

Una Solución Paralela Simple al Método de Difusión de Error

G. Abdala, R. Guerrero, F. Piccoli *

Líneas Informática Gráfica y Paralelismo y Distribución del
Laboratorio de Investigación y Desarrollo en Inteligencia Computacional
Universidad Nacional de San Luis
5700 - San Luis - Argentina
e-mail: {rag, mpiccoli}@unsl.edu.ar

Abstract

Digital halftoning is a well-known technique for electronic devices that can reproduce only black and white pixels. It converts gray scale images into black and white ones creating the illusion of continuous tones. Besides the large number of existing methods and algorithms, all of them share the same problem: image size and selected algorithm dependence. As a consequence, it is necessary to develop new techniques in an independent method and data way. This paper presents two parallel implementations of Floyd-Steinberg's diffusion error method; an inherently sequential method. The first one implements a proposed design by Metaxas, and the second one tries, proposed by authors, a simple segmentation and coarse grain parallel techniques. In addition, an experimental comparison of the implementations are shown.

Keywords: Digital Halftoning, Computer Graphics, Image Processing, Parallel Systems, Parallelism Grain.

Resumen

El Halftoning o Mediotonado Digital se utiliza para plasmar imágenes en escala de grises en dispositivos electrónicos que sólo pueden reproducir elementos binarios creando la ilusión de tonos continuo. A pesar de los numerosos métodos de halftoning existentes, todos ellos comparten la problemática de que su tiempo de procesamiento es dependiente de las características de la imagen y el algoritmo utilizado. Es de interés la generación de propuestas para acelerar los tiempos de procesamiento independientemente de los métodos y los datos. En este trabajo se presentan dos sistemas paralelos del Método de Difusión de Error establecido por *Floyd y Steinberg*, considerado un método inherentemente secuencial. El primero es la implementación del diseño desarrollado por Metaxas y el segundo, propuesto por los autores, se caracteriza por aplicar una segmentación simple de la imagen de entrada y técnicas paralelas de gránulo grueso. Además de presentar ambos sistemas, se muestran algunos resultados experimentales, tanto en el desempeño como en la calidad de las imágenes resultantes.

Palabras claves: Mediotonado Digital, Computación Gráfica, Procesamiento de Imágenes, Sistemas Paralelos, Gránulo de Paralelismo.

*Grupo subvencionado por la UNSL y ANPCYT (Agencia Nacional para la Promoción de la Ciencia y Tecnología)

1. INTRODUCCIÓN

La palabra halftoning (mediotonado) tiene sus orígenes en la antigüedad; describe una técnica utilizada durante los siglos XVII y XVIII para crear la ilusión de imágenes de tono continuo mediante una plantilla de cobre y un único tono de tinta.

En la actualidad, el “mediotonado digital” es una técnica muy conocida en procesamiento de imágenes [9], ésta permite convertir una imagen representada a través de un rango de niveles de intensidad de luz en una imagen consistente solamente de puntos blancos y negros. La propuesta ha sido concebida principalmente para el plasmado de imágenes a través de dispositivos de impresión y visualización binarios, uso de pixels blancos y negros, pero con la intención de ser percibidas como una imagen en tonos de grises. El objetivo del proceso es lograr, como declara Ulichney: *“la ilusión de imágenes en tono continuo a partir de una cuidadosa ubicación de elementos de dibujo binarios”* [13].

Por ser un proceso que proyecta una escala continua en una escala binaria, adolece de pérdida de información. Para asegurarse el éxito del proceso, la técnica se apoya en las propiedades del sistema de visión humano conjuntamente con los atributos propios de la imagen. Ésta intenta establecer un ordenamiento de los elementos de dibujo, de manera tal de generar en el observador una impresión visual lo más parecida posible al original.

Desafortunadamente, se desconocen aún todos los factores intervinientes en la percepción humana y en la formación de las imágenes, así como también las métricas cuantitativas reales que puedan correlacionar la calidad subjetiva de una imagen [4]. En consecuencia, una variedad de algoritmos para el proceso de halftoning ha sido desarrollada a lo largo de los años [5, 7, 13]. La característica común a todos es el tiempo de procesamiento de una imagen, el cual es directamente proporcional al tamaño de la misma más el tiempo asociado al método en particular. Los algoritmos simples y de bajo tiempo de proceso producen resultados de baja calidad, mientras que los algoritmos con buenos resultados poseen tiempos de ejecución prolongados. Estas características en conjunto con la natural autocorrelación de los datos involucrados convierte al problema de mediotonado en una propuesta de análisis y estudio válida para la aplicación de modelos de programación paralela [3, 16].

Este trabajo está organizado de la siguiente manera, en la próxima sección se detallan los métodos más comunes para obtener imágenes mediotonadas. En la sección 3 se explica con mayor detalle el Método de Difusión de Error. En la sección 4 se presentan las dos soluciones paralelas a dicha técnica. Finalmente, se realiza un análisis de los resultados obtenidos desde el punto de vista de tiempos de proceso y de la calidad cualitativa y cuantitativa de las imágenes.

2. TÉCNICAS PARA MEDIOTONADO DE IMÁGENES

No existe un único método para obtener una imagen mediotonada a partir de otra en tonos continuos de grises. Los métodos existentes intentan crear imágenes consistentes solamente de tonos negros y blancos que sean buenas aproximaciones de las imágenes originales en escalas de grises. La calidad de una aproximación es comúnmente evaluada en función a la calidad en la

reproducción de la intensidad local y la ausencia de artificios (discontinuidades en la imagen resultante). Las técnicas de *difusión de error* [2, 13] intentan aproximar la intensidad local de una escala de grises por medio del umbralizado de una región de la imagen y la distribución del error de cuantización de la región procesada a regiones vecinas aún sin procesar. La distribución intenta compensar los errores de cuantización locales asegurándose que sobre una región levemente mayor, la aproximación de la escala de grises es correcta. La figura 2(a) ilustra el método de difusión de *Floyd y Steinberg* de la figura 1.



Figura 1: Imagen Original

Los métodos de *matrices de umbralización*[8] usan una disposición predefinida de los pixels blancos y negros para aproximar la intensidad local de la imagen. Estos métodos no compensan los errores de cuantización local puesto que se amparan en la baja probabilidad de variación de la intensidad local. Un ejemplo de ello se muestra en la figura 2(b). Las matrices establecen además el orden en que los pixels deben ser coloreados. Si se cambia la disposición de los pixels en la matriz, se altera la forma de los pixels que son aproximados y por consiguiente la imagen resultante. Estas técnicas intentan balancear la cantidad de tonos de escala de grises aproximados con la cantidad de artificios introducidos. Las investigaciones [13] en esta área se centran principalmente en el desarrollo de matrices que minimicen la generación de artificios.



(a) Difusión de Error

(b) Matriz de Umbral

Figura 2: Imágenes Mediotonadas

Por otra parte, los métodos de *halftoning híbridos* [1, 11] intentan combinar las técnicas de distribución del error local de los métodos de difusión de error con la eficiencia de la disposición local de pixels de los métodos de matrices de umbral. Estos métodos calculan el error basados en un solo pixel o región de pixels (segmento, curva o matriz). De la misma manera, algunos métodos intentan eliminar la generación de artificios mediante la alteración del recorrido de la imagen. La aplicación repetitiva de un mismo proceso a lo largo de una ruta predefinida resulta en la aparición recurrente de patrones en la imagen resultante. Los métodos de difusión de error y matrices de umbral utilizan un paso de fila-columna pre-establecido. Otras técnicas de mediotonado proponen el uso de curvas de llenado de espacios [14] y pasos de onda (*wavefront*)[7]. El método de difusión de punto de Knuth [5] utiliza una matriz de valores para especificar el orden en el que se debe distribuir el error dentro de la matriz.

3. MÉTODO DE DIFUSIÓN DE ERROR

El método de Difusión del Error, *MDE*, para el mediotonado de imágenes en tonos de grises fue propuesto por *Floyd y Steinberg* [2] y ha sido tomado como estándar por su simplicidad y calidad de la imagen mediotonada resultante. La figura 3 muestra gráficamente como se realiza el procesamiento de los pixels a través de la técnica de Difusión de Error.

En *MDE* los pixels de una imagen de tonos de grises ($J[n]$) son procesados en forma lineal de izquierda a derecha y de arriba hacia abajo. Cada pixel, representado por un valor entre 0 y 255, es comparado con un valor de umbral, por ejemplo 128. Si el valor es mayor que el umbral, el pixel se considera como blanco y la salida ($I[n]$) es colocada en 1. Caso contrario se considera negro e $I[n]$ es instanciado en 0. La diferencia entre el valor original del pixel y el umbral es considerado *un error*, el cual es distribuido a los cuatro pixels vecinos no procesados aún. La difusión del error a los vecinos de un pixel es hecha en forma proporcional, ver fig. 4.

El proceso es netamente secuencial y exhibe una gran dependencia entre los datos de entrada, pixels en tono de gris. Analizando las dependencias, se observa que el valor del pixel de la posición (fil, col) en la imagen mediotonada depende de su valor de gris y de los errores difundidos por sus vecinos ($fil, col-1$), ($fil-1, col-1$), ($fil-1, col$) y ($fil-1, col+1$) procesados previamente. La dependencia entre los pixels es transitiva, a excepción del pixel en la posición (0,0).

El algoritmo secuencial de *MDE* consta de tres partes: lectura de la imagen de tonos de gris, procesamiento de la imagen y grabado de la imagen mediotonada resultante. En la figura 5 se muestra un pseudo-código simplificado del procesamiento de una imagen de $M*N$ pixels.

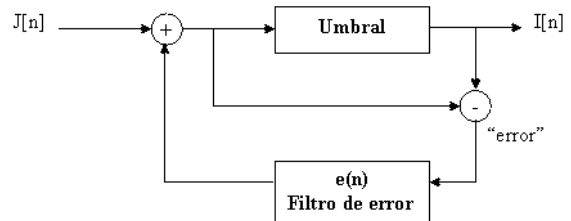


Figura 3: Diagrama de Proceso *MDE*

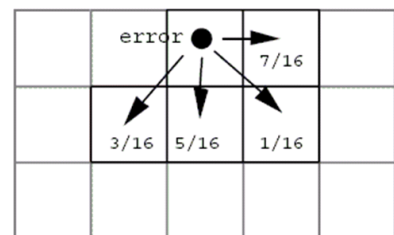


Figura 4: Difusión del Error

```

1. for i = 1 to M
2.   for j = 1 to N
3.     {
4.       I[i,j] = (J[i,j] < 128)? 0 : 1 //blanco=1 y negro=0
5.       err = J[i,j] - I[i,j]*255
6.       if(PixelValido(i+1,j)) J[i+1,j] += err*(7/16)
7.       if(PixelValido(i-1,j+1)) J[i-1,j+1] += err*(3/16)
8.       if(PixelValido(i,j+1)) J[i,j+1] += err*(5/16)
9.       if(PixelValido(i+1,j+1)) J[i+1,j+1] += err*(1/16)
10.    }

```

Figura 5: Etapa de Procesamiento de *MDE*

Los valores de blanco o de negro para cada pixel de la imagen de salida son determinados en la línea 4 y el error a difundir a todos los vecinos no procesados se calcula en la línea 5. Finalmente, se realiza la difusión del error a cada uno de los vecinos (líneas 6 a 9). La función *PixelValido()* asegura que la difusión corresponda a un pixel real de la imagen.

4. APLICANDO PARALELISMO A MDE

Como se mencionó anteriormente, aplicar técnicas paralelas en la solución computacional de *MDE* resulta, a primera vista, una tarea difícil. Proponer soluciones paralelas implica intentar resolver o ignorar la dependencia de los datos.

En este trabajo se presentan dos soluciones paralelas de *MDE*. La primera, basada en la propuesta de paralelización de Metaxas [6], presenta un paralelismo de gránulo fino, a nivel de pixel. La segunda (propuesta de los autores), en cambio, propone una solución simple y débilmente acoplada a costa de ignorar parcialmente la dependencia entre los datos, pixels de la imagen y error difundido. En las siguientes secciones se especifican las características básicas de cada una de ellas.

4.1. Solución Paralela de Gránulo Fino

A partir de su formulación, *MDE* ha sido considerado un método inherentemente secuencial. Del análisis detallado de las dependencias, en [6] se demuestra lo contrario: no existe una interrelación generalizada entre todos los pixels, sino entre un subgrupo regional de los mismos. El valor de un pixel de la imagen resultante, ubicado en la posición (fil, col) depende del valor de gris de dicho pixel en la imagen original y de los errores difundidos por sus vecinos $(fil, col-1)$, $(fil-1, col-1)$, $(fil-1, col)$ y $(fil-1, col+1)$, los cuales ya han sido procesados. A su vez, cada uno de los vecinos depende de sus propios vecinos, también ya procesados. La aplicación transitiva de esta propiedad genera un área trapezoidal de dependencias. En la figura 6 se puede observar el área trapezoidal, zona gris, de la cual depende el pixel indicado con color negro.

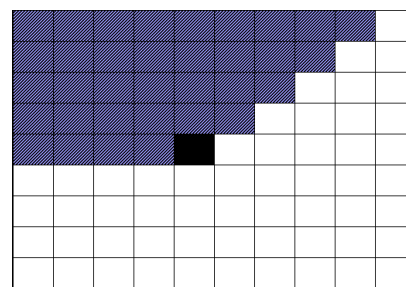


Figura 6: Dependencia de los pixels

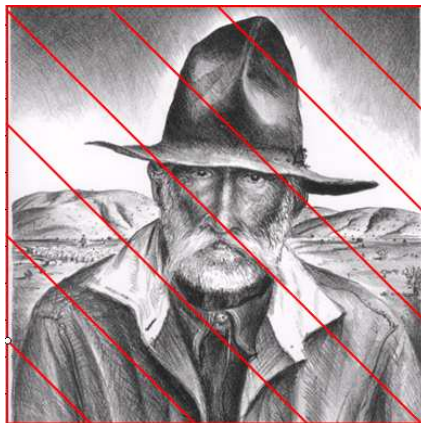
Como la relación entre los pixels debe respetarse, es necesario planificar el orden de procesamiento de cada uno de ellos. La manera de planificar el procesamiento fue denominada *Invariante I* [6]. La *Invariante I* establece que todo pixel en la posición (fil, col) es procesado en el tiempo T , siempre y cuando $T(fil, col)$ cumpla con:

$$T(fil, col) > MAX\{T(fil, col - 1), T(fil - 1, col - 1), T(fil - 1, col), T(fil - 1, col + 1)\}$$

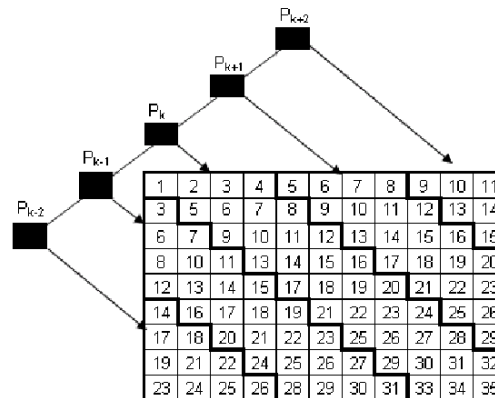
El Método Paralelo de Difusión de Error, *MPDE*, propone resolver en paralelo el mediotonado de una imagen de tonos de grises respetando la formulación original y aplicando las consideraciones enunciadas anteriormente. Para ello propone segmentar la imagen en forma diagonal, ver figura 7(a).

Los segmentos diagonales poseen *cierta independencia* en el tiempo: en un segmento existen pixels cuyo procesamiento es independiente de los pixels asignados a otros. Los pixels dependientes de los errores de otros pixels en otros segmentos son aquellos ubicados en los límites del segmento o cercanos a ellos. Cuanto mayor es la cantidad de pixels internos de un segmento, mayor será la independencia de los mismos.

Cada procesador debería ser responsable del mediotonado de un segmento de la imagen, ver figura 7(b). El procesamiento es sincronizado, todos aquellos pixels con igual identificador pueden procesarse en paralelo siempre y cuando ya fueron procesados los pixels de menor identificador.



(a) División de la Imagen



(b) Planificación y Asignación

Figura 7: Tratamiento de una Imagen en *MDEP*

En este trabajo, la implementación de *MDEP* se realizó teniendo en cuenta lo anteriormente expresado y asumiendo una imagen cuadrada con al menos 2 filas. Dichas restricciones fueron hechas a efecto de simplicidad; mínimas modificaciones son necesarias para ofrecer generalidad.

La figura 8 muestra el pseudo-código del algoritmo general de *MPDE*. Cada procesador realiza los cálculos iniciales para establecer los valores adecuados de distancia de los segmentos (ancho máximo del segmento), cantidad de procesadores utilizados y cantidad de procesadores libres (línea 1). Luego, se realiza la lectura del segmento correspondiente (línea 2). El procesamiento se inicia cuando el procesador P_k realiza el *Paso Inicial* (línea 3). Posteriormente, cada procesador realiza el mediotonado del segmento correspondiente (línea 4) y envían su segmento mediotonado al procesador P_k (línea 12). El procesador P_k se encarga de recibir todos los segmentos mediotonados y realizar el armado de la imagen resultante (líneas 5 a 10).

```

1. Cálculos iniciales.
2. Lectura del Segmentoi de la imagen.
3. if(id_procesador==k) PasoInicial
4. Mediotonado(Segmentoi, SegmentoMi).
5. if(id_procesador == k)
6. {
7.   For(i=1; i <  $P_o-1$  ; i++)
8.     Recibir SegmentoMi modificado desde procesador  $i > 10$ .
9.   Armar imagen de salida.
10. }
11. else
12.   Enviar SegmentoMi al procesador k.

```

Figura 8: Pseudo-código de *MPDE*

MDEP involucra varios aspectos:

- Los cálculos iniciales de distancia, procesadores ocupados y procesadores libres son determinados según el tamaño de la imagen y la cantidad de procesadores disponibles. El tamaño de los segmentos se determina de manera tal de maximizar el número de procesadores ocupados.

- El inicio del procesamiento es responsabilidad del procesador P_k , quien es responsable de procesar dos pixels antes de que otro procesador inicie sus tareas. El procesador P_k debe procesar y difundir el error de los pixels: $(0,0)$ y $(0,1)$. La difusión es tanto a los pixels pertenecientes a su segmento como a los pertenecientes al segmento del procesador P_{k-1} . El procesamiento del pixel en la posición $(0,0)$ es denominado “*Paso Inicial*”
- Los procesadores sincronizan su ejecución a través de los mensajes con los errores difundidos, un procesador trata un determinado pixel cuando todos los errores necesarios están presentes. Los procesadores intercambian los errores según corresponda. El uso de comunicaciones de tipo bloqueante permite sincronizar el trabajo paralelo sin tener que recurrir a sincronizaciones explícitas. Después de realizar los cálculos iniciales, la primera instrucción a ejecutar por todo procesador distinto de k , es recibir el error difundido por otro procesador.
- La estructura de almacenamiento para los segmentos es una matriz, donde la cantidad de columnas es igual a la distancia y el número de filas varía de una a otra, dependiendo del segmento. Toda fila cuya cantidad de columnas es menor a la distancia se la denomina *fila incompleta*. Una *fila incompleta* se completa con valores de relleno a fin tener la misma cantidad de columnas en todo el segmento.
- Respetando la formulación de *MDE*, el error del pixel actualmente procesado debe ser difundido a sus vecinos sin procesar aún. La tarea de difusión del error debe incorporar el error al pixel indicado si es un pixel del segmento. Si en cambio el pixel corresponde a un segmento de otro procesador, se procede a enviarle el error, error remoto. Los pixels internos del segmento realizan una difusión interna del error, no ocurre lo mismo con los pixels situados en los bordes del segmento, la difusión será realizada a los segmentos vecinos.
- El proceso aplicado a cada pixel de la imagen, a excepción del pixel procesado en el *Paso Inicial*, se encarga de: recibir y agregar los errores remotos, realizar el procesamiento del pixel indicado y difundir el nuevo error.
- Al finalizar el proceso de mediotonado de los segmentos, el procesador P_k recibe los segmentos de la imagen procesados en los otros procesadores y armar la imagen de salida.

Todos estos aspectos son coordinados en la cuidadosa planificación del procesamiento que resuelve en paralelo al *MDE* respetando la formulación original.

4.2. Solución Paralela de Gránulo Grueso

El Método de Difusión de Error Paralelo con enfoque Trivial Dinámico, *MDEP_TD*, resulta de aplicar técnicas de programación paralela al *MDE* ignorando parcialmente la propiedad de los datos de entrada. Para ello, propone resolver en paralelo *MDE* dividiendo la imagen en

subimágenes e ignorando la dependencia de los datos en los bordes de dichas particiones. Para ello, la imagen original, en tonos de grises, es dividida en porciones o segmentos, a los cuales se le aplica el proceso normal de *Floyd y Steinberg*. *MDEP_TD* tiene una estructura del tipo *farm* [10], cada proceso trabaja en forma independiente a los demás, sólo se comunica con el proceso responsable de la distribución de los segmentos de la imagen en tono de grises y del armado de la imagen mediotonada de salida. La cantidad de procesos trabajando es determinada por la cantidad de procesadores disponibles. Es por ello que la división de la imagen se fija en tiempo de ejecución según el número de procesadores disponibles. Respetando la secuencia de almacenamiento de los pixels, la división de la imagen se realiza en forma horizontal, ver figura 9.

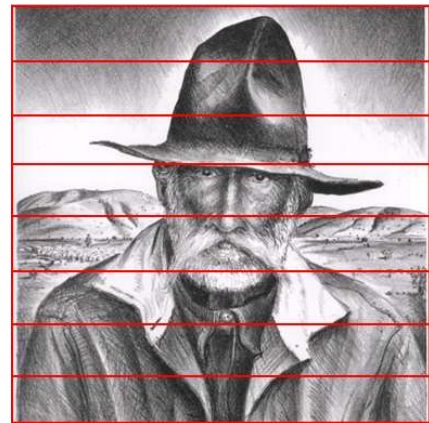


Figura 9: División en 8 segmentos

Como *MDEP_TD* no tiene en cuenta la dependencia de los pixels en el límite de los segmentos ni el error acumulado, es de esperar la aparición de artificios no presentes en la imagen original. Para minimizar su aparición se propone considerar un solapamiento de los segmentos, para cada segmento se consideran *PA* filas del segmento anterior. Esto tiene como objetivo aproximar el error acumulado de los pixels residentes en el límite superior del segmento.

En la figura 10 se muestra el pseudo-código correspondiente a la estructura general de *MDPE_TD*.

```

1.  Cálculos iniciales.
2.  Lectura del Segmentoi de la imagen.
3.  Mediotonado(Segmentoi, SegmentoMi).
4.  if(id_procesador == 0)
5.  {
6.      For(i=1; i < Procesadores_disponibles-1 ; i++)
7.          Recibir SegmentoMi modificado desde procesadori > 0.
8.          Armado imagen de salida.
9.  }
10. else
11.     Envío SegmentoMi al procesador 0.

```

Figura 10: Pseudo-código de *MDPE_TD*

Al inicio de la computación cada procesador realiza los cálculos iniciales y obtiene el segmento de la imagen correspondiente (líneas 1-2) para luego aplicar la función *Mediotonado()* (línea 3). Esta función es idéntica al proceso secuencial. Para el caso de la imagen mediotanada resultante, la responsabilidad de su armado recae en uno de los procesadores, en este caso el *procesador 0*. Por esta razón al finalizar el mediotanado de cada segmento, todos los procesadores envían sus resultados al *procesador 0* (línea 11), quien los recibe y arma la imagen mediotanada resultante (líneas 4-8).

MDEP_TD propone una solución paralela de gránulo grueso simple y escalable según el tamaño de la imagen y la cantidad de procesadores. Aunque su formulación parte de ignorar la dependencia de los datos, permite atenuar sus consecuencias.

5. RESULTADOS EXPERIMENTALES

En esta sección se muestran algunos resultados experimentales del trabajo realizado. Cada sistema se implementó en lenguaje C, utilizando la biblioteca de pasaje de mensajes MPI [12]. Todas las soluciones fueron ejecutadas sobre la computadora paralela Cray Origin 2000 perteneciente a la Secretaria de Ciencia, Técnica e Innovación Productiva *SeCyT*. Los resultados reportados corresponden a tiempos totales promedios (tiempo total= lectura de la imagen, proceso de mediotonado y armado de la imagen). En todos los casos se aplicó el proceso a imágenes cuadradas de 300, 600 y 2000 pixels cada una, denominadas *Im-C*, *Im-M* y *Im-G* respectivamente. En todas las soluciones paralelas se utilizaron 4, 16 y 24 procesadores.

En la figura 11 se presenta el desempeño de ambas propuestas paralelas. En la parte (a) se muestra la aceleración obtenida por el sistema *MDEP* para los distintos tamaños de imágenes. Se puede observar un mejor desempeño de *MDEP_TD*, parte (b). Esto obedece al grado de paralelismo involucrado en cada solución, *MDEP* es un solución paralela gránulo fino, el grado de paralelismo es determinado por el tamaño de los segmentos y los pixels involucrados. Por lo contrario, el proceso de mediotonado de cada pixel en cada segmento en *MDEP_TD* es totalmente independiente de los pixels en los otros segmentos. Además, en ambos casos se puede apreciar la influencia de las comunicaciones en el desempeño de los sistemas, tornándose evidente para imágenes más pequeñas y mayor número de procesadores.

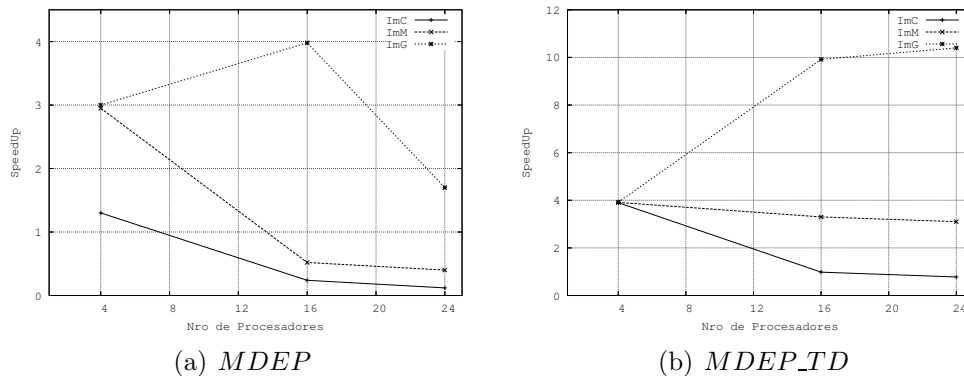


Figura 11: Aceleración obtenida para cada Método

Del análisis de los resultados y en función a las características de cada propuesta, se puede establecer que en ambos casos se obtuvieron aceleraciones aceptables, teniendo un mejor desempeño *MDEP_TD*.

Además del análisis de tiempo de los procesos, y dado que las propuestas difieren en las consideraciones de interrelación entre pixels, es necesario el análisis de la calidad de las imágenes resultantes. Para poder evaluarlas es prioritaria la valoración de las mismas mediante algún método. Tanto en el area de computacion grafica como en el área de procesamiento de imágenes se han adoptado dos tipos de métodos para la valoración de las imágenes: *medidas cualitativas y cuantitativas*. La evaluación cualitativa de las imágenes se basa principalmente en el sistema visual humano y las observaciones de los participantes de dicha evaluación. En consecuencia, las evaluaciones de las imágenes tienden a ser altamente subjetivas. La métricas cuantitativas proveen una evaluación mas objetiva de las imágenes. Sin embargo, estas evaluaciones generalmente no son completas pues no existe una única métrica para evaluar todas las dimensiones necesarias para la valoración de la imagen. Debido justamente a la falta de una métrica completa, es que uno de los métodos mas comúnmente utilizados es la comparación visual.

Las imágenes pueden ser comparadas analizando los componentes comunes de las imágenes mediotonadas, tales como la reproducción de la intensidad de la escala de grises, bordes y la preservación de la información de alta frecuencia, así como también la incorporación de artificios. Dado que no se está comparando métodos de halftoning diferentes sino diferentes implementaciones del mismo método, todas ellas reproducirán de la misma manera la escala tonal y mayormente preservarán los bordes y la información de alta frecuencia. La diferencia estará dada entonces por la incorporación de artificios producto de la segmentación de la imagen.

En el apéndice se presentan un conjunto de imágenes para la evaluación cualitativa por medio de la comparación visual de sus componentes. De la misma manera, con el objeto de poder realizar una comparación objetiva de los resultados, otras imágenes son presentadas para ser evaluadas mediante una métrica cuantitativa propuesta por Vevyovka *et al.* [15]. Dicha métrica permite medir la preservación de bordes y la creación de bordes artificiales a múltiples resoluciones de las imágenes (escalas).

Para la evaluación cualitativa se presentan dos imágenes de Lena, la cuales son el resultado de aplicar las dos propuestas presentadas (*MDEP* y *MDEP_TD*). Como era de esperar, la imagen producto de aplicar *MDEP* tiene características semejantes a la obtenida con el proceso original, ver figuras 12(a) y 2(a) respectivamente. Para el caso de la imagen mediotonada resultante del proceso *MDEP_TD*, el ojo adiestrado podría observar la aparición de artificios horizontales derivados de ignorar parcialmente la dependencia de los datos, ver figura 12(b). Dicha imagen se ha obtenido de aplicar *MDEP_TD* sin considerar solapamiento de segmentos.

Para la evaluación cuantitativa se presentan dos nuevas imágenes. La primera es una rampa de tono continuo generada por computadora que va del negro total (100 %) al 75 % de negro, ver fig. 13(a). Esta imagen es utilizada para indicar la incorporación de artificios en una región de tono continuo (artificios de bordes). La otra imagen es una imagen de negro al 50 % como fondo, con una banda blanca, una banda de rampa de tono continuo desde el negro al blanco y una banda negra. Esta imagen provee diferentes bordes definidos por medio de distintos cambios de intensidad, fig. 13(b). Es utilizada para indicar la destrucción o preservación de bordes.

La tabla 1 muestra los resultados de utilizar la métrica de arteificio de bordes de Vevyovka sobre las imágenes de la fig. 14. Los valores reportados en esta tabla reflejan la fuerza de los artificios no deseados donde los valores altos indican mayor cantidad de artificios. La tabla muestra el mínimo, promedio y máximo valor de fuerza de los bordes. El análisis de los tres valores permite formular una conclusión.

En la escala 1, las tres métricas tienen el mismo valor máximo. *MDEP_TD* introduce bordes débiles debido al particionado propio de la implementación. Analizando los valores se puede concluir que *MDEP_TD* obviamente introduce más bordes débiles en comparación con las otras dos, considerando la caída en el promedio reportado en la fuerza del borde. En la segunda escala, *MDEP_TD* se comporta peor en promedio que las otras dos implementaciones. La fuerza de borde mínima es tan pequeña como la de Floyd-Steinberg, sin embargo el máximo es el mayor de todas, esto se refleja en el incremento en el valor promedio. En las escalas 3 y 4, el valor promedio de *MDEP_TD* es nuevamente mayor que los otros dos. Sin embargo, la fuerza de borde máxima en ambas escalas es menor que la de Floyd-Steinberg. La mínima fuerza de borde para *MDEP_TD* es, sin embargo, mayor causando el incremento en el valor promedio. En la escala 5, el valor promedio de la fuerza de borde para *MDEP_TD* es comparable con la de Floyd-Steinberg y es menor que *MDEP*. Los valores de fuerza de borde máximo y mínimo para *MDEP_TD* son comparables a las de las otras dos.

La tabla 2 muestra los resultados de medir la preservación de bordes en las imágenes de la fig. 15. Esta medida toma en cuenta que el sistema visual humano no detecta pequeños

desplazamientos en la ubicación de los bordes, de modo que se preserve la geometría de los bordes. De la misma manera, bordes muy pequeños no son detectados o son detectados como ruido o artificios, en lugar de bordes, los cuales deben ser descartados. Los valores presentados en la tabla miden las distorsiones de bordes de la escala 4. Un valor de cero o cercano a cero es una valor deseable. Un valor negativo representa un suavizado de los bordes a consecuencia del proceso de halftoning y un valor positivo representa una exageración del borde a causa del proceso. *MDEP_TD* tiene una distorsión en promedio parecida a los otros dos. De la misma manera, exagera bordes producto del tratamiento particionado.

6. CONCLUSIONES

Si bien existen varios métodos para obtener imágenes mediotonadas a partir de imágenes en tonos de grises, la aplicación de un método en particular depende de los recursos disponibles y de la calidad de la imagen resultante deseada. El Método de Difusión de Error es considerado un método simple y con buenos resultados. Por mucho tiempo fue considerado inherentemente serial dada la fuerte dependencia entre los pixels. A partir de un análisis detallado de la correlación de los datos y el orden de procesamiento, el mismo puede ser paralelizado, optimizada su performance y lo más importante, hacer su procesamiento portable.

Este trabajo implementa el sistema paralelo propuesto por Metaxas y propone un sistema alternativo. Ambos sistemas fueron desarrollados para el paradigma de pasaje de mensajes. Las diferencias en los resultados obtenidos para ambos están directamente relacionadas a las características propias de cada uno: el gránulo de paralelismo implicado en la solución.

Si bien los resultados son alentadores, principalmente para *MDEP_TD* por su simplicidad y performance, un análisis más exhaustivo debe ser hecho respecto al grado de solapamiento de los segmentos. Por ello, determinada el área influyente para los pixels de cada segmento, la interrelación de los pixels de la imagen puede ser sobrellevada.

Además de evaluar detalladamente los costos-beneficios de cada una de las soluciones paralelas, resulta necesario analizar la posibilidad de migración de los sistemas propuestos a arquitecturas paralelas débilmente acopladas y evaluar, consecuentemente, su buen desempeño.

Respecto a la calidad de las imágenes mediotonadas obtenidas, ambos sistemas ofrecieron buenos resultados, no sólo comprobables visualmente (al ojo humano), sino también mediante métricas aplicadas a la imagen propiamente dicha. Perceptualmente, las imágenes obtenidas aumentan su similitud a medida que los procesos disponen de mayor información (mayor resolución), no presentando diferencias evidentes para ambos métodos. Para el caso de *MDEP_TD*, la aparición de artificios fue mínima cuando no existía solapamiento de los segmentos. Al introducir solapamiento se obtuvo una mejor calidad de imagen manteniéndose los tiempos de procesamiento.

La medida de artificios de bordes propuesta por Veryovka *et al.* muestra que en las escalas 1 y 5 nuestra propuesta se desarrolla en forma semejante a Floyd-Steinberg, sin embargo en las escalas de 2 a 4, los artificios de borde promedio introducidos son mas fuertes y aparentes. La medida de preservación de bordes muestra que nuestra propuesta exagera los bordes sólo en las regiones de unión de segmentos procesados. Estos resultados se correlacionan con aquellos que surgen de la evaluación perceptual.

En conclusión, dependiendo de las características de resolución y composición de la imagen los bordes se evidenciarán en mayor o menor medida. A grandes rasgos se puede afirmar que ambas propuestas generan imágenes mediotonadas altamente semejantes a costos totalmente disímiles.

REFERENCIAS

- [1] H. Aguirre, K. Tanaka, T. Sugimura, and S-Oshita. Halftone image generation with improved multiobjective genetic algorithm. In *EMO*, pages 501–515, 2001.
- [2] Robert W. Floyd and Louis Steinberg. An adaptive algorithm for spatial grayscale. In *Proceedings of the Society for Information Display 17*, volume 2, pages 75–77, 1976.
- [3] A. Grama, A. Gupta, G. Karypis, and V. Kumar. *Introduction to Parallel Computing*. Addison- Wesley, 2003.
- [4] Timothy A. Grogan. Image quality evaluation with a contour-based perceptual model. *Human Vision, Visual Processing and Digital Display III*, 1666, 1992.
- [5] D.E. Knuth. Digital halftone by dot diffusion. In *ACM transaction on Graphics*, volume 6, pages 245–273. ACM Collection, October 1987.
- [6] P. Metaxas. Parallel digital halftoning by error-diffusion. In *ACM International Conference Proceeding Series*, volume 41, pages 35–41, 2003.
- [7] A.C. Naiman and D.T.W. Lam. Error diffusion: Wavefront traversal and contrast considerations. In *Computer Graphics (SIGGRAPH '96 Proceeding)*, volume 19, pages 78–86, 1996.
- [8] V. Ostromoukhov, R.D. Hersch, and I. Amidror. Rotated dispersed dither: A new technique for digital halftoning. In *Computer Graphics (SIGGRAPH 94 Proceedings)*, pages 123–130, 1994.
- [9] J. Parker. *Algorithms for Image Processing and Computer Vision*. J. Wiley & Sons, 1997.
- [10] S. Pelagatti. *Structured Development of Parallel Programs*. Taylor& Francis, 1998.
- [11] H. Saito and N. Kobayashi. Evolutionary computation approach to halftoning algorithm. In *Proc. of the First IEEE Conf. on Evolutionary Computation*, volume 2, pages 787–791, June 1994.
- [12] M. Snir, S. Otto, S. Huss-Lederman, D. Walker, and J. Dongarra. *MPI: The complete Reference*. MIT Press, 1996.
- [13] R. Ulichney. A review of halftoning techniques. *SPIE - International Society for Optical Engineering*, 3963:378–391, 2000.
- [14] L. Velho and J. de Miranda Gomes. Digital halftoning with space filling curves. In *Computer Graphic (SIGGRAPH '91 Proceedings)*, volume 25, pages 81–90, July 1991.
- [15] O. Veryovka, A. Founier, and J. Buchanan. Muti-scale edge analysis for halftoned images. In *Proceedings, SPIE-The International Society for Optical Engineering: Human Vision, Visual Processing, and Digital Display VIII*, January 1998.
- [16] B. Wilkinson and M. Allen. *Parallel programming: Techniques and Application using Networked Workstations*. Prentice-Hall, 1997.

7. APÉNDICE

Imágenes y tablas de evaluación cualitativa y cuantitativa.



(a) *MDEP*



(b) *MDEP_TD*

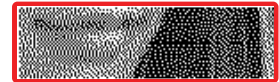
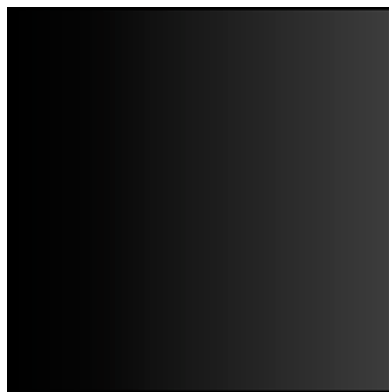
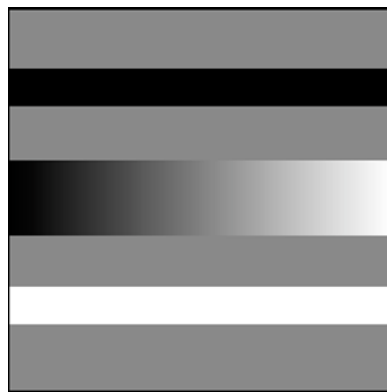


Figura 12: Imágenes Mediotonadas para Evaluación Cualitativa



(a) Rampa del 100%-75% negro



(b) Imagen de bandas

Figura 13: Imágenes para la Evaluación Cuantitativa

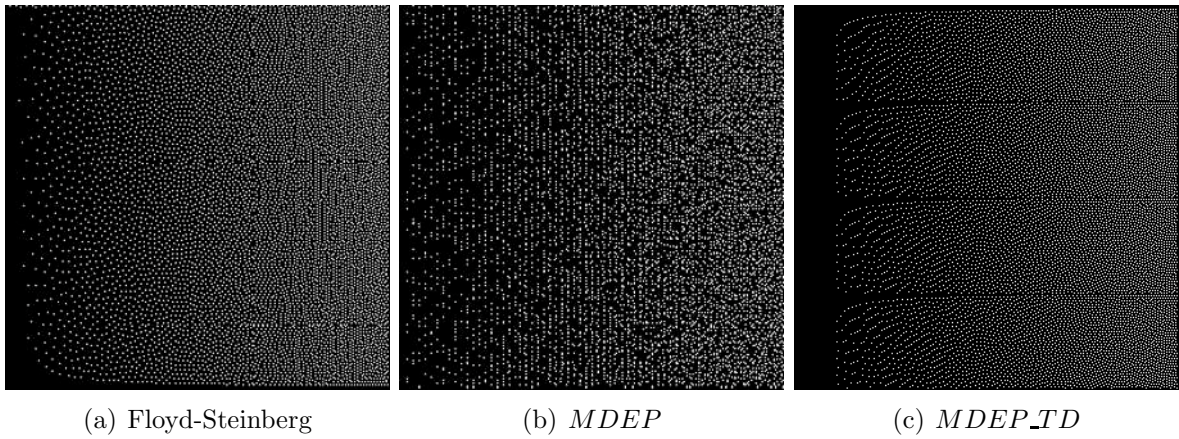


Figura 14: Imágenes Mediotonadas para Evaluación de Incorporación de Artificios

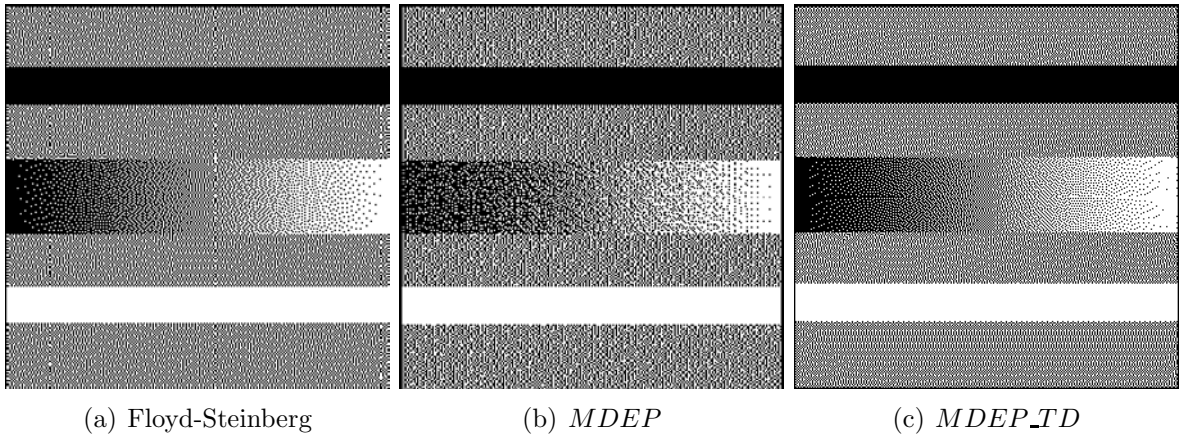


Figura 15: Imágenes Mediotonadas para Evaluación de Preservación de Bordes

Tabla 1: Incorporación de Artificios

Técnica	Valores	Escala 1	Escala 2	Escala 3	Escala 4	Escala 5
Floyd-Steinberg	Min.	85.166	1.774	0.171	0.560	0.015
	Promedio	103.979	14.406	5.257	5.728	3.022
	Max.	120.443	33.336	26.964	23.674	21.853
<i>MDEP</i>	Min.	26.719	8.871	0.484	0.619	0.220
	Promedio	101.298	16.273	5.439	5.627	5.285
	Max.	120.443	31.937	20.822	18.303	16.862
<i>MDEP_TD</i>	Min.	14.361	1.774	2.065	1.619	0.219
	Promedio	98.352	22.127	8.655	8.145	3.107
	Max.	120.443	44.357	22.134	16.684	16.016

Tabla 2: Preservación de Bordes

Técnica	Mínimo	Promedio	Máximo
Floyd-Steinberg	-5.779	3.289	83.600
<i>MDEP</i>	-5.679	5.463	79.723
<i>MDEP_TD</i>	-6.624	4.148	71.746