

Proyecto Individual

Desarrollo de una aplicación para la generación de gráficos y texto

Fecha de asignación: 12 marzo 2025
Grupos: 1 persona

Fecha de entrega: 4 abril 2025
Profesores: Luis Chavarría Zamora

Mediante el desarrollo de este proyecto, el estudiante aplicará los conceptos de arquitectura de computadores en el diseño e implementación de una aplicación generación de interpolación de imágenes usando interpolación bilineal. Atributos relacionados: **Análisis de Problemas (AP)**, el cual se encuentra en **Avanzado (A)**.

1. Descripción General: Interpolación en imágenes

Para este proyecto se aplicarán los conceptos de arquitectura de computadoras para el diseño y uso de lenguaje ensamblador para resolver un problema. El algoritmo que se implementará es interpolación bilineal, es recomendada para imágenes de escenarios con valores continuos.

La implementación de dos algoritmos de interpolación se observan en la Figura 1.

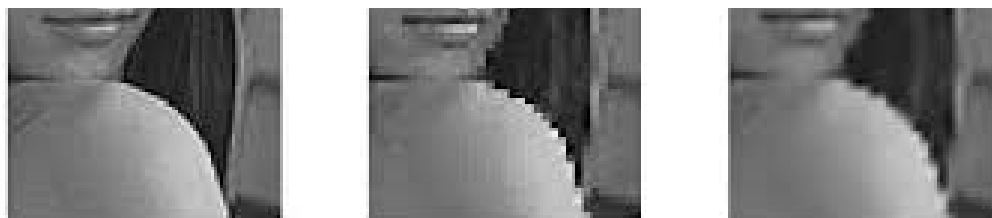


Figura 1: Implementación de algoritmos de interpolación (de izquierda a derecha): *groundtruth*, vecino más cercano y bilineal.

La implementación de este algoritmos se muestra a continuación para la imagen

$$I = \begin{bmatrix} 10 & 20 \\ 30 & 40 \end{bmatrix}, \quad (1)$$

donde cada valor de la matriz es un píxel.

1.1. Interpolación bilineal

El proceso de interpolación bilineal es un poco más complejo pues trata de rellenar el espacio con información continua, no replicando un mismo valor. Por ejemplo, el proceso de interpolación bilineal para la imagen de la ecuación (1). Primero se comienza con los valores que ya se conocen.

1.1.1. Píxeles conocidos

Los píxeles que se usan de referencia son los marcados en **rojo**, los índices se muestran entre paréntesis en color **azul**

$$I' = \begin{bmatrix} \textcolor{red}{10}(\textcolor{blue}{1}) & a(\textcolor{blue}{2}) & b(\textcolor{blue}{3}) & \textcolor{red}{20}(\textcolor{blue}{4}) \\ c(\textcolor{blue}{5}) & d(\textcolor{blue}{6}) & e(\textcolor{blue}{7}) & f(\textcolor{blue}{8}) \\ g(\textcolor{blue}{9}) & h(\textcolor{blue}{10}) & i(\textcolor{blue}{11}) & j(\textcolor{blue}{12}) \\ \textcolor{red}{30}(\textcolor{blue}{13}) & k(\textcolor{blue}{14}) & l(\textcolor{blue}{15}) & \textcolor{red}{40}(\textcolor{blue}{16}) \end{bmatrix}. \quad (2)$$

El procedimiento para obtener los valores de la a a l . Se muestra a continuación.

1.1.2. Píxeles horizontales y verticales

Estos píxeles se calculan como si fuera una interpolación lineal. Se le da un mayor peso a los valores más cercanos al conocido. Los valores horizontales y verticales se muestran a continuación (son operaciones con enteros):

1. $a = \frac{4-\textcolor{red}{2}}{4-1} \times 10 + \frac{\textcolor{red}{2}-1}{4-1} \times 20 = \frac{2}{3} \times 10 + \frac{1}{3} \times 20 = 13$
2. $b = \frac{4-\textcolor{red}{3}}{4-1} \times 10 + \frac{\textcolor{red}{3}-1}{4-1} \times 20 = \frac{1}{3} \times 10 + \frac{2}{3} \times 20 = 17$
3. $c = \frac{13-\textcolor{red}{5}}{13-1} \times 10 + \frac{\textcolor{red}{5}-1}{13-1} \times 30 = \frac{2}{3} \times 10 + \frac{1}{3} \times 30 = 17$
4. $g = \frac{13-\textcolor{red}{9}}{13-1} \times 10 + \frac{\textcolor{red}{9}-1}{13-1} \times 30 = \frac{1}{3} \times 10 + \frac{2}{3} \times 30 = 23$
5. $k = \frac{16-\textcolor{red}{14}}{16-13} \times 30 + \frac{\textcolor{red}{14}-13}{16-13} \times 40 = \frac{2}{3} \times 30 + \frac{1}{3} \times 40 = 33$
6. $l = \frac{16-\textcolor{red}{15}}{16-13} \times 30 + \frac{\textcolor{red}{15}-13}{16-13} \times 40 = \frac{1}{3} \times 30 + \frac{2}{3} \times 40 = 37$
7. $f = \frac{16-\textcolor{red}{8}}{16-4} \times 30 + \frac{\textcolor{red}{8}-4}{16-4} \times 40 = \frac{2}{3} \times 20 + \frac{1}{3} \times 40 = 27$
8. $j = \frac{16-\textcolor{red}{12}}{16-4} \times 30 + \frac{\textcolor{red}{12}-4}{16-4} \times 40 = \frac{1}{3} \times 20 + \frac{2}{3} \times 40 = 33$

Para mayor guía se rellenan los valores faltantes en

$$I' = \begin{bmatrix} \textcolor{red}{10}(\textcolor{blue}{1}) & 13(\textcolor{blue}{2}) & 17(\textcolor{blue}{3}) & \textcolor{red}{20}(\textcolor{blue}{4}) \\ 17(\textcolor{blue}{5}) & d(\textcolor{blue}{6}) & e(\textcolor{blue}{7}) & 27(\textcolor{blue}{8}) \\ 23(\textcolor{blue}{9}) & h(\textcolor{blue}{10}) & i(\textcolor{blue}{11}) & 33(\textcolor{blue}{12}) \\ \textcolor{red}{30}(\textcolor{blue}{13}) & 33(\textcolor{blue}{14}) & 37(\textcolor{blue}{15}) & \textcolor{red}{40}(\textcolor{blue}{16}) \end{bmatrix}. \quad (3)$$

1.1.3. Píxeles intermedios

Estos píxeles se calculan usando los píxeles verticales y horizontales. Igual que el método anterior se le da mayor peso a los píxeles cercanos, en este caso los recién calculados en el paso anterior, como se observa en

$$I' = \begin{bmatrix} 10(1) & 13(2) & 17(3) & 20(4) \\ 17(5) & d(6) & e(7) & 27(8) \\ 23(9) & h(10) & i(11) & 33(12) \\ 30(13) & 33(14) & 37(15) & 40(16) \end{bmatrix}. \quad (4)$$

Para interpolar los píxeles d , e , h , i . Para estos valores se puede inferir ya sea horizontal o verticalmente, sin importar el eje se obtiene el mismo valor. En este caso se infiere horizontalmente:

1. $d = \frac{8-6}{8-5} \times 17 + \frac{6-5}{8-5} \times 27 = \frac{2}{3} \times 17 + \frac{1}{3} \times 27 = 20$
2. $e = \frac{8-7}{8-5} \times 17 + \frac{7-5}{8-5} \times 27 = \frac{1}{3} \times 17 + \frac{2}{3} \times 27 = 24$
3. $h = \frac{12-10}{12-9} \times 23 + \frac{10-9}{12-9} \times 33 = \frac{2}{3} \times 23 + \frac{1}{3} \times 33 = 26$
4. $i = \frac{12-11}{12-9} \times 23 + \frac{11-9}{12-9} \times 33 = \frac{1}{3} \times 23 + \frac{2}{3} \times 33 = 30$

La representación final de la matriz se observa en

$$I' = \begin{bmatrix} 10(1) & 13(2) & 17(3) & 20(4) \\ 17(5) & 20(6) & 24(7) & 27(8) \\ 23(9) & 26(10) & 30(11) & 33(12) \\ 30(13) & 33(14) & 37(15) & 40(16) \end{bmatrix} = \begin{bmatrix} 10 & 13 & 17 & 20 \\ 17 & 20 & 24 & 27 \\ 23 & 26 & 30 & 33 \\ 30 & 33 & 37 & 40 \end{bmatrix}. \quad (5)$$

Esta operación se realiza para cada cuadrante identificado.

1.2. Resultado de interpolación

Se observa que para la imagen

$$I = \begin{bmatrix} 10 & 20 \\ 30 & 40 \end{bmatrix}, \quad (6)$$

y para interpolación bilineal sería

$$I'' = \begin{bmatrix} 10 & 13 & 17 & 20 \\ 17 & 20 & 24 & 27 \\ 23 & 26 & 30 & 33 \\ 30 & 33 & 37 & 40 \end{bmatrix}. \quad (7)$$

2. Especificación

Se le solicita desarrollar en **ensamblador** (asm) un programa que realice el proceso de interpolación bilineal. Se usará una imagen de entrada libre con dimensión mínima de 390×390 (este valor se puede modificar si se habla y justifica con el profesor). En la salida se podrá seleccionar alguno de los siguientes cuadrantes mostrados en la Figura 2. Al cuadrante seleccionado se le aplicará la interpolación bilineal.

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

Figura 2: Cuadrícula de selección de la imagen de entrada

El sistema debe cumplir las siguientes características:

1. El procesamiento de la imagen **SOLO** se realizará en ensamblador.
2. Las imágenes de entrada y salida deben ser almacenadas en el mismo directorio.
3. El programa en ensamblador debe generar un archivo de salida con la imagen interpolada.
4. El programa debe permitir que el usuario escoja el cuadrante (16 posibilidades) y mostrar el resultado de la interpolación.
5. La imagen mostrada debe ser escrita en un **.img** que será leído e interpretado por un software de alto nivel libre.
6. El formato de las imágenes será en escala de grises con píxeles con valores entre $[0, 255]$.
7. La imágenes deben ser visualizadas en algún software de alto nivel (e.g., Python, Matlab, Octave, etc), **solo** la visualización del archivo se realizará aquí. El **procesamiento** de la imagen es **completamente en ensamblador (asm)**. En la defensa se comparará la similitud del archivo de salida y el procesado en el programa de alto nivel de visualización.
8. En la interfaz del programa de alto nivel se deben visualizar por lo menos tres cosas: imagen de entrada, cuadrante seleccionado (puede ser por texto o seleccionado en un recuadro rojo por el programa en alto nivel) y resultado de la interpolación.

-
9. El simulador del ISA (solo puede usar uno de los siguientes: ARM, x86, RISC-V, no necesariamente tiene que usar las herramientas de la **Tarea 1**) y otras herramientas de desarrollo podrán ser elegidas abiertamente por cada estudiante. Todo el procesamiento de la imagen debe ser en este simulador seleccionado, recuerde que el lenguaje de alto nivel es solo para visualización, **si usa lenguaje de alto nivel en procesamiento obtendrá nota de cero.**
 10. Por facilidad, todo este programa debe ser integrado en un solo *framework* automatizado que tome el archivo de texto, lo procese en ensamblador, genere una imagen salida y la lea el programa de alto nivel.

El proceso de diseño debe incluir propuestas y comparación de viabilidad de las mismas, estas deben ser no fácilmente descartables.

3. Metodología de trabajo

El proyecto debe seguir los siguientes aspectos de desarrollo, sino, la parte funcional no será calificada y obtendrá nota de cero:

1. Utilice una cuenta de repositorio gratuita.
2. Cree un repositorio con el siguiente nombre: `<user_id>.computer_architecture_1_2025`. El `user_id` estará compuesto por la primera letra del nombre y el apellido. Por ejemplo, para el estudiante Luis Chavarría, el nombre del repositorio será:
`lchavarria_computer_architecture_1_2022`.
3. Si el repositorio es privado, proporcione acceso a `luchazam` (bitbucket) o `luchazam` (GitHub).
4. El repositorio de Git contendrá dos ramas principales: `master` y `development`.
5. Inicialmente, la rama de `development` se crea a partir del `master`.
6. Al trabajar en un proyecto, el estudiante debe crear una nueva rama de trabajo desde `develop` y cuando la función esté lista, la rama debe fusionarse para `develop`. Cualquier corrección o modificación adicional después de `merge` debería requerir que se repita el proceso (es decir, crear la rama desde `develop` y fusionar los cambios más tarde). Una vez que el código de desarrollo esté listo, se fusionará con `master` y se debe crear una `tag`. El proceso se describe en la siguiente Figura 3.

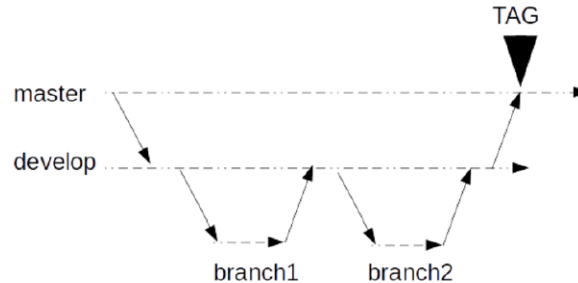


Figura 3: *Git workflow*

Adicionalmente se coloca este [enlace recomendado](#).

7. Después de haber realizado algunos proyectos la rama `master` debe verse así:

- `master/`
 - `proyecto_1`
 - `proyecto_2`
 - ...
- ...

Donde cada directorio de `proyecto_x` contiene todos los entregables para cada proyecto.

No es permitido realizar todo el trabajo en un solo commit, es decir, que realice el trabajo de forma local y solo suba el último entregable en el repositorio. Si no, obtendrá nota de cero. Debe mostrar avance incremental (se revisarán estadísticas).

4. Evaluación y entregables

La evaluación del proyecto se da bajos los siguientes rubros:

- Presentación proyecto 100 % funcional (75 %): Una defensa de 20 minutos donde el profesor evaluará el rendimiento del algoritmo. El/la estudiante llevará a la defensa una imagen libre que cumpla los requerimientos. **Debe notificar al profesor del repositorio a más tardar el lunes después de realizar la entrega del enunciado a las 11:59 pm.**
- Documentación de diseño (25 %): Este documento se encuentra directamente ligado con el atributo AP. La documentación del diseño deberá contener las siguientes secciones:

1. Listado de requerimientos del sistema: Cada estudiante deberá determinar los requerimientos de ingeniería del problema planteado, considerando partes involucradas, estado del arte, estándares, normas, entre otros.
 2. Elaboración de opciones de solución al problema: Para el problema planteado deberán documentarse al menos dos opciones de solución. Cada solución deberá ser acompañada de algún tipo de diagrama. **Estas opciones de solución no deben ser fácilmente descartables y deben llevar un análisis objetivo con base en criterios técnicos o teóricos.**
 3. Comparación de opciones de solución: Se deberán comparar explícitamente las opciones de solución, de acuerdo con los requerimientos y otros aspectos aplicables de salud, seguridad, ambientales, económicos, culturales, sociales y de estándares. **La comparación debe ser objetiva basado en las características técnicas y teóricas del ISA.**
 4. Selección de la propuesta final: Se deberá evaluar de forma objetiva, válida y precisa las soluciones planteadas al problema y escoger una solución final.
 5. Archivo tipo README donde especifiquen las herramientas que usaron y las instrucciones de uso. **Es un documento README.MD aparte.**
- Los documentos serán sometidos a control de plagios para eliminar cualquier intento de plagio con trabajos de semestres anteriores, actual o copias textuales, tendrán nota de cero los datos detectados. Se prohíbe el uso de referencias hacia sitios no confiables.
 - No coloque código fuente en los documentos, quita espacio y aporta poco. Mejor explique el código, páselo a pseudocódigo o use un diagrama.