

Disminución en resoluciones de imágenes

Programación paralela laboratorio 1

Vladimir Vargas Diaz, José Miguel Carvajal Jimenez , Alci Rene Ramirez Soto

Abstract—This document presents the effect of decreasing resolution using the openCV library and the C++ language, through the use of multiple posix threads and the openMP tool, this work focuses on 3 different resolutions (720p, 1080p and 4k). You will find the results of: Response time of the algorithm in each resolution, and graph of speed up. The implemented algorithm is bilinear interpolation.

Index Terms—Algorithms; interpolation; images; openCV; reduction; threads.

1 RESUMEN

Este documento presenta el efecto de disminución de resolución usando la librería openCV y el lenguaje C++, mediante el uso de múltiples hilos posix y la herramienta openMP, este trabajo se centra en 3 diferentes resoluciones (720p, 1080p y 4k). Encontrará los resultados de: Tiempo de respuesta del algoritmo en cada resolución, y gráfica de speed up. El algoritmo implementado es el de interpolación bilineal. Palabras clave - Algoritmos; hilos; interpolación; imágenes; openCV; reducción.

2 INTRODUCCIÓN

En el mundo actual el manejo de imágenes y generación de contenidos basados en ellas es fundamental y muy usado, sea por empresas, personas afines a contenidos multimedia, personas que tienen un hobby en creación de imágenes entre otros, esta usabilidad y versatilidad de compartir información por un medio visual, llama la atención de nuevas tecnologías y herramientas de desarrollo, nuevos algoritmos así como también nuevas formas de interpretar, organizar y ver este contenido.

Entonces la forma en la que nosotros compartimos y vemos imágenes, sea por medio de dispositivos móviles como tabletas, smartphones o computadores de escritorio influye mucho en este tipo de contenido, esto se debe a la resolución de pantalla de estos dispositivos, pero ¿qué es la resolución de una imagen? Básicamente es la calidad de la imagen medida por la cantidad de píxeles que esta tiene cuando multiplicamos su ancho por alto.

El proceso en el que se centra este documento es en el cambio de esas resoluciones, para este cambio en los tamaños de imágenes es conocido el método de interpolación por aproximación, se usa para adaptar cambiar o mejorar los tamaños de las imágenes para tener de ellas una

mejor forma de verlas, es decir cuando pasamos de una alta resolución a baja resolución o viceversa.

Uno de los problemas más comunes de hacer interpolación de imágenes es la deformación del contenido o la disminución del contenido.

Las técnicas más conocidas para manejar la resolución de una imagen son las siguientes:

- 1) Bicúbica
- 2) Bicúbica enfocada
- 3) En escalera
- 4) Bilineal
- 5) Por aproximación

3 METODOLOGÍA

El desarrollo del trabajo se hizo mediante la aplicación de una técnica de interpolación expuestas en el punto anterior, se tomó esta decisión basado las siguientes definiciones.

Bicúbica: Esta interpolación es una de las mejores dado que toma 16 píxeles adyacentes. **Bicúbica enfocada:** Esta interpolación parte de la bicúbica pero se enfoca en arreglar partes de la imagen final. **Escalera:** Este método se enfoca en incrementar usualmente en un 10% del avance anterior la interpolación de cada píxel.

Bilineal: Este método calcula 4 píxeles por lo que su aproximación y riesgo de diseminación es mucho menor.

Por aproximación: Este método repite o descarta píxeles mediante un promedio de los dos píxeles consecuentes a cada lado, se usa más en gráficas que en fotografías.

Otros trabajos o herramientas relacionadas a la reducción de resolución de imágenes son programas como photoshop, pixlr, microsoft paint entre otros.

3.1 MÉTODOS PARA EL MANEJO DE RESOLUCIÓN DE IMÁGENES

3.1.1 Interpolación Bilineal

La interpolación bilineal es el uso del promedio 4 píxeles cercanos a un píxel, para obtener un acercamiento al color y posición que tiene el píxel en cuestión y así con este poder acomodar este nuevo píxel en la resolución objetivo perdiendo la menor cantidad de información posible[1], [2].

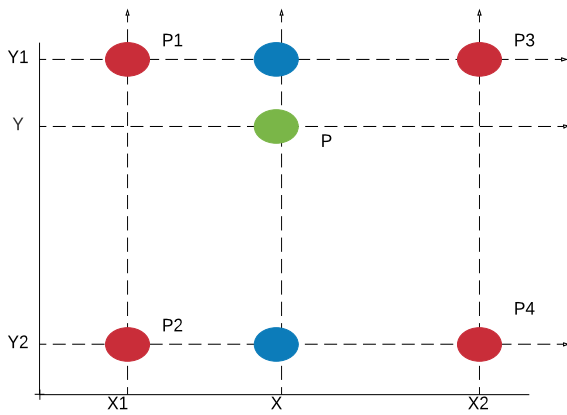


Fig. 1. Representación del promedio de píxeles adyacentes

3.2 OBJETIVOS Y ENTORNOS DE TRABAJO

EL objetivo de este trabajo es la disminución en resolución de imágenes, se usaran 3 resoluciones (720p, 1080p y 4k) para comprobar el procedimiento, se utilizara el algoritmo con 2,4,8 y 16 hilos, se tomaran los tiempos de resultados de ejecución se utilizaran para exponer tiempos de respuesta y speed up del algoritmo en cada resolución para cada numero de hilos usado en el proceso.

4 RESULTADOS

A continuación observaremos 2 figuras asociadas a cada una de las resoluciones propuestas en el laboratorio: 720p, 1080p y 4k, las convenciones serán que los datos presentados en azul serán los del speed up y los datos de amarillo ilustran el tiempo que tarde la ejecución.

4.1 Pthread

En esta figura se observa el speed up y el tiempo de respuesta para la resolución imagen de 720p convertida a 480p.

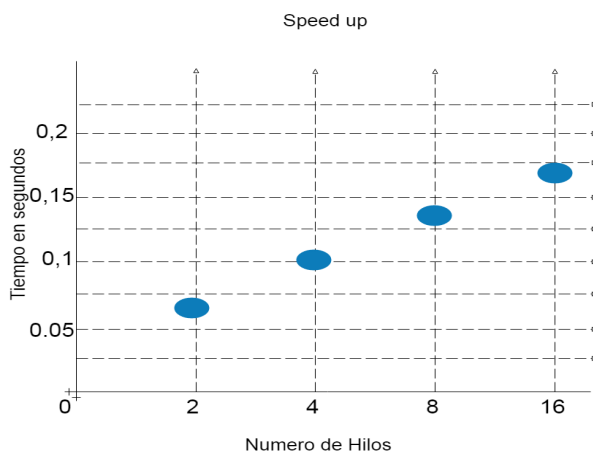


Fig. 2. Speed up de resolución 720p a 480p con posix

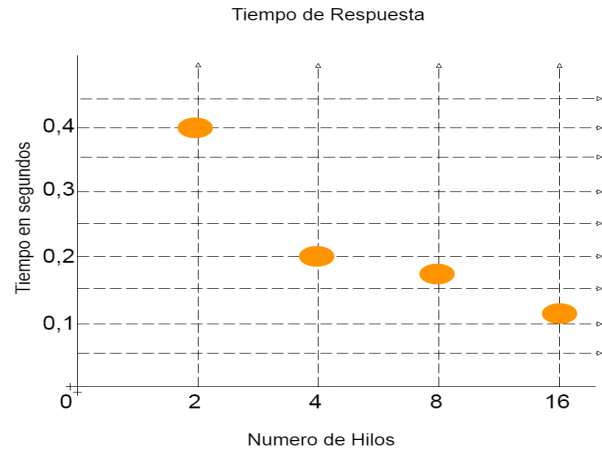


Fig. 3. Tiempo de respuesta de resolución 720p a 480p con posix

Ahora observaremos los resultados para la resolución imagen de 1080p convertida a 480p.

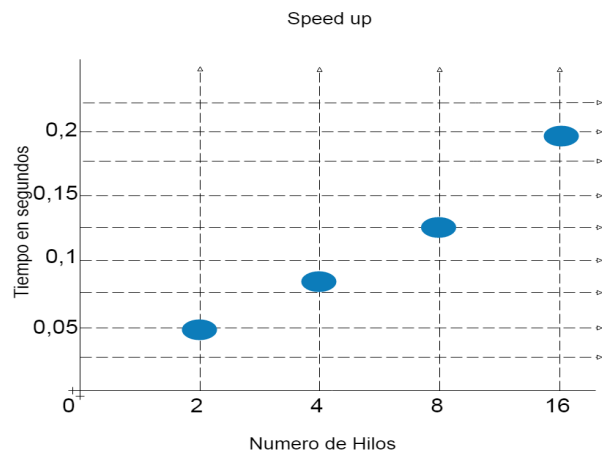


Fig. 4. Speed up de resolución 1080p a 480p con posix

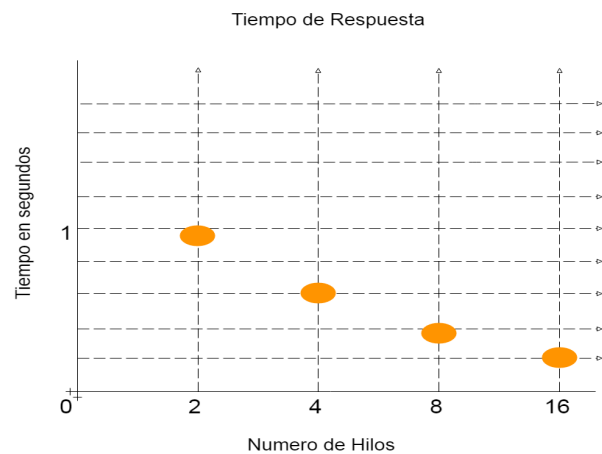


Fig. 5. Tiempo de respuesta de resolución 1080p a 480p con posix

Por ultimo de resultados posix observamos los resultados correspondientes a la imagen de resolución 4k pasada a una resolución de 480p.

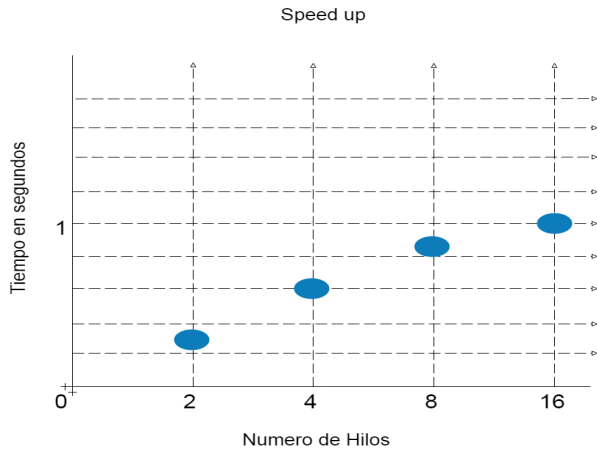


Fig. 6. Speed up de resolución 4k a 480p con posix

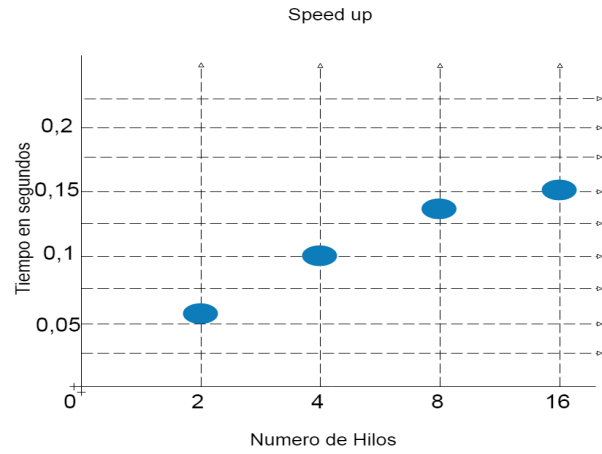


Fig. 8. Speed up de resolución 720p a 480p con openMP

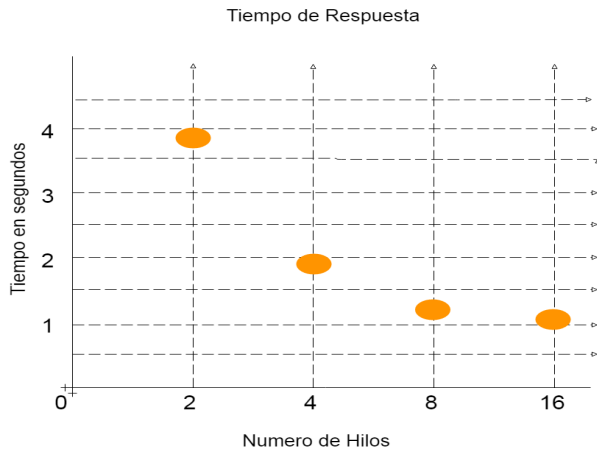


Fig. 7. Tiempo de respuesta de resolución 4k a 480p con posix

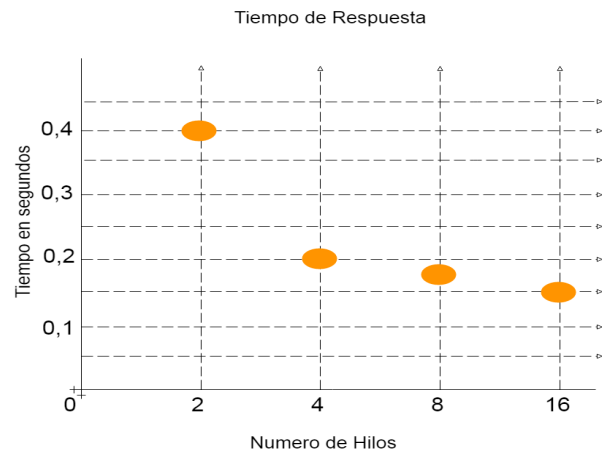


Fig. 9. Tiempo de respuesta de resolución 720p a 480p con openMP

Ahora observaremos los resultados para la resolución imagen de 1080p convertida a 480p.

4.2 OpenMP

Terminado el proceso de ilustrar los resultados con hilos posix procederemos a mostrar los resultados mediante la herramienta openMP.

Al igual que las ilustraciones anteriores se maneja la convención de azul para el speed up y amarillo para el tiempo de respuesta.

Cabe aclarar que las imágenes usadas para los ejemplos de reducción de resolución con hilos posix y la herramienta openMP fueron las mismas.

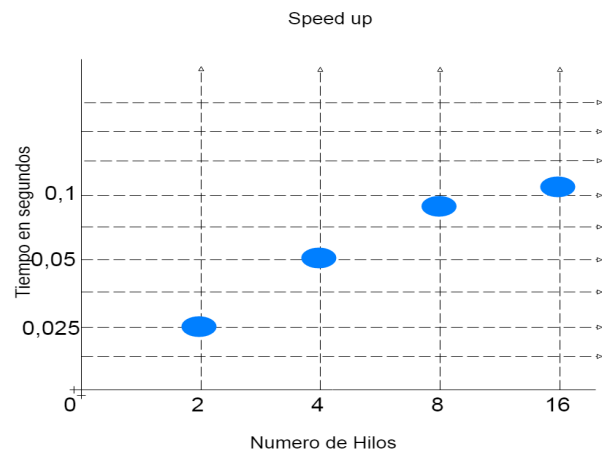


Fig. 10. Speed up de resolución 1080p a 480p con openMP

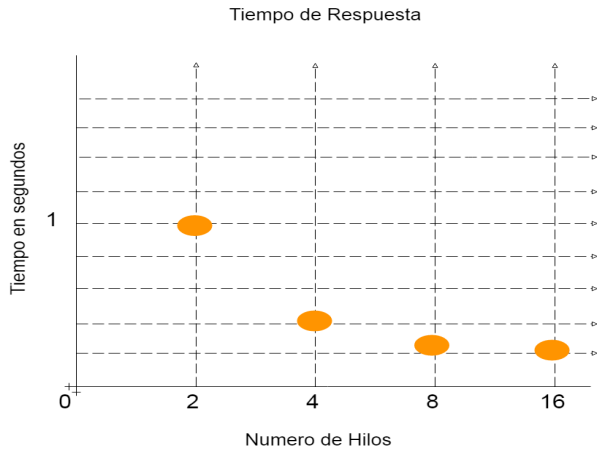


Fig. 11. Tiempo de respuesta de resolución 1080p a 480p con openMP

Por ultimo de resultados posix observamos los resultados correspondientes a la imagen de resolución 4k pasada a una resolución de 480p.

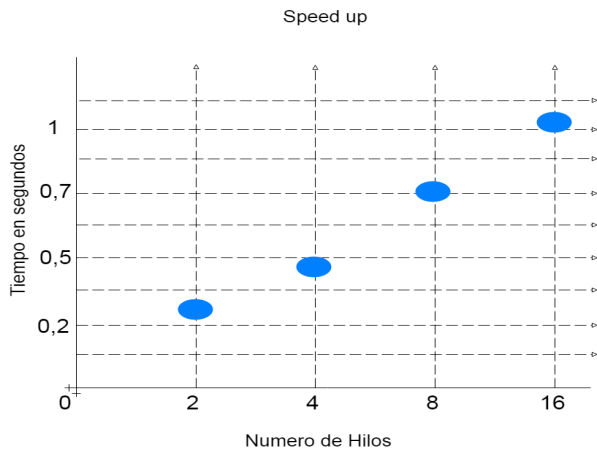


Fig. 12. Speed up de resolución 4k a 480p con openMP

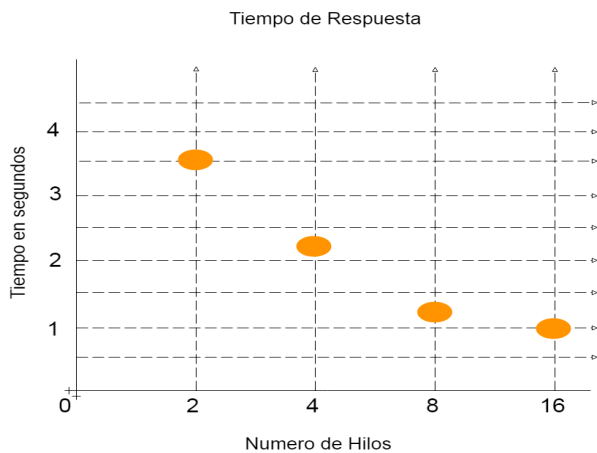


Fig. 13. Tiempo de respuesta de resolución 4k a 480p con openMP

Con esto concluimos la presentación de los resultados de openMP, vemos que los resultados mediante hilos posix

y la herramienta openMP, no es significativa, y en algunos casos estos resultados son muy parecidos.

Al tener esta poca diferencia en tiempos de respuesta y resultados de rendimiento podemos ver que ambas opciones son útiles para el proceso de reducción de resolución en imágenes.

4.3 CUDA

Continuaremos nuestra comparación en reducción de resoluciones mostrando ahora el uso de una herramienta en la nube, Google colab.

En esta herramienta haremos la disminución de la resolución de nuestras 3 imágenes de prueba, utilizaremos nuestro código pero ahora estará dispuesto para ser ejecutado mediante CUDA, el cual nos permitirá tener una nueva arquitectura interna para ejecutar nuestro algoritmo.

Para este caso de CUDA se utilizaran dos casos de prueba, los cuales conservaran el numero de hilos: 2, 4, 8 y 16 respectivamente, pero tendremos una nueva consideración los BLOQUES en los que estos hilos serán ejecutados.

Se mostraran los resultados con las mismas convenciones expuestas en la imágenes pasadas empezando con los resultados de cuda para 20 bloques.

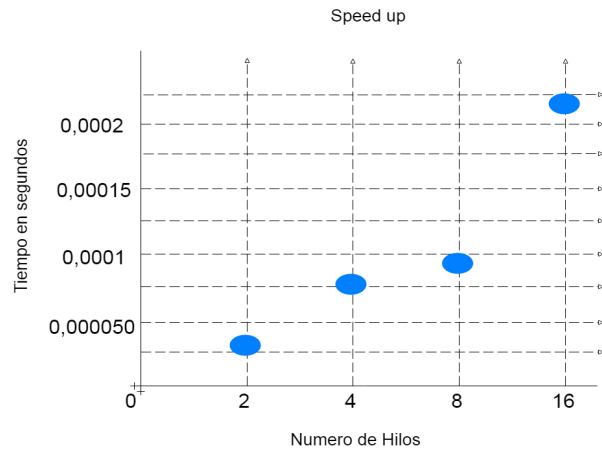


Fig. 14. Speed up de resolución 720 a 480p con cuda 20 bloques

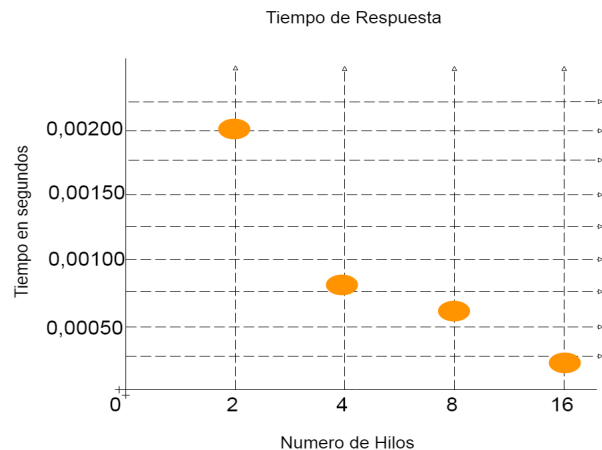


Fig. 15. Tiempo de respuesta de resolución 720 a 480p con cuda 20 bloques

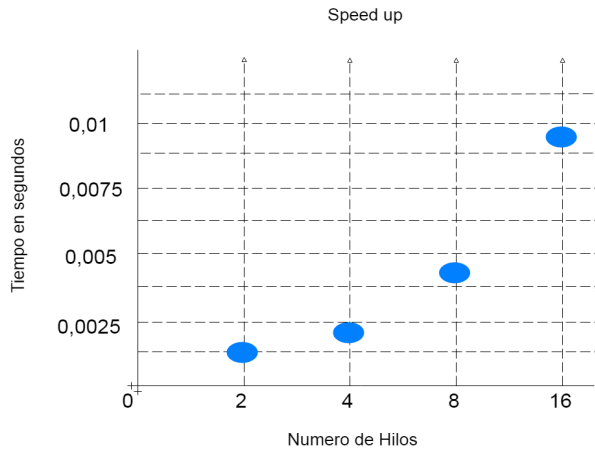


Fig. 16. Speed up de resolución 1080 a 480p con cuda 20 bloques

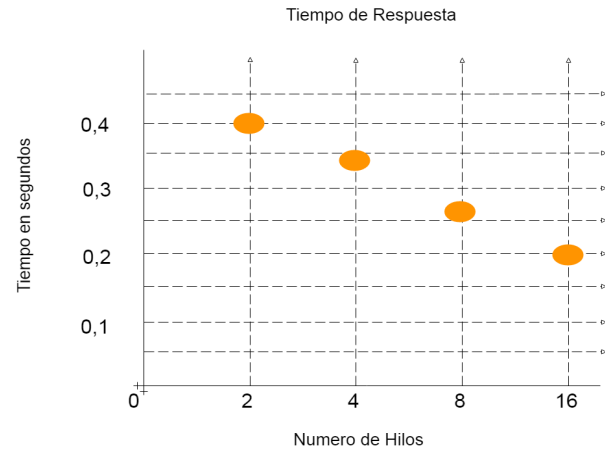


Fig. 19. Tiempo de respuesta de resolución 4k a 480p con cuda 20 bloques

Ahora mostraremos las imágenes de los resultados de cuda, con el manejo de 40 bloques, el numero máximo de bloques fue 40 por que la maquina momentánea que nos presto Google colab fue una "Tesla t40".

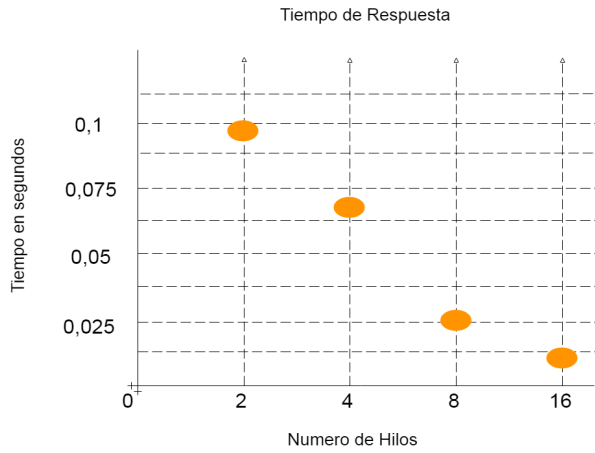


Fig. 17. Tiempo de respuesta de resolución 1080 a 480p con cuda 20 bloques

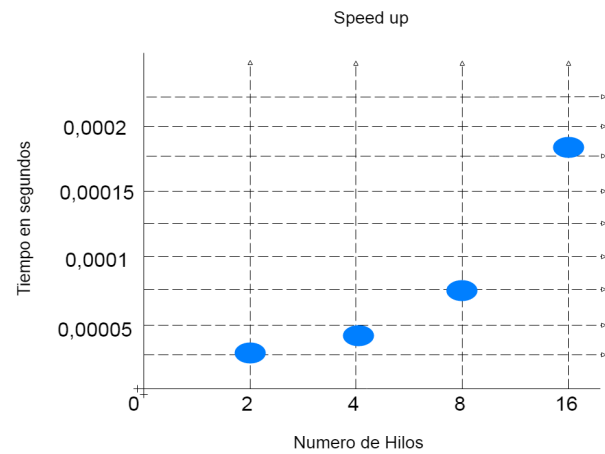


Fig. 20. Speed up de resolución 720 a 480p con cuda 40 bloques

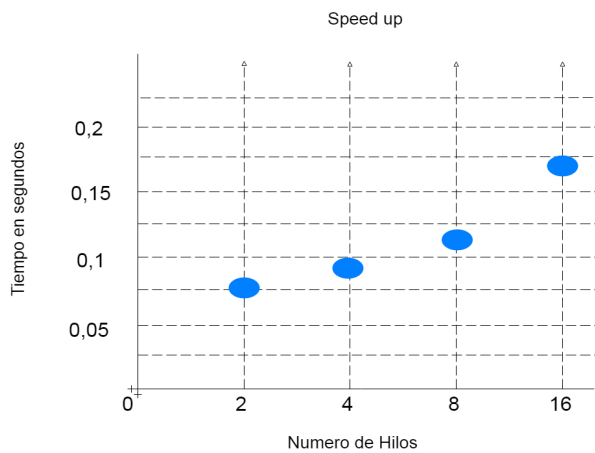


Fig. 18. Speed up de resolución 4k a 480p con cuda 20 bloques

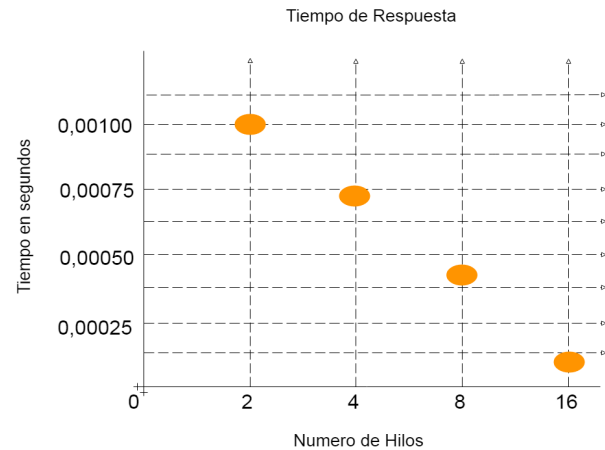


Fig. 21. Tiempo de respuesta de resolución 720 a 480p con cuda 40 bloques

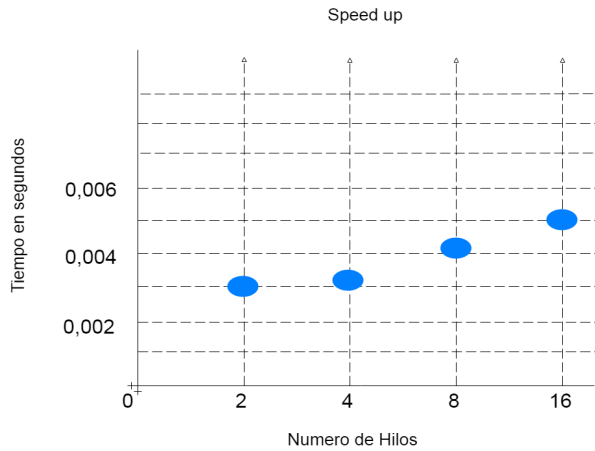


Fig. 22. Speed up de resolución 1080 a 480p con cuda 40 bloques

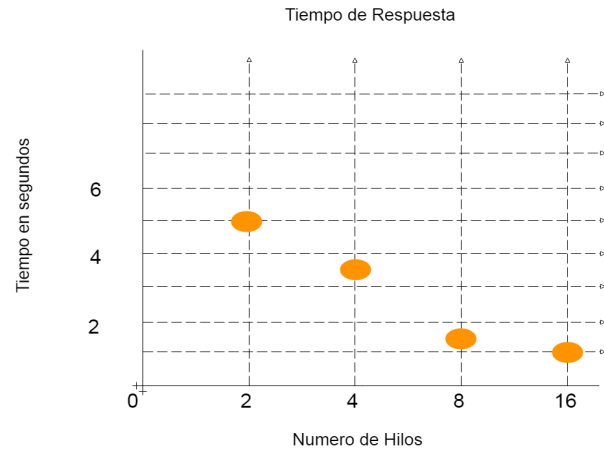


Fig. 25. Tiempo de respuesta de resolución 4k a 480p con cuda 40 bloques

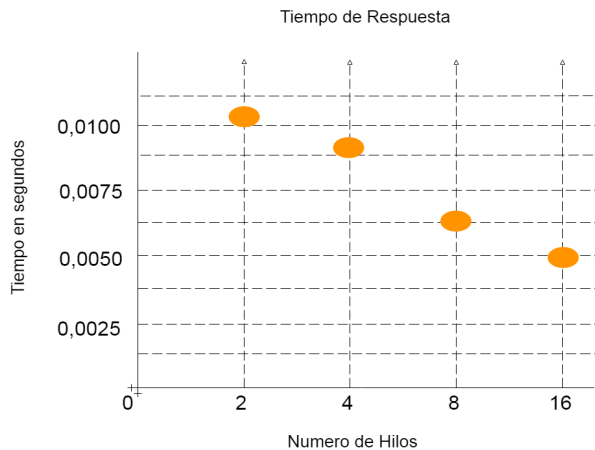


Fig. 23. Tiempo de respuesta de resolución 1080 a 480p con cuda 40 bloques

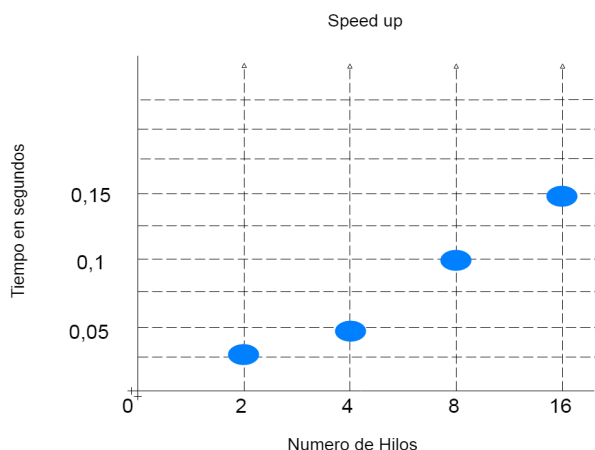


Fig. 24. Speed up de resolución 4k a 480p con cuda 40 bloques

5 CONCLUSIONES

- 1) La disminución en la resolución de imagen lleva consigo posible pérdida de información de la imagen resultado.
- 2) La resolución de una imagen trae ventajas al momento de ser enviada o usada.
- 3) Existen múltiples formas de disminuir la resolución de una imagen sin llegar a perder tanta información.
- 4) Los bordes de las imágenes son los que llegan a definir que tan bueno es el algoritmo de reducción de imagen.
- 5) La diversidad de algoritmos ayuda reconocer en que tipo de imagen sirve, es decir, un algoritmo de disminución de imagen tiene resultados diferentes cuando lo que se quiere disminuir es la resolución de una imagen que contenga un gráfico.
- 6) El uso de Google colab mejoro notablemente el tiempo para el código de CUDA, llevando casi todos los valores a números cercanos a 0.
- 7) El código de CUDA solo se centra en el algoritmo de reducción, como no toma el tiempo de tomar la imagen y copiarla toma menos tiempo que los otros algoritmos.

REFERENCES

- [1] Wikipedia, [https://es.wikipedia.org/wiki/Interpolaci3n_\(fotograf3a\)](https://es.wikipedia.org/wiki/Interpolaci3n_(fotograf3a)), octubre 2020.
- [2] Open course ware, http://ocw.uniovi.es/pluginfile.php/4994/mod_resource/content/5/T3b_Interpolaci3n_imagen_2.pdf Octubre 2020.
- [3] ResolutionAlgorithm, <https://github.com/alreraso/ResolutionAlgorithm> Octubre 2020.
- [4] Cuda Basics, <https://www.nvidia.com/docs/IO/116711/sc11-cuda-c-basics.pdf>