

Práctica #1

Operaciones

puntuales

1ra. Parte



3



22 OCTUBRE

Análisis de Imágenes
María Elena Cruz Meza



Práctica #1 Analisis de Imágenes

Objetivo: Mostrar la habilidad para identificar los parámetros y el manejo e implementación de los algoritmos de operaciones punto a punto a imágenes binarias, en grises y en color.

Tareas en esta primera parte (objetivos específicos):

- ✓ Abrir y cerrar una imagen (operación identidad)
- ✓ Transformación de una imagen en niveles de gris (elegir algún método)
- ✓ Operación negativo o inverso a una imagen (binaria, en grises y en color)

Instrucciones

- **Conocimientos previos**
 - a) Habilidades en la implementación de vectores y matrices, así como graficar histogramas.
- **Sistema operativo y lenguaje de programación**
 - a) Elegir el sistema operativo que vaya a utilizar para desarrollar las prácticas, Windows o algún distro de Linux.
 - b) Elegir el lenguaje de programación en el que tengas mayor conocimiento o manejo y que sea sencilla la inclusión y manipulación de librerías para manipular histogramas.
- **Entrega de evidencia**
 - a) Medio para su entrega: Entregar la tarea en la plataforma Microsoft Teams una vez terminada toda la práctica (dos algoritmos). Elaborar un informe del desarrollo de esta práctica en un archivo con formato de video (consultar rúbrica).
 - b) Crear un archivo en formato de video, elija la herramienta de su preferencia y subirlo a alguna plataforma para compartir el link para visualizarlo (drive, Dropbox, youtube, etc). El archivo debe enviarse con una nomenclatura específica y con el siguiente contenido:
 - Nombre al archivo como: Si se elige una plataforma donde se tiene acceso al archivo, este deberá nombrarlo como: P1-ABC-.FOR, donde “P1” indica el número de práctica reportada, ABC corresponderá a iniciales de tu nombre (empezando por el apellido), y “.FOR” es el formato correspondiente al archivo tipo video, etc. Si optas por compartirlo en youtube o alguna herramienta similar, entonces el nombre deberá ser: “Practica #1-ABC– Clasificador Bayesiano”, donde ABC corresponde a las iniciales de tu nombre. Como solo es para fines académicos se sugiere compartirlo en forma privada, pero, tú decides su publicación libre.
 - Portada: Es importante agregar algunos datos al inicio del vídeo: identidad politécnica (Nombre o logos de la unidad académica: IPN y Escom), Nombre de la unidad de aprendizaje, Número de la práctica, Objetivo de la práctica, Nombre de quien presenta, es opcional mostrar el Nombre del(a) facilitador(A) (profesora) y Fecha de entrega.
 - Marco teórico: El vídeo debe incluir una breve información formal que sustenta el trabajo que se desarrolla (para este caso puede tomar como base o referencia

lo que se anexa al final de este documento, notas del curso facilitadas en la plataforma o consultar algún autor), si consulta alguna página, esta debe ser validada por alguna universidad u organismo educativo (paper o artículos, cursos en línea, presentaciones de docentes o investigadores, libros gratuitos en línea o por alguna editorial, etc.).

- Programa y pruebas: En el vídeo debe explicarse que lenguaje de programación es utilizado para implementar el algoritmo, explicar los segmentos de código principales, y posteriormente el funcionamiento de este con al menos dos ejemplos
 - En caso de no terminar todos o bien crea que aun tenga fallas en la programación, presentar los que se hayan realizado. Al entregar el vídeo también anexa un archivo en formato PDF o TXT, con el código completo para su revisión y ponderación respectiva. El nombre de este archivo deberá nombrarse como el archivo de vídeo
- Créditos: Agregar las referencias consultadas para el desarrollo de la actividad.
- La duración del vídeo es recomendable entre 5 a 10 minutos.

Prácticas relacionadas con las Unidades I y II con los temas: Creación de imágenes digitales y su almacenamiento, y operaciones punto a punto

Fundamentos

Las operaciones orientadas al punto transforman a la imagen modificando un pixel a la vez, en general sin importar el estado de los pixeles vecinos o todos ellos en forma global (toda la escena). La transformación se puede aplicar a toda la imagen o a una región de ella.

Definición. Sea \mathbf{x} un pixel de una imagen I , es decir $\mathbf{x} \in I$. Supongamos que la función $\mathbf{y} = f(\mathbf{x})$ transforma a un pixel \mathbf{x} mediante la regla f , generando un nuevo valor para él, digamos \mathbf{y} . Entonces diremos que la nueva imagen I' , donde $\mathbf{y} \in I'$, es el producto de aplicar f sobre I . Simbólicamente diremos que,

$$I' = f(I). \quad (1)$$

El proceso de transformación en la mayor parte de los casos será tal que

$$\text{sí } \mathbf{x} = I[i, j] \in \mathbf{y} = I'[i, j], \text{ donde } \mathbf{y} = f(\mathbf{x}).$$

Para que la transformación f no ocasione problemas de representación, si el dominio de \mathbf{x} está en el intervalo $D = [0, L - 1]$, donde $L = 2^p$, donde p es la profundidad en bits de la imagen,

entonces se va a exigir que $\mathbf{y} \in \mathbf{D}'$, donde en general $\mathbf{D}' \in \mathbf{D}$. Lo cual implica que el mecanismo de representación de la imagen sobre elementos de la clase \mathbf{x} , seguirá siendo válido para la clase a la que pertenece \mathbf{y} . Esta condición permite que los métodos desarrollados para la visualización de la imagen \mathbf{I} se pueden utilizar para \mathbf{I}' .

El algoritmo básico de transformación bajo f para una región rectangular de \mathbf{I} definida por

$$R = [i1 \dots i2, j1 \dots j2],$$

es el siguiente:

```

for i = i1, i2 {
  for j = j1, j2
  {
    I'[i, j] = f ( I[i, j] )
  }
}

```

fig. 1. Algoritmo básico para transformar una región de una imagen \mathbf{I} bajo f .

Consideraciones

Supongamos el caso en que: $i1 = 0$, $i2 = M$ (donde $M = \text{Imagen.Ancho}-1$), $j1 = 0$, N (donde $N = \text{Imagen.Alto}-1$); el proceso modificará a toda la imagen. Notemos que al cambiar f la transformación será diferente. Es decir, si definimos la composición de transformaciones de la manera habitual, tendremos que:

$$f_1 \circ f_2 (\mathbf{I}) = f_1 (f_2 (\mathbf{I})). \quad (2)$$

En general debemos conocer que al aplicar dos transformaciones a una imagen en diferente orden, es recomendable que no se deba esperar que la imagen resultante sea la misma, es decir, la composición de transformaciones no es conmutativa, por lo que simbólicamente tendremos que:

$$f_1 \circ f_2 (\mathbf{I}) \neq f_2 \circ f_1 (\mathbf{I}). \quad (3)$$

Definiremos una *batería* o *serie* de transformaciones f_k mediante la composición de ellas. Muchas de las operaciones de mejora de la imagen, detección de bordes, etc., se definen como una batería. El sentido de ésta es similar a la composición de las funciones que generan cada transformación. Sean f_1, f_2, \dots, f_n las funciones que definen cada proceso sobre la imagen, entonces la transformación compuesta o batería será:

$$F (\mathbf{I}) = f_1 \circ f_2 \circ \dots \circ f_n (\mathbf{I}) = f_1 (f_2 (\dots f_{n-1} (f_n (\mathbf{I})) \dots)). \quad (2.3)$$

Gráficamente podemos representar el proceso de transformación múltiple mediante celdas, donde cada celda representa una transformación o *filtro*. La figura siguiente (fig. 2.2) ilustra la situación.



fig. 2. Representación gráfica de la composición de procesos.

Operaciones Elementales

La operación más simple es la *Identidad*, ésta deja a la imagen igual. Esta operación suele usarse para realizar por ejemplo copias de una imagen, notemos que la imagen copiada es la que podemos estar transformando y la imagen original la dejamos intacta, hacer esto es una buena práctica, recordemos que una vez transformada nuestra imagen, no hay transformación inversa que nos permita recuperar la imagen original. Veamos como es el proceso para invertir una imagen.

Sea I una imagen en colores, con un dominio $[0, L]$ para los valores del tono de cada canal para los pixeles, en la representación RGB estándar, de ancho M y alto N . La función correspondiente es: $y = x$, de donde $f(x) = x$. Si representamos esta función de manera gráfica visualizaremos una ecuación de mapeo lineal simple (fig. 3). Esta función nos indica que el tono w es mapeado al tono w' .

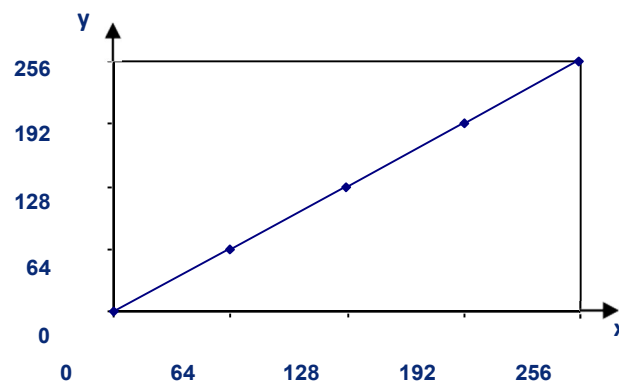


Fig. 3. Transformación identidad entre las variables: $y = f(x) = x$, ($p=8$, $L=255$)

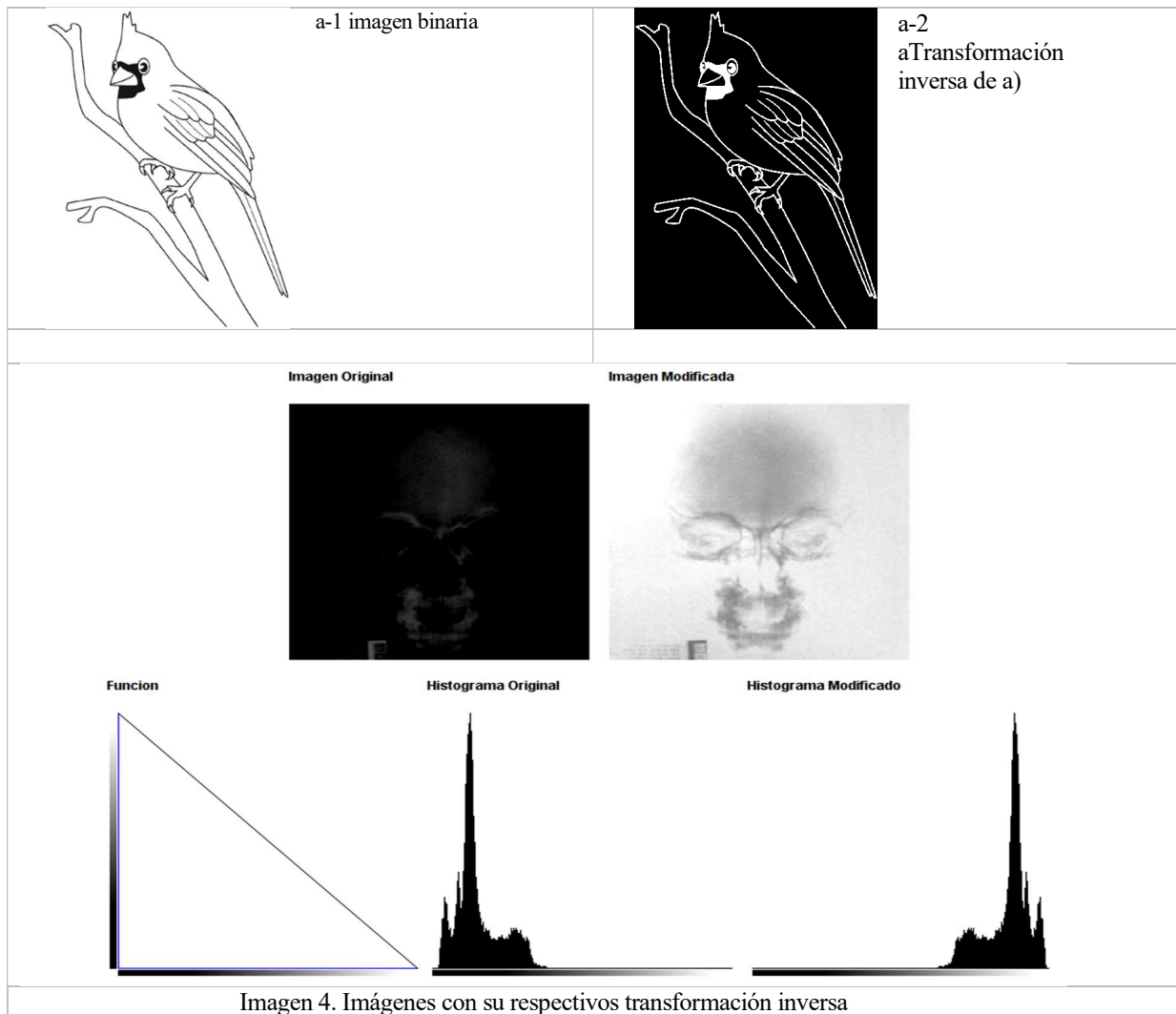
Negativo de una imagen

Una primera transformación lógica muy simple que le podemos aplicar a una imagen es el **negativo**, ésta se construye de alguna de las siguientes maneras, sea $x = (r, g, b)$ un pixel de la imagen I , entonces el negativo de x se puede hallar simplemente como:

$$x' = (\sim r, \sim g, \sim b) = (\Lambda - r, \Lambda - g, \Lambda - b), \quad (2.4)$$

donde $\Lambda = L - 1$ y $L = 2^p$.

Veamos un ejemplo de esta transformación a una imagen en tonos de gris ($r=g=b=z$), donde el negativo de un pixel $x = (z, z, z)$ será $x' = (\sim z, \sim z, \sim z)$, este efecto se muestra en la figura 4-a y 4-b. En la primera se muestra el proceso sobre dos imágenes de ejemplo, una monocroma o binaria y otra en tonos de gris. Por su parte, para obtener el negativo de una imagen en color, debes transformar cada canal RGB.



Conversión de una imagen en color a una imagen en niveles de gris

Una transformación que generalmente es muy útil aplicar a una imagen es la de transformarla de color en niveles de gris, usualmente el método más sencillo es utilizar el esquema de promedio ponderado, es decir, sea $\mathbf{x} = (r, g, b)$ un pixel de la imagen \mathbf{I} , entonces su nivel de intensidad o brillo (gris) en ese punto se puede hallar simplemente como:

$$X = \frac{R+G+B}{3}$$

Sin embargo hay otras maneras, las estandarizadas por la IEEE, debido a la respuesta del ojo al espectro visible. El ojo reconoce los patrones de iluminación en color en las siguientes proporciones para cada componente: Rojo:30%, Verde:59% y Azul:11%. Notemos que el ojo humano percibe distintas intensidades de luz en función del color que se observe, se dice que es más sensible al verde y al rojo que al azul.

Esto se puede modelar, y la expresión matemática del fenómeno es:

$$Y = R*0.3+G*0.59+B*0.11$$

Donde:

Los factores de ponderación de cada componente de color nos indican la sensibilidad del ojo humano a las frecuencias del espectro cercanas al rojo, verde y azul. Así, obtenemos el pixel de color gris con la iluminación adecuada para que nuestro ojo lo perciba como un mejor equivalente a su versión en color