

ARQUITECTURA DE COMPUTADORES	
UNIVERSIDAD ICESI	Docente: Duvan García
Facultad de Ingeniería	Periodo: Enero – Mayo de 2020
Departamento de Tecnologías Información y Comunicación	Proyecto final
Integrantes	Mateo Gallego Cristian Gironza Paola Veloza N

PROYECTO FINAL ARQUITECTURA DE COMPUTADORES

RESUMEN

El siguiente proyecto se realiza con el objetivo de cuantificar y explicar el impacto en el desempeño que tiene 5 versiones equivalentes de algoritmos en varían en la localidad espacial al recorrer sus píxeles, teniendo en cuenta también el tamaño y la profundidad. Este proyecto se realiza con código desarrollado en lenguaje C#, apoyado por las herramientas visual studio, excel y minitab.

Para el desarrollo de este proyecto se analizaron cinco versiones de código brindadas en la guía, se recopilaron datos y posteriormente se analizaron las relaciones e interacciones entre las variables algoritmo, tamaño, imagen y profundidad. También, este análisis se desarrolla sobre tres imagen escogidas por un integrante del equipo, las cuales fueron obtenidas de google.

Con la información recopilada, se obtuvo un gráfico que relaciona el tiempo de ejecución respecto a la variable algoritmo, tamaño, imagen y profundidad. Seguidamente, se obtuvo una gráfica donde se evidencia la interacción entre las variables algoritmo, tamaño, imagen y profundidad. Posteriormente se realiza un análisis de varianzas, donde se puede evidenciar el valor P de dichas variables y de sus relaciones.

INTRODUCCIÓN

Los diseños de experimentos desempeñan un papel muy importante en el diseño y desarrollo de ingeniería y en la fase de validación de un servicio, modelo o desarrollo de un componente software o hardware de un sistema en condiciones más cercanas a los escenarios de trabajo. El diseño de experimentos (DoE) es una técnica formal y estructurada para estudiar cualquier situación. Eso implica una respuesta que varía en función de una o más variables independientes. El DoE está diseñado específicamente para abordar problemas complejos donde más de una variable puede afectar una respuesta y dos o más variables pueden interactuar entre sí. [1]

El proyecto final tiene como objetivo la aplicación del diseño y análisis de experimentos con el fin de evaluar el desempeño de invertir el color de una matriz que representa una imagen de mapa de bits.

Cuantificar y explicar el impacto en el desempeño que tiene 5 versiones equivalentes de algoritmos en varían en la localidad espacial al recorrer sus píxeles. También se desea estudiar el impacto del tamaño de las imágenes en el tiempo de respuesta normalizado en la ejecución de la operación de invertir el color. Otra variable de interés es la afectación de la profundidad de color en el tiempo de ejecución, para ello se utilizará imágenes de 8, 16 y 32 bits por canal RGB.

Las hipótesis planteadas inicialmente es que la información de factor no afecta el tiempo de ejecución del código, segundo, entre las variables fundamentales no existe ninguna interacción para el tiempo de ejecución. Es importante tener en cuenta que el nivel de confianza utilizado es del 95%. De esta forma La hipótesis nula se acepta con valores p mayores al alfa de 0.05.

DESARROLLO DE LA PRÁCTICA DE SIMULACIÓN

Para el desarrollo de esta práctica fue requerido el análisis de 5 algoritmos escritos en C# brindados por la guía del proyecto, los cuales tenían, cada uno, la finalidad de invertir los colores de una imagen. Dentro de este análisis uno de los factores más importantes fue el tiempo, esto con la finalidad de medir cuál algoritmo era el más veloz bajo distintas circunstancias.

El primer algoritmo se encargaba de recorrer el mapa de bits fila por fila, invirtiendo los canales rojo, verde y azul en el mismo recorrido, empezando siempre por el píxel con posición 0,0:

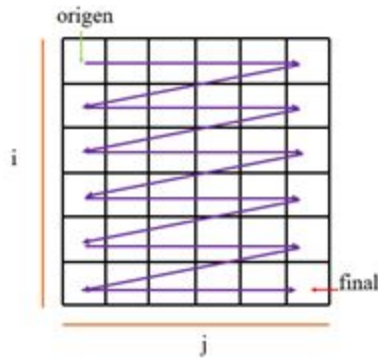


Figura 1. recorrido algoritmo 1.

El segundo algoritmo funciona de la misma forma que el algoritmo 1, con la diferencia de que hace un recorrido por cada canal de color, recorriendo e invirtiendo primero el canal rojo, después el verde y por último el azul; empezando siempre por el píxel con posición 0,0:

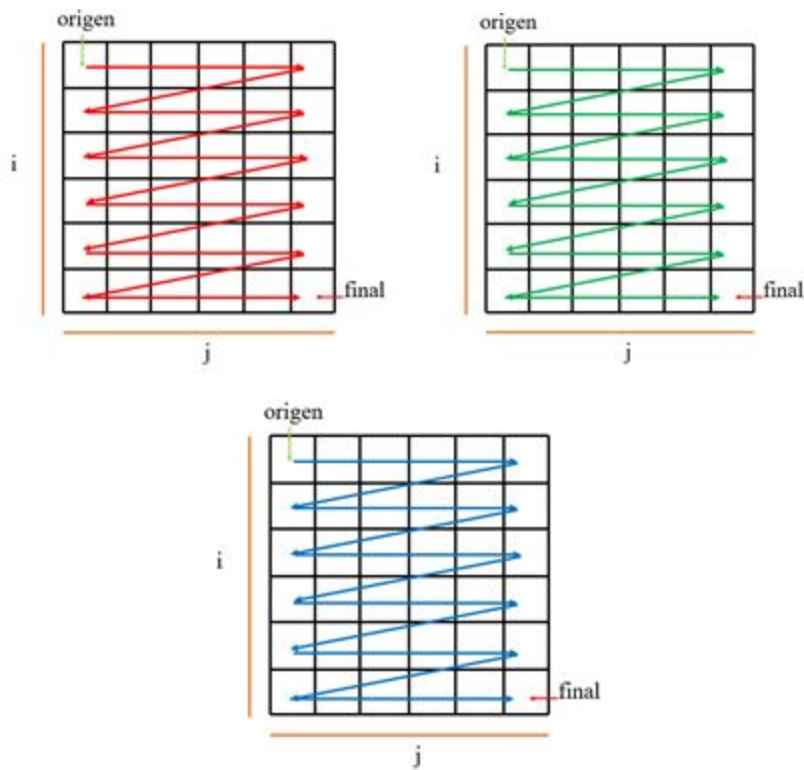


Figura 2. Recorrido algoritmo 2.

El tercer algoritmo se encarga de recorrer el mapa de bits por columnas, invirtiendo los canales rojo, verde y azul de cada píxel en cada recorrido. Empezando siempre por el píxel con posición 0,0:

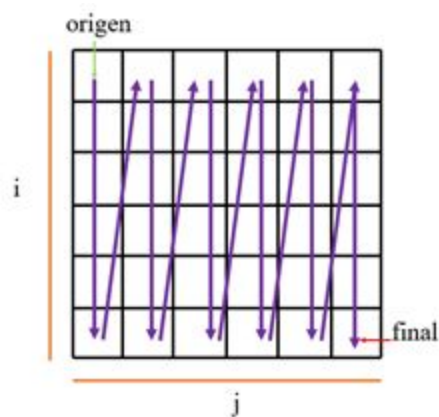


Figura 3. Recorrido algoritmo 3.

El cuarto algoritmo recorre primero por filas el canal rojo siendo este invertido, partiendo desde el píxel en la posición 0,0. Después, en un segundo ciclo se recorren e invierten los canales de color verde y azul, estos se recorrerán por filas partiendo desde el píxel con posición $i-1, j-1$:

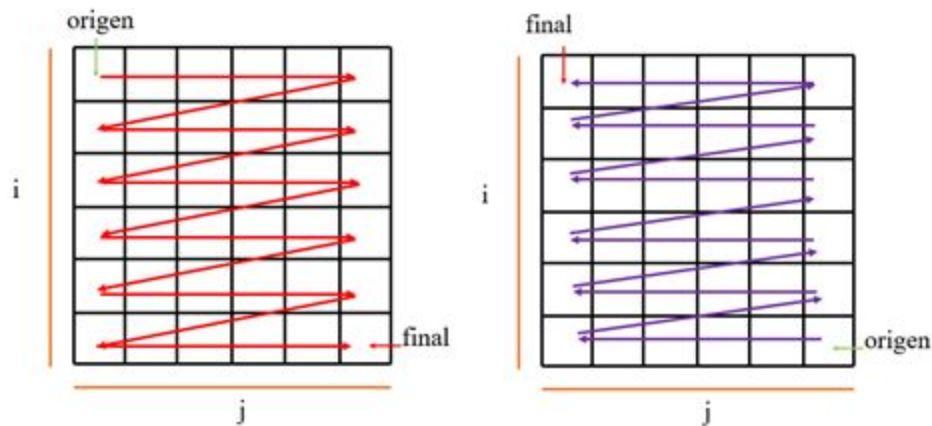


Figura 4. Recorrido algoritmo 4.

El último algoritmo se encarga de recorrer e invertir los colores de los píxeles por conjuntos. Es decir dentro del mismo ciclo se invierten los colores del píxel con posición (i,j) y sus adyacentes $(i+1,j)$, $(i,j+1)$, $(i+1,j+1)$:

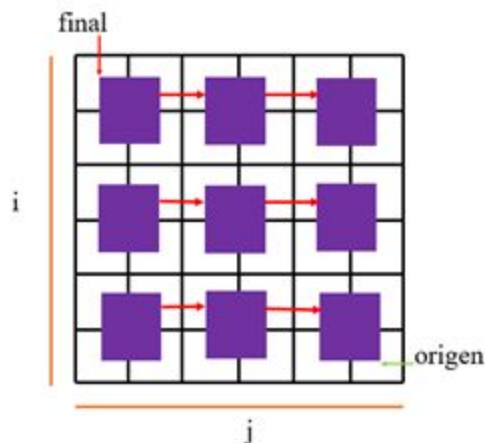


Figura 5. Recorrido algoritmo 5.

Ahora bien, teniendo en cuenta como funciona cada algoritmo los podemos poner a prueba bajo distintas circunstancias. Como fue dicho anteriormente, en esta práctica

se busca evaluar los algoritmos con respecto a distintos tamaños de imagen y la profundidad de color. para ello se escogieron 3 imágenes distintas, cada una de ellas vendrá en 4 tamaños: 64x64, 160x160, 512x512, 1500x1500. Cada uno de estos tamaños también será probado con una profundidad de color de 8 bits y luego con 16 bits. A continuación se presentan las imágenes seleccionadas:



Figura 6. Imágenes utilizadas para la experimentación.

Para cada imagen y sus distintas profundidades y tamaños se busca realizar 100 pruebas por cada algoritmo, para que sean tomadas las mediciones de cuánto tarda cada uno en ejecutarse, para así, analizar qué tanto influyen el tamaño de la imagen y su profundidad en el desempeño de cada uno de los algoritmos.

De esta forma, se consideraron cuatro variables fundamentales (algoritmo, imagen, tamaño y profundidad) las cuales se evidencian en la figura 6. Posteriormente, estos datos se fueron recopilando en tablas de excel como se evidencia en la figura 7.

Información del factor

Factor	Tipo	Niveles	Valores
Algoritmo	Fijo	5	1; 2; 3; 4; 5
Tamaño	Fijo	4	64; 160; 512; 1500
Imagen	Fijo	3	1; 2; 3
Profundida	Fijo	2	8; 16

Figura 6. Variables en consideración.

Algoritmo▼	Tamaño▼	Imagen▼	Profundid▼	Tiempo▼	Normalizad▼
1	64	1	8	7811500	1907,104492
1	64	1	8	7665200	1871,386719
1	64	1	8	7257200	1771,777344
1	64	1	8	7152000	1746,09375
1	64	1	8	8531000	2082,763672
1	64	1	8	10265400	2506,201172
1	64	1	8	7242000	1768,066406
1	64	1	8	8093199	1975,878662
1	64	1	8	8565100	2091,088867

Figura 7. Recopilación de datos.

RESULTADOS

Esta sección es la más importante del informe. Se presentan y explican ordenadamente los resultados obtenidos, siguiendo la secuencia indicada en el desarrollo de la práctica, se deben apoyar con tablas y gráficas.

Por medio de la herramienta brindada en clase Minitab se desarrollaron los análisis de los resultados. En donde inicialmente se obtuvo el análisis de varianza representado en la figura 8. En donde se puede evidenciar el valor P para poder concluir respecto a la hipótesis planteada.

Análisis de Varianza

Fuente	GL	SC Ajust.	MC Ajust.	Valor F	Valor p
Algoritmo	4	18533439937	4633359984	40755,46	0,000
Tamaño	3	267607777	89202592	784,63	0,000
Imagen	2	14813212	7406606	65,15	0,000
Profundida	1	1636377715	1636377715	14393,73	0,000
Algoritmo*Imagen	8	24406496	3050812	26,84	0,000
Algoritmo*Profundida	4	401591336	100397834	883,11	0,000
Tamaño*Imagen	6	8811217	1468536	12,92	0,000
Tamaño*Profundida	3	3491215	1163738	10,24	0,000
Imagen*Profundida	2	1751309	875655	7,70	0,000
Algoritmo*Imagen*Profundida	8	16297540	2037192	17,92	0,000
Tamaño*Imagen*Profundida	6	4554500	759083	6,68	0,000
Error	10752	1222360984	113687		
Falta de ajuste	60	127028951	2117149	20,67	0,000
Error puro	10692	1095332032	102444		
Total	10799	22304342334			

Figura 8. Análisis de varianza.

Debido a diversas fallas no se pudo realizar el análisis de la interacción entre las variables justificas en la figura 9. Esto mismo es notificado por la herramienta minitab.

No se pueden estimar los siguientes términos y se eliminaron:
Algoritmo*Tamaño; Algoritmo*Tamaño*Imagen; Algoritmo*Tamaño*Profundida; Algoritmo*Tamaño*Imagen*Profundida

Figura 9. Términos que no se entraron en consideración.

Seguidamente, se realiza la gráfica de efectos principales para la normalización, en donde se entra a evaluar el tiempo respecto a cada variable fundamental en consideración. Esto se puede evidenciar en la figura 10.

Gráfica de efectos principales para Normalizado (ns)

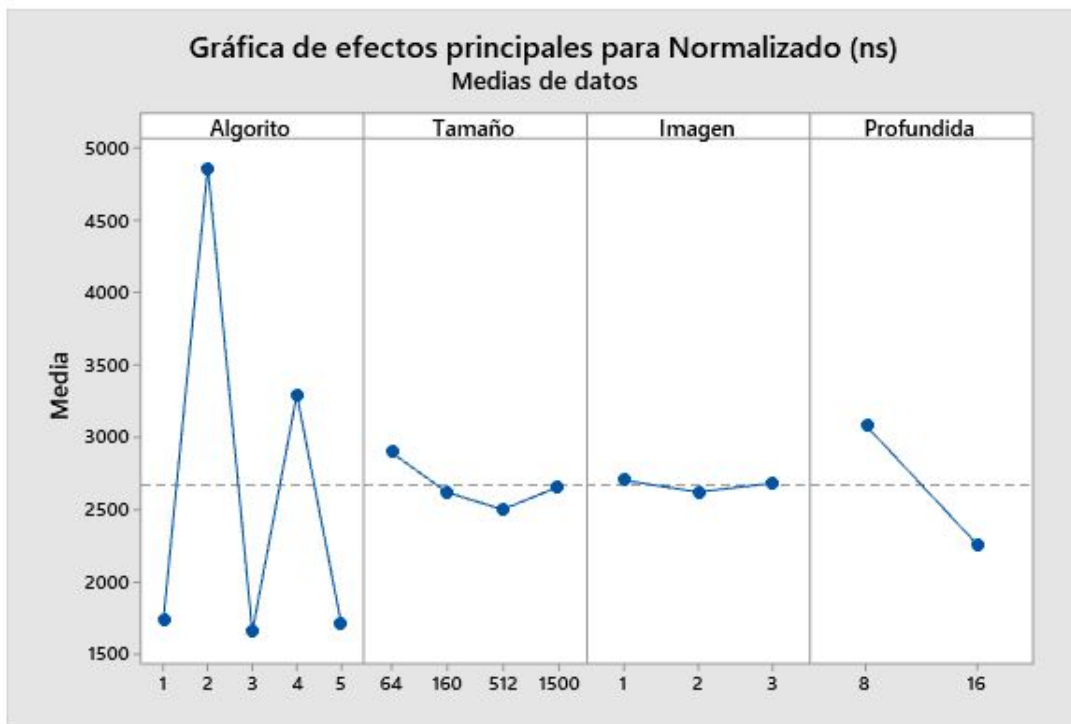


Figura 10. Tamaño vs Variable.

Por último, se realiza la se realiza la gráfica de interacción para normalizado, en donde se puede evidenciar en ilustraciones si al ser paralelas no tienen relación o si al ser perpendiculares si tienen interacción. Esta gráfica se evidencia en la figura 11.

Gráfica de interacción para Normalizado (ns)

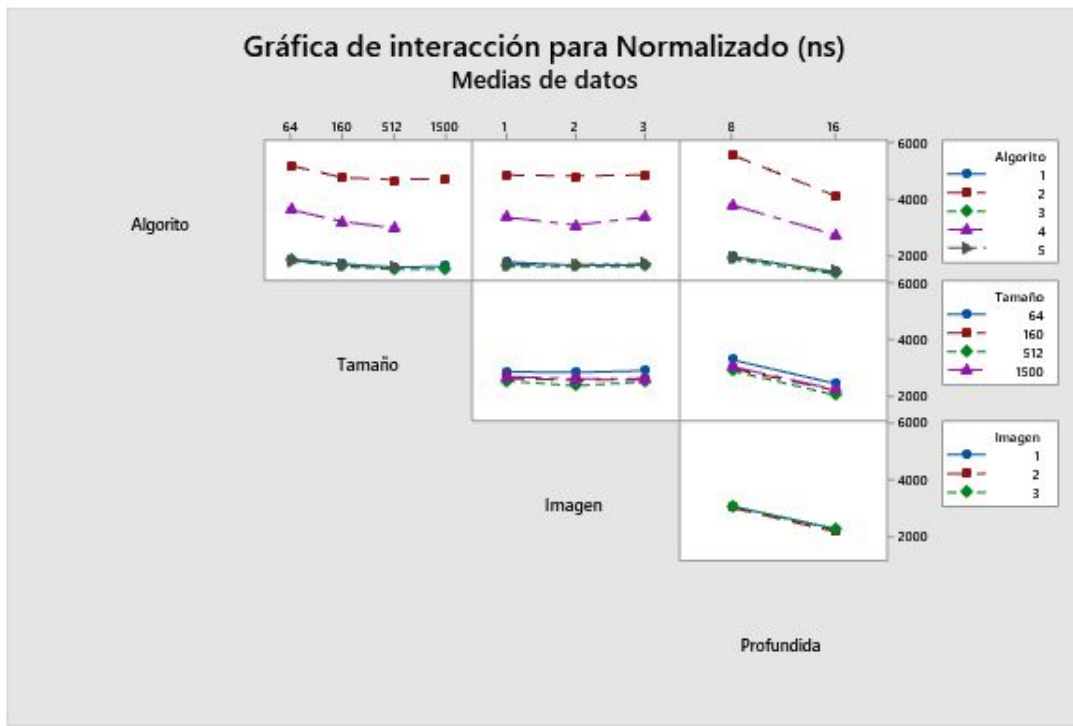


Figura 11. Gráficas de interacciones.

CONCLUSIONES

- Respecto a la figura 8, se puede llegar a que las variables algoritmo, tamaño, profundidad e imagen, junto con las interacciones, algoritmo * imagen, algoritmo * profundidad, tamaño * imagen, tamaño * profundidad, imagen * profundidad, algoritmo * imagen * profundidad, y tamaño * imagen * profundidad, tienen un valor P inferior a 0.05 por lo cual se rechaza la hipótesis nula.
- Respecto a la figura 9, es probable que las variables sí interactúen, pero debido a el nivel de complejidad de complejidad del algoritmo 4 y 5 consumió todos los recursos, o sea que nuestra unidad experimental no funcionó, no se puede concluir respecto a la interacción de estas variables.
- Respecto a la figura 10, teniendo en cuenta la relación tiempo vs algoritmo, el algoritmo 1, 3 y 5 son mejores porque aprovechan el principio de espacialidad

y la memoria ram por eso se puede concluir que tanto su miss rate y su ammat son mejor que el algoritmo 2 y 4. Respecto al tamaño vs tiempo la cantidad de espacios que carga del cache es óptimo al tamaño del mapa de bits de la imagen 512 x 512. Respecto a la imagen vs tiempo no tenemos control aunque afecta. Respecto a la profundidad vs tiempo, consume menos tiempo con profundidad 16 bits RGB esto debido a que la unidad experimental tiene mejor procesamiento de este tipo de imagen.

- Respecto a la figura 11, se puede ver que las variables algoritmo vs tamaño no tienen interacción directamente, pero se podría afirmar que puede haber un punto en que el que sí se intersectan. Respecto a algoritmo vs imagen, se observa que no hay interacción. Respecto a tamaño vs Imagen no hay interacción; respecto a algoritmo y profundidad, no hay interacción, aunque se podría creer que puede llegar un punto en el que si se interactúen. Respecto a tamaño vs profundidad, no hay interacción y finalmente respecto a imagen y profundidad no se evidencia interacción.

REFERENCIAS BIBLIOGRÁFICAS

- [1] Y. Motivacion, "Diseño de Experimentos-Terminología y Conceptos Arquitectura de Computadores / Interfaces y Arquitectura Hardware."

Adobe. (14 de mayo de 2018). Acerca de las imágenes de mapa de bits. Obtenido de <https://helpx.adobe.com/es/photoshop/using/image-essentials.html>

<https://docs.microsoft.com/>. (s.f.). Tipos de mapas de bits. Obtenido de <https://docs.microsoft.com/es-es/dotnet/framework/winforms/advanced/types-of-bitmaps>

Imagenes: Autoría de imágenes de múltiples medios (buscador de google).