
Práctica 3:

Operaciones entre imágenes

1. Objetivos

- Realizar operaciones aritméticas entre imágenes.
- Entender los procesos de saturación en los formatos usados para imágenes.
- Entender y crear una curva de precisión y cobertura.
- Entender las funciones de transformación de intensidad.
- Implementar transformaciones lineales definidas a trozos.

2. Parámetros de entrega

Sicua

- Adjuntar un único archivo (a menos que se especifique lo contrario) con los códigos de la siguiente forma: *main_CódigoEstudiante#1_CódigoEstudiante#2.m*.
- Adjuntar **TODO** el código que se entregó en el ítem anterior, en formato *.txt*. Llámelo de igual manera: *CódigoEstudiante#1_CódigoEstudiante#2.txt*. Esto con el fin de poder evaluar en Sicua automáticamente cualquier intento de copia o similitud entre los algoritmos. Cualquier intento de copia será tratado de acuerdo al reglamento de la Universidad. Aquel grupo que no incluya este ítem en su entrega tendrá una nota de 0 en el laboratorio.
- Adjuntar un único archivo con el informe en PDF con el mismo nombre del primer ítem: *CódigoEstudiante#1_CódigoEstudiante#2.pdf*. El informe debe presentarse tipo artículo con formato CVPR.
- En el caso de que no tenga compañero, **DEBE** utilizar un cero como el código de su compañero, e.i. *main_CódigoEstudiante_0.m* para el archivo de matlab, *CódigoEstudiante_0.pdf* para el informe y *CódigoEstudiante_0.txt* para el archivo de texto.
- **NO** se permiten archivos comprimidos tales como zip, rar, 7z, tar, gz, xz, iso, cbz, etc. Aquel grupo que envíe su informe como un archivo comprimido no tendrá calificación.
- No se recibirá ningún archivo por algún medio diferente a Sicuaplus.

3. Algunas reglas

- La asistencia a la sección de laboratorio inscrita es obligatoria. Acorde con el Reglamento General de Estudiantes de Pregrado de la Universidad de los Andes, la inasistencia a más del 20 % de las clases resultará en la reprobación de la materia.
- Cada práctica (Informe, código, etc.) tendrá plazo de entrega hasta las 11:59 pm del domingo de la semana de la sesión. El vínculo para el envío del laboratorio dejará de estar disponible luego de esta hora. Así mismo, se les entregará la guía de la práctica el sábado anterior a la sesión. Por tanto, dispondrán de una semana completa para la realización del mismo.
- Los informes deben realizarse **únicamente** con la pareja. Esto quiere decir que aunque es válido discutir los problemas con sus compañeros, la solución y el código, deben ser de su completa autoría. Está prohibido copiar literalmente el algoritmo y/o procedimiento desarrollado por otro grupo. Si llega a obtener un código de internet, asociado al problema a resolver, este debe estar debidamente referenciado y usted debe entenderlo por completo.

4. Parámetros de calificación

Resultados

- Todos los códigos deben mantener orden y coherencia en la ejecución de comandos, es decir, cada vez que se muestre una figura, el programa debe esperar para que se presione una tecla, para así continuar con la siguiente y así sucesivamente (para esto utilice *pause*). Tenga en cuenta que si se quieren contrastar dos imágenes use *subplot* o *imshowpair*.
- Toda figura debe llevar su título y descripción en el informe.
- El código debe estar debidamente comentado.
- NUNCA utilice rutas absolutas para leer o guardar archivos. Este es el error más común en la ejecución de los códigos.
- Para generar rutas utilice *fullfile* ya que puede que corra los laboratorios usando Linux o Windows y los separadores de archivos cambian dependiendo del sistema operativo.
- Asuma que dentro de la carpeta de ejecución del código se encuentran los archivos necesarios para el laboratorio.

Ejemplo: dentro del código principal el estudiante quiere leer la imagen *im.png*.

- Forma incorrecta: `imread('C:/Estudiante/MisDocumentos/ElLab/imagenes/im.png')`.
- Forma correcta: `imread(fullfile('imagenes','im.png'))`.

Informe

Todos los laboratorios deben realizarse en formato CVPR. Para aquellos que desean realizarlo en \LaTeX , pueden obtener una plantilla del formato en el siguiente link. Cabe resaltar que los informes del laboratorio no deben contener ninguna sección de artículo científico, esto significa que no deben incluir ninguna división como resultados, abstract o conclusiones. Por consiguiente, deben responder únicamente a las preguntas del informe. Recuerden incluir imágenes de sus resultados y documentarlas debidamente. Por último, el informe tiene una longitud máxima de 2 páginas. Se pueden incluir imágenes como anexos pero las imágenes principales deben ser parte del informe.

Bonos

Cada grupo ganará puntos que le suben la nota por cada una de las siguientes características que se cumpla:

1. Parametrizar el código, es decir, no incluir cantidades fijas en el código y en caso tal de ser inevitable definir las como parámetros de entrada.
2. Desarrollar el informe en \LaTeX . (Aquellas personas que lo desarrollen en \LaTeX , deben escribir al final del informe "**Realizado en \LaTeX** ". De lo contrario no se contará el bono. Los grupos que intenten reproducir la frase en un informe realizado en Word tendrán 0 en la nota de dicho laboratorio. Para poder escribir el logo utilice el comando " \LaTeX " en su informe de latex.

Estos puntos se asignarán de acuerdo al criterio de las profesoras.

5. Tipos de datos en imágenes

Por defecto Matlab almacena la mayoría de los datos en matrices de tipo *double*. Los datos en estas matrices son almacenados como números de punto flotante de doble precisión (64 bits). Todas las funciones de Matlab trabajan con estos arreglos. PERO, para procesamiento de imágenes esta representación no es la más ideal. El número de píxeles de una imagen puede ser muy grande, por ejemplo, una imagen de 1000x1000 tiene 1 millón de píxeles; si cada píxel fuera representado como un dato tipo *double* habría que tener 8MB para leer una sola imagen.

Por lo cual, el toolbox de imágenes de Matlab carga por defecto las imágenes como matrices de tipo *uint8* (8bits) y *uint16* (16bits) requiriendo así 1/8 o 1/4 de memoria comparado con arreglos tipo *double*.

Ejemplo: en el laboratorio pasado, la variable *vol* que contenía el MRI era de tamaño 240x240x155x4, si fuera de tipo *uint8*, se necesitarían 35MB de RAM para que pueda ejecutar sin problemas. Si fuera tipo *double* se necesitarían $\sim 3285\text{MB} \sim 0,28\text{GB}$ de RAM por volumen, eso sin tener en cuenta el espacio en RAM que Matlab requiere para funcionar; si se tiene una RAM total de 2GB, ya el computador va a comenzar a sentir el peso de Matlab.

Para realizar operaciones aritméticas *Pixel a Pixel* con imágenes, estas deben tener la misma clase (*uint8*, *uint16*, *single*, *double*, etc) y el mismo tamaño.

Operaciones aritméticas:

Operación	Sintaxis	Descripción
Suma	<i>imadd(a,b)</i> $a+b$	Suma dos imágenes o agrega una constante a una imagen.
Resta	<i>imsubtract(a,b)</i> $a-b$	Resta una imagen de otra, o una constante de una imagen.
Multiplicación	<i>immultiply(a,b)</i> $a.*b$	Multiplica dos imágenes, o una imagen por una constante.
División	<i>imdivide(a,b)</i> $a./b$	Divide una imagen entre otra, o una imagen por una constante.
Complemento	<i>imcomplement(a)</i> $\sim a$	Realiza el complemento de una imagen.
Combinación lineal	<i>imlincomb(k1,a,k2,b,...)</i> $k1 * a + k2 * b + \dots$	Combinación lineal de imágenes.

6. Procedimiento

Antes de comenzar, tenga en cuenta las reglas de saturación presentes al realizar operaciones entre imágenes: *Overview of Image Arithmetic Functions*.

6.1. Código tutorial

1. Realice la suma, resta y multiplicación de las imágenes *saturn.png* y *cameraman.tif*. Verifique el tamaño de ambas imágenes y la clase de estas para poder realizar las operaciones. De ser necesario cambiar el tamaño, use *imresize*.
2. Realice la resta entre *saturn.png* y *football.jpg* en ambos sentidos.
3. Realice una combinación lineal de los 3 canales (colores) de *onion.png*, usando *imapplymatrix*.

7. Informe

7.1. Operaciones entre imágenes (20 %)

1. Cargue la imagen *football.jpg* y modifíquela de forma tal en que obtenga la Fig. 1



Figura 1: Balón de football

2. ¿Cuál sería la forma de realizar una operación entre una imagen a color y una imagen en escala de grises? ¿Cuál sería la forma de realizar una operación entre estos dos tipos de imágenes, **sin** perder la información de los 3 colores?. (**Sólo escribir su respuesta en el informe, no es necesario realizar el código en Matlab de este punto**).
3. Sume las imágenes *trees.tif* y *eight.tif* de forma tal en que obtenga la Fig.2.

Tip: debe realizar operaciones adicionales a la adición.



Figura 2: Suma especial de Trees y Eight

4. Realice la multiplicación de *cameraman* y *rice*, de modo que el resultado sea observable.

Tip: tenga en cuenta la clase

5. Extraiga un círculo inscrito de cualquier imagen de entrada aplicando las operaciones aritméticas. Esto es, sin importar el tamaño de la imagen, extraer el círculo más grande posible dentro de ella. Vea la Figura 3. Tip: Use el radio del círculo y la distancia entre puntos

Tip2: tenga en cuenta hacer esto sólo con imágenes en escala de gris

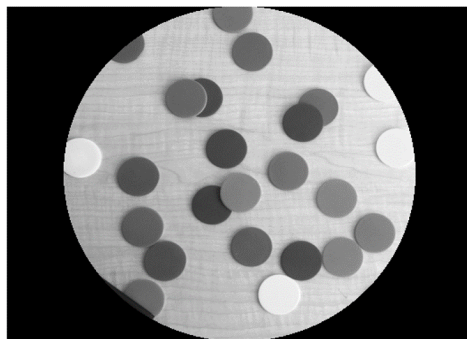


Figura 3

6. Dada dos imágenes binarias, ¿cómo podría representar las funciones AND y OR como funciones matemáticas? ¿Podría extender esas definiciones a imágenes en escalas de grises?

7.2. Precisión y Cobertura (40 %)

En este punto se va a trabajar con el resultado de un algoritmo de detección de peatones. Por cada imagen se cuenta con un bounding box de anotación, un bounding box detectado a partir del algoritmo y la confianza (score) con el cual se detectó el peatón.

- Descargue el archivo `detection_method.mat` del siguiente *link*) y póngalo en su espacio de trabajo. Este archivo contiene un arreglo de celdas donde cada celda corresponde a la información de una imagen. Dentro de la celda correspondiente a cada imagen se encuentra un bounding box de anotación, un bounding box detectado a partir del algoritmo y la confianza (score) respectivamente.
1. Construya la curva de precisión y cobertura sobre los datos proporcionados y analícelos.
 2. Diga por qué la curva muestra que a medida que aumenta la precisión disminuye la cobertura y viceversa.
 3. Calcule la F-medida
 4. Determine la precisión promedio (Average Precision, AP) de la curva.
 5. ¿Qué ocurre cuando se varía el umbral del solapamiento entre las anotaciones y las detecciones?
 6. Comente qué pasaría con la curva si en lugar de tener una detección por imagen, tuviera cien detecciones.
 7. ¿Por qué en la curva la cobertura nunca llega a 1?

7.3. Funciones de transformación de intensidad (40 %)

Realice una función llamada *Intensity_CódigoEstudiante#1_CódigoEstudiante#2.m* en donde realice una transformación lineal definida a trozos de una imagen de entrada. La función recibe como parámetro una imagen en escala de grises y 4 números, que son r_1 , r_2 , s_1 , s_2 , como se muestra en la figura 4 y retorna la imagen transformada.

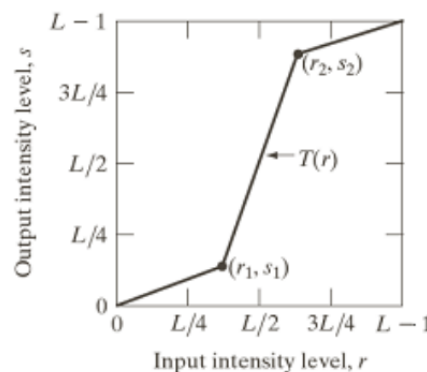


Figura 4: Función de transformación lineal definida a trozos.

Asimismo, la función debe mostrar en un subplot la imagen original y la transformada y en otra figura una gráfica representando la forma de la función de transformación $T(r)$ como se muestra en las figuras 5 y 6. **Ojo: Tenga en cuenta que su función también debe funcionar para umbralización.**

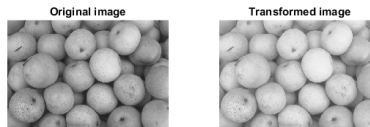


Figura 5: Imagen original vs imagen transformada.

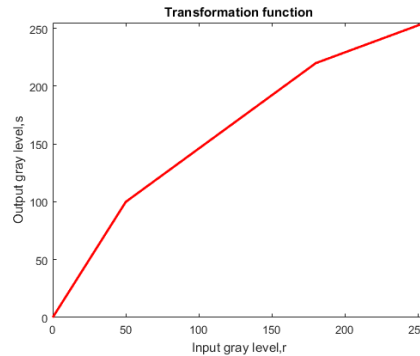


Figura 6: Forma de la función de transformación $T(r)$.

1. Analice las imágenes resultantes de su función para diferentes valores de parámetros.
2. ¿Cómo sería la forma de una función de transformación $T(r)$ que retorna una imagen más oscura que la original?
3. Use su función para umbralizar una imagen. ¿Cómo cambian los parámetros de la transformación?
4. Aplique su función a *'cell.tif'* para crear una imagen calibrada (con expansión del contraste). ¿Cómo modificó los parámetros de la transformación para lograr el resultado?

BONO

Usando *trees.tif* y *moon.tif* reproduzca la siguiente imagen.



Figura 7: Árboles dentro de la luna.