconvenientes como: lentitud en consultas de los expedientes, perdida de archivos, imágenes defectuosas, entre otros. Para solucionar estos problemas existen técnicas que permiten mejorar el rendimiento y la optimización de los recursos [2] como: Enfoque de interpolación no uniforme, Enfoque de dominio de frecuencia, Enfoque de reconstrucción SR (Súper Resolución) regularizado.

Históricamente para mejorar el tiempo de los procesos y aumentar el rendimiento de un dispositivo informático se trabajaba en la mejora de los semiconductores. Actualmente se utilizan dos conceptos para mejorar el rendimiento de dispositivos, la paralelización, y la especialización[3].

Asimismo, desde hace varios años, el paradigma de programación orientada a agentes se considera como una de las tecnologías más innovadoras para el desarrollo de sistemas de software distribuidos [4]. La tecnología de agentes en la ingeniería de software es un campo que se está investigando para poder brindar estabilidad, escalabilidad y rendimiento a diferentes áreas de Tecnología de Información (TI).

Se desarrolló un método para reducir regiones obtenidas después de la segmentación, mejorando y resaltando áreas sospechosas con malignidad. El método propuesto detecta las estructuras curvilíneas aplicando Sobel con una máscara de 11x11, luego se calcula la gradiente de la imagen utilizando Canny, posteriormente se realiza el cálculo de gradiente proporcionando los puntos mínimos en la región de la imagen, y para reducir la sobre-segmentación se aplica la máscara encontrada con la gradiente de la imagen. Finalmente se aplica Watershed (WS) en la imagen resultante. Este método se utilizó con imágenes mamográficas, obteniendo buenos resultados[5].

Utilizando un histograma local, morfología matemática para escala de grises y WS, se mejora y suaviza la imagen, y posteriormente aplican la gradiente que resulta en regiones segmentadas. Se utilizó la dataset MIAS, 8 imágenes con masas circunscritas y 8 imágenes con masas espiculadas. Sin embargo, para obtener buenos resultados, las masas deben estar bien definidas[6].

Se han propuesto técnicas de administración de energía, desarrolladas en programación paralela OpenMP para dispositivos móviles, identificando rutinas de biblioteca invocadas y uso dinámico de CPU dependiendo del tamaño de la aplicación. Como resultado al apagar los núcleos innecesarios disminuyo el consumo de energía en un 18% en comparación con otros métodos de administración de energía convencionales[7].

Para realizar una simulación de flujo se presentaron dos patrones de diseño paralelos basados en una cuadrícula, denominado método de Lattice-Boltzmann (LBM) en un dispositivo móvil con Android. Los diseños fueron, memoria compartida basada en tareas y memoria distribuida basada en hilos[8].

El Median Filter se usa normalmente para reducir el ruido en una imagen [9], esta técnica considera cada píxel de una máscara, de la cual el centro es el punto semilla y se busca a los vecinos cercanos (pixeles) para decidir si son representativos o no de su entorno [10].

Umbral Otsu selecciona un valor de umbral de nivel de gris óptimo para separar los objetos del fondo de una imagen en función de su distribución de nivel de gris [11]. El histograma de nivel de gris de una imagen generalmente se considera como una herramienta eficiente para el desarrollo de algoritmos de umbral de imagen. La creación de umbrales crea imágenes binarias a partir de imágenes de nivel de gris al convertir todos los píxeles debajo de algún umbral a cero y todos los píxeles sobre ese umbral a uno. El umbral de Otsua diferencia de K Means, calcula histogramas a nivel de grises, lo cual es muy útil al momento de utilizar en imágenes médicas que contengan ruido de sal y pimienta [12].

La morfología matemática es un conjunto de operaciones matemáticas de procesamiento de imágenes basadas en geometría. Las operaciones morfológicas transforma la forma o estructura de los objetos de una imagen, proporcionando información útil [13], las cuales sirven para el procesamiento de imágenes, análisis, codificación y compresión industrial automatizada, análisis de texturas y espacios de escala [14]. La técnica maneja ciertas operaciones morfológicas como: Dilatation, Erosion, Opening, Closing [13, 14]. La morfología matemática extrae estructuras geométricas de formas conocidas tales como: círculo, cuadrado, rectángulo, cruz. La forma y el tamaño se escogen dependiendo a la extracción de forma que se desea obtener [15].

Watershed es un algoritmo propuesto por [16], y fundamentado en el concepto de inmersión. La inmersión utiliza la siguiente analogía: se idealiza una superficie con varios agujeros. Esta superficie se sumerge lentamente en un lago a partir del mínimo valor de intensidad (agujero

más bajo), el agua llenará progresivamente las diferentes cuencas de captación de la imagen (superficie) [17]. Conceptualmente, el algoritmo construye una represa para evitar una situación en la que el agua proviene de dos o más mínimos locales diferentes. El proceso de inundación continúa hasta que las aguas de cuencas se unan, formando líneas de unión que representarán las fronteras de regiones homogéneas, y constituyen el resultado de la segmentación [18]. Sin embargo, esta técnica produce sobre segmentación, para resolver este problema utiliza técnicas de pre y pos procesamiento. Entre los métodos de pre-procesamiento, se propone la reducción de ruido utilizando Median Filter, morfología de color y otros filtros de suavizado, post-procesamiento utiliza técnicas basadas en histogramas para encontrar nuevas regiones [17]. Para implementar Watershed existen tres métodos:

Transformada de Distancia (DT): consiste en la distancia (mascara) y busca en la máscara los pixeles diferentes de cero más cercanos [18, 19]. Hay muchas maneras de definir la distancia entre dos píxeles $[i_1,j_1]$ y $[i_2,j_2]$ en una imagen [18] y son: Euclidiana, Cityblock, Chessboard (Ecuaciones 1,2 y 3).

$$d_{Euclidean}([i_1,j_1],[i_2,j_2]) = \sqrt{(i_1-i_2)^2+(j_1-j_2)^2} \quad (1)$$

$$d_{Cityblock}([i_1,j_1],[i_2,j_2]) = |i_1-i_2|+|j_1-j_2| \quad (2)$$

$$d_{Chessboard}([i_1,j_1],[i_2,j_2]) = max|i_1-i_2|,|j_1-j_2| \quad (3)$$

Método de Gradiente: se utiliza para procesar una imagen en escala de grises, como resultado de este método se obtiene valores de pixel alto a lo largo de los bordes y valores bajos en los objetos de la imagen, produciéndose sobre segmentación [18].

Acercamiento Controlado por marcador: un marcador es un componente conectado que pertenece a la imagen, se utiliza para modificar el degradado de la imagen, es un método robusto y flexible para la segmentación de objetos con contornos cerrados, donde los límites se expresan como crestas. Los marcadores se colocan dentro de un objeto de interés; los marcadores internos se asocian con objetos de interés y los marcadores externos se asocian con el fondo. Después de la segmentación, los límites de las regiones de la cuenca hidrográfica se organizan en las crestas deseadas, separando así cada objeto de sus vecinos [18].

Transforma el histograma (HS) de una imagen es un histograma específico para lograr rangos de nivel de gris resaltados [20], y utiliza la ecuación 4. Es decir primero mejora la imagen en escala de grises y luego los niveles de gris se vuelven a correlacionar con los niveles de gris existentes en el histograma [21].

Dónde:

$$V_k = C(Z_k) = \sum_{i=0}^{k} P(r_i) = S_k \quad (4)$$

k=0, 1, 2… L-1, $P(r_i)$ = pixel en escala de grises.

OpenMP es una API (ApplicationProgramming Interface), que representa a un modelo de paralelismo (incremental) multihilo (estático y/o dinámico), portable e independiente del hardware, para bucles y secciones en sistemas de memoria compartida, mediante sincronización y comunicación utilizando variables compartidas. Está conformado de componentes como, directivas, funciones de librería y variables de entorno [22].

JADE (Java AgentDevelopment Framework), es una herramienta utilizada para el desarrollo de sistemas multi-agentes [23]. Un agente es un programa autónomo con características propias [24] y tiene varios estados como: Iniciado, Activo , Suspendido, Espera, Eliminado, Tránsito, Copiar y Gone. JADE-LEAP es una versión modificada y liviana de JADE para ejecutarse en dispositivos con recursos limitados, como dispositivos móviles, y utiliza algunos estados de JADE [25].

En este trabajo se presenta una propuesta para resolver este problema mediante la implementación de una aplicación paralelizada sobre CPU´s y GPU´s, que utiliza la tecnología de agentes y fue desarrollada en JADE para la plataforma Android.

Materiales y Métodos

Para realizar el aplicativo Jade-Leap para segmentación se utilizó en siguienteprocedimiento:

Inicia el contenedor principal de JADE, y el contenedor secundario con el agente en el dispositivo móvil. La opción abrir (Figura 1B) envía un archivo zip con las imágenes que serán analizadas en el dispositivo móvil. El agente aSegDICOM está localizado en el contenedor secundario JADE-LEAP y los estados abrir, enlazar y copiar se encuentran en tránsito, la clase descomprime se encuentra en estado suspendido y espera, que el archivo zip se encuentre en el repositorio secundario. El agente aSegDICOM es el encargado de los procesos de segmentación (Fig.1C y 1D) de imágenes.

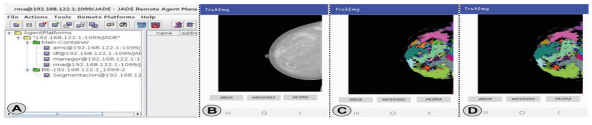

Figura 1. Escenario de Jade, B) App en el dispositivo, C) pre-procesamiento de imagen con 626 segmentos, D) post-procesamiento de imagen con 482 segmentos

Los pasos que contempla la segmentación son: Pre-procesamiento, Watershed, Post-procesamiento, Paralelización y Rendimiento.

Pre-Procesamiento:

La entrada y salida de datos de imágenes se realiza con OpenCV, que es una librería de visión artificial escrita en C/C++, la cual proporcionando un mejor manejo imágenes en dispositivos móviles [26]. A continuación se transforma la imagen en escala de grises Esta transformación resulta útil para varias tareas de análisis de imágenes [27]. La imágenes pueden contener ruido o degradación de la misma, la cual produce pérdida de calidad [28]. Para solventar este problema se utilizó Median Filter con una máscara de 3x3 para la eliminación de ruido [5, 14]. Para separar el área de estudio y el fondo de la imagen se utilizó Otsuy calcular el umbral automático de la imagen en escala de grises; transformando los datos de grises a binario [29]. Por otro lado para aplicar Morfología matemática previamente se identificó la forma de los segmentos, para este caso se utilizó una elipse, y luego se aplicó Opening y Closing con tres iteraciones [14].

Watershed (WS):

Para aplicar WS previamente se aplicó la transformación de distancias de ocho pixeles técnica utilizada para el cálculo de segmentos[19]. Finalmente se aplicó WS a la imagen mejorada con Median Filter, para poder visualizar de mejor manera los segmentos, y también la coloración de cada segmento encontrado [30].

Post-Procesamiento:

Después de aplicar WS a las imágenes segmentadas es posible mejorar el resultado [5], aplicando HS, la cual compara el segmento obtenido con todos los segmentos de la imagen; busca histogramas similares y tiene en cuenta la distancia de los segmentos, el uso de esta técnica reduce el número de segmentos encontrados [21].

Paralelización:

Para optimizar la aplicaciónmóvil se realiza paralelización de procesos en los bucles con OpenMP. La estructura de la paralelización es la siguiente: #pragmaompparallel, es la etiqueta que inicia la paralelización en la aplicación, el proceso se realizó para CPU y GPU, por lo que se utilizó la siguiente etiqueta #pragmaomp target teamsdistributeparallelforque corresponde a GPU. Ya que WS es un cálculo que necesita la mayoría de los recursos se utiliza la etiqueta #critical[31].

Rendimiento:

Para evaluar el rendimiento de las técnicas se utilizaron métricas tales como Mean Square Error (MSE), y PeakSignaltoNoise Ratio (PSNR).

Resultados y discusión

Los resultados del experimento se evaluaron en dos escenarios, ejecución secuencial utilizando CPU's, y ejecución en paralelo utilizando CPU's y GPU's.

En primera instancia se evaluó la carga y transformación a escala de grises, y posteriormente las técnicas de segmentación.

En el experimento se utilizaron 53 imágenes de INbreast, organizadas de la siguiente manera: 25 masas y 28 Microcalcificaciones (MCs), de las cuales para las masas: 13 imágenes con vista craneocaudal (CC) y 12 con vista en medio lateral oblicua (MLO); para Microcalcificaciones (MCs) 14 imágenes con vista CC y 14 con vista MLO. El tamaño de las imágenes para masas y MCs varía entre 3328x2560 y 4084x3328 pixeles. Para ejecutar los escenarios se utilizó un dispositivo Android con procesador Exynos 9610, con una CPU de 8 Cores y una GPU Mali-G72 con 18 Cores, y memoria RAM de 4GB.

Para el primer escenario se evaluó procesos y técnicas de segmentación desarrolladas en C++ e implementadas en secuencia, y se extrajoel tiempo de uso del CPU (T-CPU) y tiempo de ejecución (T-E) en segundos de cada técnica. Los resultados que se describen a continuación corresponden a la ejecución de 53 imágenes.

Para la transformación a escala de grises 31,21% de T-CPU y 19,32% de T-E. Para Otsu 17,39% de T-CPU y 9,82% de T-E. Para Morfología Matemática 31,70% de T-CPU y 21,42%. Para Distancia 4,33% de T-CPU y 21,48% de T-E. Para Watershed 7,06% de T-CPU y 16,54% T-E. Para Histograma 8,31% de T-CPU y 11,37% de T-E.

En la Tabla 1, se resume y organizan los resultados, de imágenes (MLO y CC) con senos izquierdos y derechos.

Para el segundo escenario se evaluaron las técnicas de segmentación desarrolladas en C/C++ e implementadas en paralelo, y se extrae el tiempo de uso del CPU (T-CPU), tiempo de uso del GPU (T-GPU) y tiempo de ejecución (T-E) de cada técnica en segundos. Para la transformación a escala de grises 1,38% de T-CPU, 2,77% de T-GPU, 11,64 de T-E, 7% CORES y 11% HILOS. Para Otsu 0,41% de T-CPU, 0,91% de T-GPU, 6,06 de T-E, 11% CORES y 15% HILOS. Para Morfología Matemática 19,26% de T-CPU, 8.01 de T-GPU, 12,30% T-E, 18% CORES y 15% HILOS. Para Distancia 14,21% de T-CPU, 25,70% de T-GPU, 20.02% de T-E, 16% CORES y 17% HILOS. Para Watershed 38,10% de T-CPU, 8,01% T-GPU, 25,96% T-E, 24% CORES y 23% HILOS. Para Histograma 26,64% de T-CPU, 54,59% de T-GPU, 24,12% T-E, 24% CORES y 19% HILOS.

En las Tablas 2 y 3, se resume y organizan los resultados, de imágenes (MLO y CC) con senos izquierdos y derechos.

Tabla 1. Resumen de procesos y técnicas en secuencia.

Detalle		#img[a]	Escala Gris (s)		Otsu (s)		Morfología (s)		Histogramas (s)		Distancia +Watershed (s)			
			CPU	T	CPU	T	CPU	T	CPU	T	CPU	T	CPU	T
Masas	CC	06 L	273,87	487,82	262,97	298,27	256,95	450,01	235,82	388,72	234,35	450,01	247,52	976,06
		07 R	559,55	451,76	349,40	204,34	435,06	626,32	16,90	178,24	6,97	626,32	28,38	248,64
	MLO	05 L	635,88	616,81	290,28	228,44	793,72	654,85	151,08	431,36	5,86	654,85	25,88	256,66
		07 R	557,08	463,43	227,29	295,21	572,46	513,68	21,22	190,01	7,54	513,68	28,60	248,45
MCs	CC	09 L	0,94	0,54	0,34	0,33	0,86	0,50	5,32	0,53	6,89	0,68	28,08	0,32
		05 R	1,22	0,50	0,45	0,34	1,07	0,46	33,42	0,28	5,51	0,69	42,33	0,33
	MLO	08 L	0,90	0,59	0,33	0,31	0,77	0,62	6,63	0,66	6,92	0,68	22,96	0,31
		06 R	1,03	0,45	0,38	0,36	1,03	0,46	18,04	0,27	7,73	0,77	35,30	0,35

[a] L = Izquierda. R = Derecho. T = Tiempo de ejecución.

Tabla 2. Resumen de Técnicas en Paralelo.

Detalle		#img[a] CPU	Escala Gris (s)a					Otsu (s)					Morfología (s)				
			CPU	GPU	T	C	H	CPU	GPU	T	C	H	CPU	GPU	T	C	H
Masas	CC	06 L	1,87	1,02	0,40	2	17	0,21	0,25	0,21	2	26	1,46	1,30	0,52	4	26
		07 R	0,68	0,52	0,38	1	18	0,25	0,25	0,18	3	26	85,50	10,12	0,44	4	26
	MLO	05 L	0,56	0,75	0,54	1	18	0,21	0,30	0,20	2	25	1,47	1,21	0,57	5	34
		07 R	0,68	0,50	0,46	1	18	0,24	0,15	0,18	2	27	1,53	1,32	0,50	5	33
MCs	CC	09 L	0,67	0,56	0,38	5	28	0,24	0,18	0,23	8	36	0,60	0,10	0,34	11	35
		05 R	0,84	0,55	0,36	4	27	0,32	0,20	0,23	8	37	0,78	0,26	0,33	11	35
	MLO	08 L	0,64	0,66	0,42	4	28	0,23	0,19	0,22	8	37	0,53	0,28	0,44	10	35
		06 R	0,72	0,66	0,31	4	28	0,27	0,20	0,26	8	37	0,73	0,50	0,32	12	35

[a] L = Izquierda. R = Derecho. T = Tiempo de ejecución. C= Cores. H=Hilos

Tabla 3. Continuación: Resumen de Técnicas en Paralelo

Detalle		#img[a]	Distancia +Watershed (s)										Histogramas (s)				
			CPU	GPU	T	C	H	CPU	GPU	T	C	H	CPU	GPU	T	C	H
Masas	CC	07 R	35,21	25,49	2,39	5	32	36,11	1,30	5,83	11	45	54,5	40,62	1,07	11	45
		05 L	4,95	3,50	0,43	5	34	19,75	10,12	0,20	11	46	10,12	9,52	0,22	11	46
	MLO	07 R	3,94	4,20	0,41	5	33	17,85	1,21	0,17	10	46	4,81	2,31	1,38	10	46
		09 L	5,32	2,30	0,45	5	34	19,67	1,32	0,19	12	45	11,54	9,35	2,95	12	45
MCs	CC	05 R	4,79	3,20	0,48	8	38	19,99	0,10	0,22	9	51	4,13	2,25	0,40	9	35
		08 L	3,87	3,24	0,46	9	39	28,84	0,26	0,22	10	50	23,64	25,32	0,20	10	36
	MLO	06 R	4,81	3,24	0,47	8	38	16,26	0,28	0,21	9	53	6,74	2,23	0,38	9	36
		07 R	5,43	3,24	0,54	8	38	24,70	0,50	0,25	9	52	12,61	11,20	0,19	10	36

[a] L = Izquierda. R = Derecho. T = Tiempo de ejecución. C= Cores. H=Hilos

En las Figura 2 y Figura 3, se puede observar los resultados del experimento durante la implementación realizada en paralelo empleando OpenMP; disminuye considerablemente el tiempo de procesamiento y de ejecución con respecto a las implementaciones secuenciales, lo que se traduce en la posibilidad de procesar imágenes DICOM de gran tamaño en menor tiempo. El tiempo paralelo es comparado con el tiempo secuencial, donde el procesamiento en paralelo disminuye considerablemente, lo que representa una mejora del 81,35%

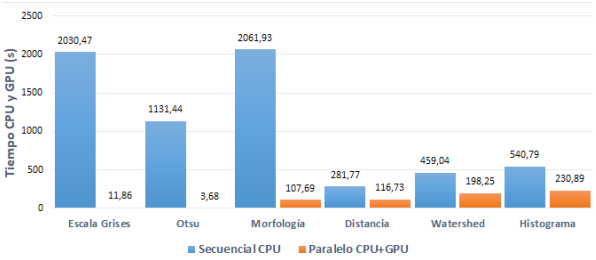

Figura 2. Tiempo de uso de CPUs con procesos en secuencia y uso de CPUs con GPUs con procesos en paralelo.

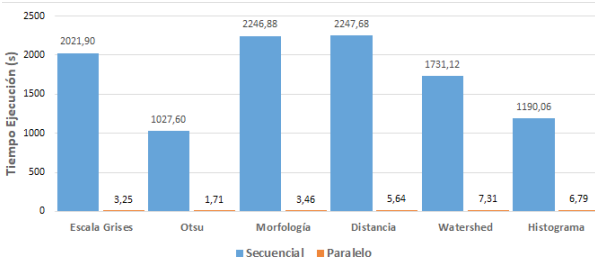

Figura 3. Tiempo de ejecución de técnicas entre procesos secuenciales con CPUs y procesos en paralelo con CPUs y GPUs

En la fase pre-procesamiento se utilizan técnicas como Otsu y Morfología Matemática, y a continuación se aplica la técnica de Watershed, y para la fase de post-procesamiento los resultados obtenidos luego de aplicar Watershed se aplica Histogramas. Los resultados de segmentación obtenidos en pre-procesamiento para masas fueron 11113 segmentos y para MCs 17742 segmentos; para post-procesamiento, se generan 9416 segmentos para masas, y para MCs se generan 15760 segmentos. En consecuencia, al aplicar la fase de pos-procesamiento se obtiene una mejora del 6.81% (Figura 4). Asimismo, se evidenció un incremento de 12% en el uso de memora RAM cuando el proceso se realiza en paralelo. En la Tabla 4, se resume y organizan los resultados, por conjunto de imágenes de masas y MCs con vista MLO y CC.

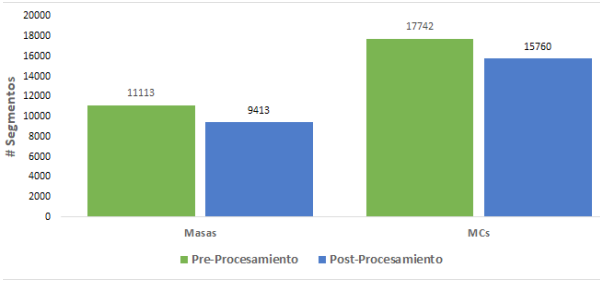

Figura 4. Segmentos generados utilizando Watershed en imágenes con masas y MCs

Tabla 4. Segmentos para Pre-Procesamiento y Post-Procesamiento

Detalle		#img	# Segmentos		Rendimiento RAM (Mb)	
			PreProcesamiento	PostProcesamiento	Secuencial	Paralelo
Masas	CC	06 L	2166	2084	311,15	348,33
		07 R	3658	3014	297,94	339,65
	MLO	05 L	2723	2308	290,76	322,82
		07 R	2566	2007	370,92	390,32
MCs	CC	09 L	5005	4741	388,30	434,89
		05 R	2675	1989	393,78	433,16
	MLO	08 L	4796	4538	357,55	396,88
		06 R	5266	4492	404,82	445,30

[a] L = Izquierda. R = Derecho. T = Tiempo de ejecución. C= Cores. H=Hilos

El rendimiento promedio global de las técnicas se agrupa en pre-procesamiento y post-procesamiento. Para pre-procesamiento se evalúa Median Filter con el conjunto de imágenes. Para masas, la comparación entre la imagen original y la imagen aplicada los resultados con MSE fueron 0.24 y con PSNR fue 55.34dB. Para las MCs, la comparación entre la imagen original y la imagen aplicada los resultados con MSE fueron 126, y PSNR con 52.27dB. Para post-procesamiento se evalúa Watershed y se utiliza G-GU, G-GC, D-PE y D-ND para masas y MCs y los resultados se resumen en la Tabla 4.

Tabla 5. Resultado de Métricas para Técnicas de Segmentación

Detalle		#img	Img (FxC)	Median Filter		Wathershed			
				MSE	PSNR(dB)	G-GU	G-GC	D-PE	D-ND
Masas	CC	11	3328×2560	0,35	53,04	0,897	0,989	0,578	0,532
		02	4084×3328	0,24	55,34	0,846	0,987	0,202	0,402
	MLO	11	3328×2560	0,88	50,78	0,821	0,957	0,345	0,603
		01	4084×3328	0,70	49,68	0,800	0,909	0,201	0,503
MCs	CC	14	3328×2560	158,01	51,47	0,808	0,901	0,301	0,545
		06	4084×3328	126,00	52,27	1,000	0,999	0,231	0,654
	MLO	15	3328×2560	185,44	51,32	0,825	0,997	0,211	0,578
		05	4084×3328	229,14	48,84	0,852	0,989	0,201	0,575

[a] L = Izquierda. R = Derecho. T = Tiempo de ejecución. C= Cores. H=Hilos

Conclusiones

Para el desarrollo de la aplicación móvil se implementó el paradigma fundamentado en agentes y se utilizó el Framework de JADE sobre el sistema operativo Android, y las técnicas se desarrollaron en C/C++, por tal motivo para relacionar Java y C/C++ se utilizó JNI.

Se evidencia la necesidad de aplicar técnicas de pre y post procesamiento para reducir el número de segmentos generados por la aplicación de WS. En consecuencia, al aplicar la fase de pre-procesamiento (53,41%) y pos-procesamiento (46,59%) se obtiene una mejora del 6,81%.

Se redujó el alto consumo computacional de WS al aplicar técnicas en paralelo para CPU's y GPU's utilizando OpenMP, alcanzando una optimización del 81,35% comparando los dos escenarios; donde el primer escenario utilizó el 90,67% de tiempo de CPU y el segundo escenario utilizo el 9,33% de tiempo de CPU y GPU.

Se alcanzó detectar que la mayor carga de trabajo se realizó en la CPU, a pesar de que el proceso se dividió en CPU´s y GPU´s; identificando que la CPU realizo el 57,8% de uso y del GPU el 42,2% de uso durante el procesamiento.

Agradecimientos

Los autores reconocen el apoyo de programa de Ph.D. de Tecnología de la Información, campus de CUCEA, Universidad de Guadalajara, y el apoyo financiero para esta investigación de la Universidad Politécnica Salesiana.

Referencias Bibliográficas

[1] Doi, K. "Computer-aided diagnosis in medical imaging: historical review, current status and future potential". Computerized medical imaging and graphics, Vol. 31, Nº 4-5, (2007), 198-211.

[2] Park S. C., Park M. K. and Kang M. G. "Super-resolution image reconstruction: a technical overview". IEEE Signal Processing Magazine, Vol. 20, Nº 3, (2003), 21-36.

[3] Pulli, K., Baksheev A., Kornyakov K. andEruhimovV. "Real-time computer vision with OpenCV".Communications of the ACM, Vol. 55, Nº. 6(2012), 61-69.

[4] García Flores R. «Ingeniería concurrente y tecnologías de la información». Ingenierías, R. García Flores, «Ingeniería concurrente y tecnologías de la información,» Ingenierías, vol. 7, no. 22, pp. 39-44, 2004ol. 7, Nº. 22, (2004), 39-44.

[5] Sharma J.and SharmaS., "Mammogram image segmentation using watershed".Int J Info Tech and Knowledge Management, Vol. 4, (2011), 423-425.

[6] GulsrudT. O, EnganK. and HanstveitT. "Watershed segmentation of detected masses in digital mammograms".In 2005 IEEE Engineering in Medicine and Biology 27th Annual Conference, (2005), 3304-3307.

[7] Hwang Y.and ChungK.S. "Dynamic Power Management Technique for Multicore Based Embedded Mobile Devices". IEEE Transactions on Industrial Informatics, Vol. 9, Nº 3, (2013), 1601-1612.

[8] Harwood A. R.and RevellA. J. "Parallelisation of an interactive lattice-Boltzmann method on an Android-powered mobile device". Advances in Engineering Software, Vol. 104, (2017), 38-50.

[9] Boyle R. D. and ThomasR. C."Computer vision: a first course". Blackwell Scientific Publications, Ltd., (1988).

[10] RaniV. "A brief study of various noise model and filtering techniques".Journal of global research in computer science, Vol. 4, Nº 4, (2013), 166-171.

[11] OtsuN., "A threshold selection method from gray-level histograms".IEEE transactions on systems, man, and cybernetics, Vol. 9, Nº. 1, (1979), 62-66.

[12] ValaH. J. and BaxiA. "A review on Otsu image segmentation algorithm". International Journal of Advanced Research in Computer Engineering & Technology (IJARCET), Vol. 2, Nº 2, (2013) 387-389.

[13] TulsaniH., SaxenaS. and YadavN. "Segmentation using morphological watershed transformation for counting blood cells". IJCAIT, Vol. 2, Nº. 3, (2013), 28-36.

[14] Shareef, S. R. "Breast cancer detection based on watershed transformation". IJCSI International Journal of Computer Science Issues, Vol. 11, Nº 1, (2014),237-245.

[15] Ortiz Zamora F. G."Procesamiento morfológico de imágenes en color: aplicación a la reconstrucción geodésica.Universidad de Alicante. Departamento de física, Ingeniería de sistemas y Teoría de la señal, (2002).

[16] Vincent L. and Soille P. "Watersheds in digital spaces: an efficient algorithm based on immersion simulations" IEEE Transactions on Pattern Analysis & Machine Intelligence, Vol. 6, (1991), 583-598.

[17] Romero-ZalizR. and Reinoso-GordoJ. "An updated review on watershed algorithms". In Soft computing for sustainability science, (2018) 235-258.

[18] KaurA."Aayushi.Image Segmentation Using Watershed Transform "Blue Eyes Intelligence Engineering & Sciences Publication Pvt. Ltd. International Journal of Soft Computing and Engineering (IJSCE) Vol. 4, (2014) 5-8.

[19] AcharjyaP.A.,Sinha, S. Sarkar, S. Dey, and GhoshS. "A new approach of watershed algorithm using distance transform applied to image segmentation". International Journal of Innovative Research in Computer and Communication Engineering, Vol. 1, Nº 2, (2013) 185-189.

[20] Abdullah-Al-WadudM., KabirM. H., A. DewanM. A. and ChaeO. "A Dynamic Histogram Equalization for Image Contrast Enhancement". IEEE Transactions on Consumer Electronics, Vol. 53, Nº 2, (2007), 593-600.

[21] RomaniS., SobrevillaP. and MontsenyE., "Obtaining the relevant colors of an image through stability-based fuzzy color histograms". The 12th IEEE International Conference on Fuzzy Systems, FUZZ '03,Vol. 2, (2003), 914-919.

[22] Barney, B. "Using the Sequoia and Vulcan BG/Q Systems. Tutorial Lawrence Livermore National Laboratory, Lawrence Livermore National Laboratory". (2014). URL https://computing.llnl. gov/tutorials/bgq/# ParallelIO, 8.

[23] BellifemineF., BergentiF., CaireG. and PoggiA. "JADE—a java agent development framework," Multi-Agent Programming.Springer, (2005), 125-147.

[24] BellifemineF. L.,CaireG. and Greenwood D. "Developing multi-agent systems with JADE2.John Wiley & Sons, (2007).

[25] CaireG.andPieriF., "Leap user guide" TILab, (2006).

[26] KaehlerA. and BradskiG."Learning OpenCV 3: computer vision in C++ with the OpenCV library". O'Reilly Media, Inc.", (2016).

[27] VincentL. "Morphological Grayscale Reconstruction in Image Analysis: Applications and E cient Algorithms".IEEE transactions on image processing, Vol. 2, Nº 2, (1993), 176-201.

[28] PreimB.andBartzD."Visualization in medicine: theory, algorithms, and applications". Elsevier, (2007).

[29] Omer A. M. and ElfadilM. "Preprocessing of Digital Mammogram Image Based on Otsu's Threshold," American Scientific Research Journal for Engineering, Technology, and Sciences (ASRJETS), Vol. 37, Nº 1, (2017), 220-229,.

[30] Lewis S. H. and DongA. "Detection of breast tumor candidates using marker-controlled watershed segmentation and morphological analysis".In 2012 IEEE Southwest Symposium on Image Analysis and Interpretation, (2012), 1-4.

[31] KooJ. J. P. and QuintalL. F. C. «Análisis del desempeño de aplicaciones paralelas con openMP en dispositivos móviles multicore, caso de estudio: multiplicación de matrices». Pistas Educativas, Vol. 36, Nº 114(2018), 6-17.

UNIVERSIDAD
DEL ZULIA

REVISTA TECNICA

DE LA FACULTAD DE INGENIERIA
UNIVERSIDAD DEL ZULIA

Volumen Especial, 2020, No. 1, pp. 03 - 55_____

Esta revista fue editada en formato digital y publicada en Febrero de 2020, por el ***Fondo Editorial Serbiluz, Universidad del Zulia. Maracaibo-Venezuela***

www.luz.edu.ve
www.serbi.luz.edu.ve
www.produccioncientifica.luz.edu.ve