



PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE
ESCUELA DE INGENIERÍA
DEPARTAMENTO DE CIENCIA DE LA COMPUTACIÓN
SEGUNDO SEMESTRE DE 2014

IIC 1103 - Introducción a la Programación

Tarea 3

Objetivo General

Esta tarea tiene como objetivo que te enfrentes a un desafío de programación que te permita practicar los conocimientos adquiridos de Python, centrados principalmente en el tópico de *listas*. Para el logro de este objetivo deberás hacer una implementación en Python de un programa de procesamiento de imágenes *InstaIntro*.

Introducción

La fotografía digital usa un conjunto de sensores electrónicos para capturar la imagen enfocada por el lente de una cámara. Estos sensores perciben la intensidad de luz y convierten la imagen enfocada en un conjunto de *pixeles*, donde el color de cada *pixel* es la combinación de los tres colores básicos: rojo, verde y azul (RGB¹).

Una imagen es representada por una matriz (lista de listas) de pixeles, donde cada pixel es representado por una tupla de tres números enteros entre 0 y 255. El primer elemento de un pixel representa la intensidad de rojo, el segundo la intensidad verde y el tercero la intensidad de azul.

Las dimensiones de una imagen digital representan la cantidad de pixeles que esta contiene a lo ancho (x) y alto (y) (Figura 1(a)). En esta tarea una imagen es representada por una matriz (*lista de listas*) de pixeles de tamaño $y * x$, es decir la matriz tiene y filas y x columnas (Figura 1(b)).

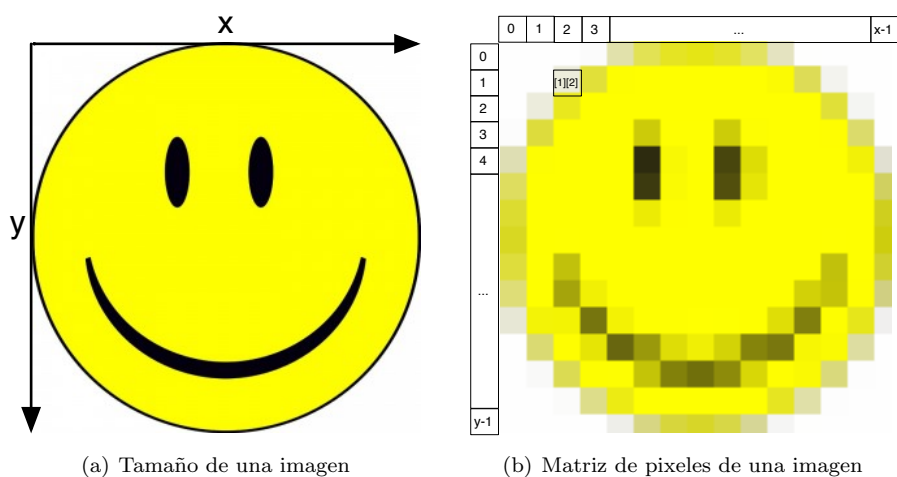


Figura 1: Imagen digital como un matriz de pixeles.

¹http://en.wikipedia.org/wiki/RGB_color_space

Operadores

Algunas de las operaciones que se pueden hacer en una imagen son:

Girar

Gira la imagen 90 grados en el sentido de las manecillas del reloj.

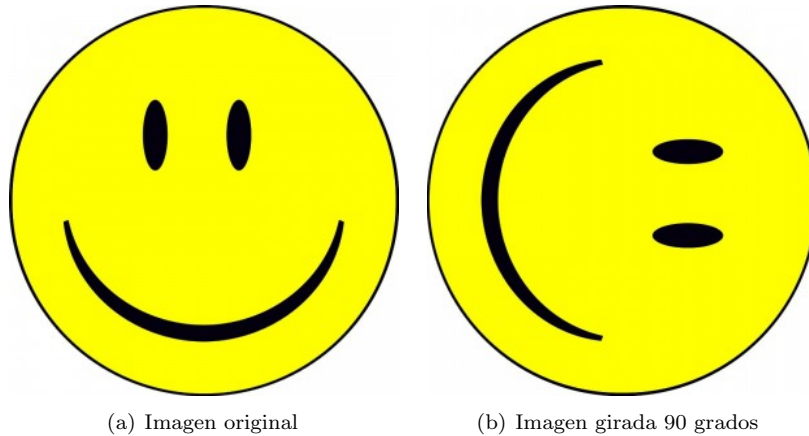


Figura 2: Girar la imagen.

Escala de Grises

Convierte una imagen de color a otra en escala de grises. Para que una imagen sea vea en tonos de gris se requiere que los tres componentes básicos del color (rojo, verde, azul) tengan más o menos la misma intensidad. Por lo tanto, podemos decir que si queremos convertir un pixel a su equivalente en escala de grises bastaría con cambiar la intensidad de cada color del pixel por el promedio de los tres colores básicos. Por ejemplo, el pixel $(255, 0, 0)$ será convertido al $(85, 85, 85)$.

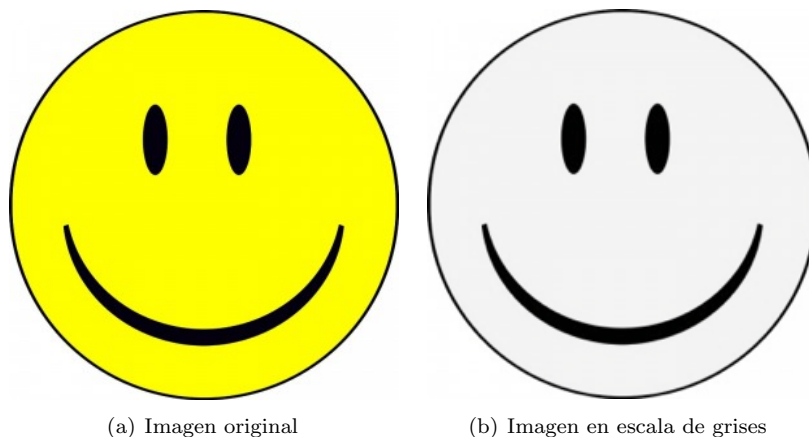


Figura 3: Convertir una imagen a otra en escala de grises.

Bordes

Una de las formas de obtener los bordes de una imagen es usando el operador *Sobel*. El resultado es una imagen transparente con líneas negras y algunos restos de color (Figura 4).

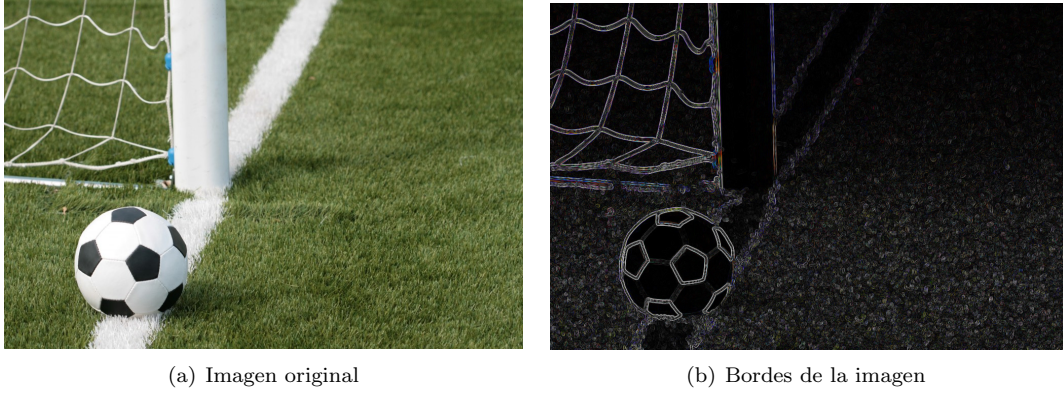


Figura 4: Obtener los bordes de una imagen.

Este operador usa dos matrices para detectar los bordes verticales y horizontales. Si definimos A como la imagen original, el resultado que se obtiene de aplicar cada operador a la imagen son dos imágenes G_x y G_y .

$$\mathbf{G}_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} * \mathbf{A} \quad \text{y} \quad \mathbf{G}_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * \mathbf{A}$$

El operador de *convolución* ($*$) entre una matriz de 3 x 3 como la que se muestra en la Figura 5 y una imagen A genera una nueva imagen A' definida como:

$$\mathbf{A}'[i][j] = a * \mathbf{A}[i-1][j-1] + b * \mathbf{A}[i-1][j] + c * \mathbf{A}[i-1][j+1] + d * \mathbf{A}[i][j-1] + e * \mathbf{A}[i][j] + f * \mathbf{A}[i][j+1] + g * \mathbf{A}[i+1][j-1] + h * \mathbf{A}[i+1][j] + i * \mathbf{A}[i+1][j+1]$$

Finalmente la imagen resultante \mathbf{G} es definida como $\mathbf{G}[i][j] = \sqrt{\mathbf{G}_x[i][j]^2 + \mathbf{G}_y[i][j]^2}$

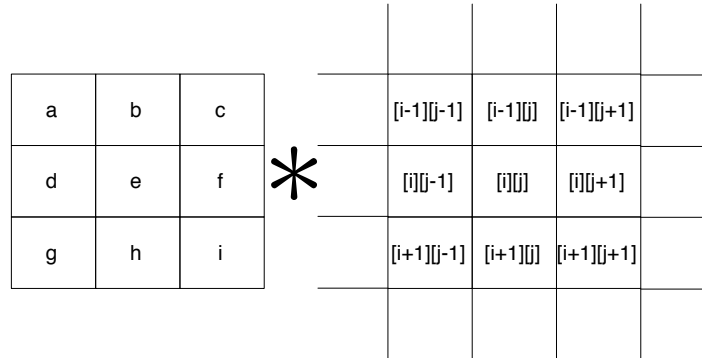


Figura 5: El operador de convolución.

Mosaico

Transforma una imagen reemplazando porciones de la imagen original por porciones similares de otras imágenes. Cada porción tiene la forma de un cuadrado y para encontrar la menor diferencia entre dos porciones es necesario comparar cada pixel de la imagen (Figura 6).

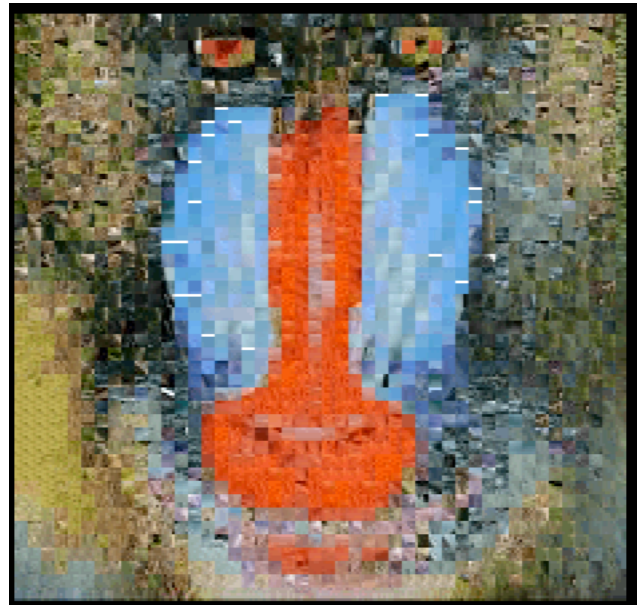
La diferencia entre dos pixeles se calcula como suma de los los cuadrados de la diferencia de colores. Es decir, la diferencia entre el pixel $p1 = (r_1, g_1, b_1)$ y $p2 = (r_2, g_2, b_2)$ es el resultado de $(r_1 - r_2)^2 + (g_1 - g_2)^2 + (b_1 - b_2)^2$. La diferencia entre dos porciones se calcula como la suma de la diferencia de cada uno de sus pixeles.

Ten en cuenta que una comparación exhaustiva entre una porción de la imagen original contra todas las otras posibles porciones de las otras imágenes puede tomar mucho tiempo. Por lo tanto, para encontrar la porción de

pixeles más similar a la porción original, deberás programar un algoritmo que evite una comparación exhaustiva usando técnicas de ordenación y búsqueda.



(a) Imagen original



(b) Imagen como un mosaico

Figura 6: Transformar una imagen a un mosaico de porciones de imágenes.

Problema

En esta tarea deberás escribir el programa *InstaIntro*, que permite modificar imágenes (como una versión simplificada de Instagram). En particular tu programa deberá permitir al usuario abrir y modificar una imagen usando los operadores escritos en la sección de Operadores.

Librerías

Para esta tarea cuentas con la librería:

`instaintro_gui`

Esta librería te permitirá manejar el programa a través de una interfaz gráfica. Al iniciar tu programa se abrirá una interfaz gráfica que tiene los elementos necesarios para que funcione el sistema de edición de imágenes (ver Figura 7).

Para que tu programa pueda interactuar con el usuario correctamente, la librería `instaintro_gui` ofrece las siguientes instrucciones:

- `obtener_pixeles()`: Retorna la matriz de pixeles de la imagen cargada usando el botón **Abrir**.
- `actualizar_imagen(pixeles)`: Actualiza la imagen usando la matriz de pixeles que recibe como parámetro.
- `esperar_click()`: Al llamar a esta instrucción el programa se detiene hasta que el usuario haga click en alguno de los botones de la interfaz. Luego retorna el nombre en minúsculas del botón que se presionó. Por ejemplo, si el usuario presionó el botón **Gris** entonces esta función retornará la cadena de texto `gris`.

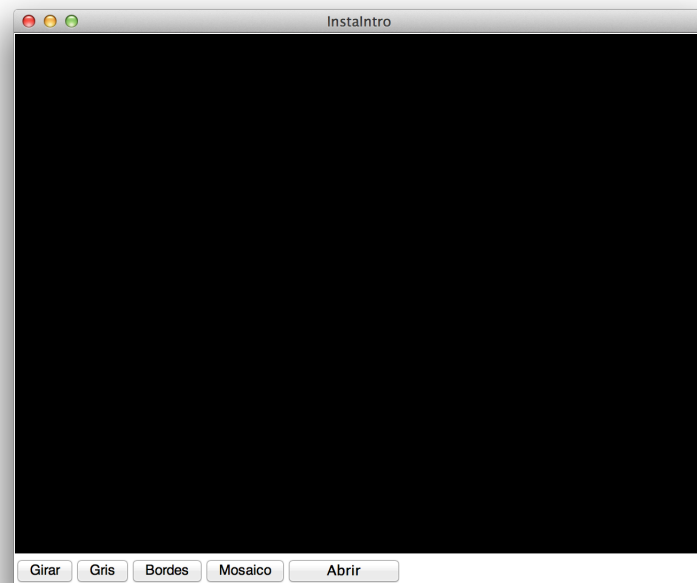


Figura 7: Ventana principal.

- `obtener_imagenes_mosaico()`: Al llamar a esta instrucción el programa procesará aquellas imágenes de tipo GIF (.gif) que estén contenidas en la carpeta `mosaico`. Luego de esto, la función retornará la lista de imágenes procesadas. Ten en cuenta que cada imagen es una matriz de pixeles.
- `salir()`: Cierra el programa.

Configuración

Para que tu proyecto funcione correctamente, debes tener en una misma carpeta tu proyecto junto con los archivos que se encuentran junto a este enunciado en la página del curso:

- `instaintro_gui.py`
- `mosaicos`

Además, el archivo de tu proyecto debe cumplir con la plantilla que se muestra en el siguiente fragmento de código y que también podrás encontrar en la página del curso.

```

1 import instaintro_gui
2
3 def tarea():
4     #Aquí empieza tu tarea
5
6
7 app = intro_gui.Application("tarea")
8 app.title('InstaIntro')
9 app.loadProgram(tarea)
10 app.start()
```

Guarda tu proyecto con el nombre `tarea3_nroalumno_seccion.n.py` donde debes reemplazar `nroalumno` por tu número de alumno y `n` por el número de tu sección. Por ejemplo, Juan Pérez de la sección 1 con número de alumno 12345, guardará su proyecto con el nombre `tarea3_12345_seccion.1.py`.

Indicaciones Generales

- Recuerda que la tarea es individual, y que las copias serán sancionadas con nota 1,1 final en la asignatura.
- La entrega de esta tarea es a través del SIDING en la sección cuestionario. Podrás subir tu tarea las veces que creas necesario mientras el plazo de entrega sea válido.
- La fecha última de entrega es el miércoles 22 de octubre hasta las 23:59 PM. No esperes hasta última hora, pues el buzón (cuestionario) se cierra automáticamente a esa hora. No se aceptarán entregas atrasadas ni entregadas por otros medios.
- Revisa bien lo que entregas, aunque esta vez podrás subir tu tarea ilimitadas veces, la versión final a corregir será la última que hayas subido.
- Solo podrás usar las librerías que estén instaladas por defecto en Python.
- En caso de dudas, recurre al foro del curso (`edx.ing.puc.cl`). Es probable que ya encuentres una respuesta, o en caso contrario ayudarás a resolver las dudas de tus compañeros.