


Facultad de Ciencias
UNAM

MANUAL DE USUARIO

MODELADO Y PROGRAMACIÓN

Integrantes:
Flores Arriola Rafael Edson - 423118018.
González Lucero Alan Uriel - 320148204.
Ortega Medina David - 319111866.
Rivera Soto Aline Daniela - 320333035.

LICENCIATURA
EN CIENCIAS DE
LA COMPUTACIÓN

ÍNDICE

1. Proyecto

1.1 Introducción

El presente manual tiene como finalidad dar a conocer de manera detallada y sencilla la estructura del programa presentado para que cualquier persona pueda sacar el máximo partido de la misma. El programa fue diseñado para que cada quien pueda, de una forma intuitiva, realizar la petición del índice de cobertura nubosa de una imágen, así como también tener la posibilidad de recibir dicha imágen en blanco y negro.

1.2 Objetivo

Brindar una descripción clara y detallada sobre el funcionamiento del programa para guiar al usuario sobre cómo usarla y guiar a desarrolladores en la búsqueda de información.

2. Lineamientos generales

2.1 Tratados internacionales

- Tratado sobre la promoción, protección y disfrute de los derechos humanos en internet.

2.2 Leyes

- Ley Federal de Protección de Datos en Posesión de los Particulares.
- Ley General de Responsabilidades Administrativas.
- Ley General de Transparencia y Acceso a la Información Pública.

2.3 Reglamentos

- Reglamento de la Ley Federal de Transparencia y Acceso a la Información.

2.4 Código de Conducta

Nuestro Compromiso

Nosotros, como miembros, colaboradores y líderes, nos comprometemos a hacer que la participación en nuestra comunidad sea una experiencia libre de acoso para todos, sin importar la edad, el tamaño del cuerpo, la discapacidad visible o invisible, la etnia, las características sexuales, la identidad y expresión de género, el nivel de experiencia, la educación, el estatus socioeconómico, la nacionalidad, la apariencia personal, la raza, la religión, la identidad sexual u orientación.

Nos comprometemos a actuar e interactuar de manera que contribuya a una comunidad abierta, acogedora, diversa, inclusiva y saludable.

Nuestros Estándares

Ejemplos de comportamientos que contribuyen a un ambiente positivo para nuestra comunidad incluyen:

* Demostrar empatía y amabilidad hacia otras personas.
* Ser respetuoso con opiniones, puntos de vista y experiencias diferentes.
* Dar y aceptar con gracia retroalimentación constructiva.
* Aceptar la responsabilidad y disculparse con aquellos afectados por nuestros errores, aprendiendo de la experiencia.
* Enfocarse en lo que es mejor no solo para nosotros como individuos, sino para la comunidad en general.

Ejemplos de comportamientos inaceptables incluyen:

* El uso de lenguaje o imágenes sexualizadas, y la atención o avances sexuales de cualquier tipo.
* Hacer trolling, comentarios insultantes o despectivos, y ataques personales o políticos.
* Acoso público o privado.
* Publicar información privada de otros, como una dirección física o de correo electrónico, sin su permiso explícito.
* Cualquier otra conducta que razonablemente pueda considerarse inapropiada en un entorno profesional.

Responsabilidades del Programa

Los líderes de la comunidad son responsables de aclarar y hacer cumplir nuestros estándares de comportamiento aceptable y tomarán las medidas correctivas apropiadas y justas en respuesta a cualquier comportamiento que consideren inapropiado, amenazante, ofensivo o dañino.

Los líderes de la comunidad tienen el derecho y la responsabilidad de eliminar, editar o rechazar comentarios, commits, código, ediciones de wiki, problemas y otras contribuciones que no estén alineadas con este Código de Conducta, y comunicarán las razones de sus decisiones de moderación cuando sea apropiado.

Alcance

Este Código de Conducta se aplica en todos los espacios de la comunidad, y también cuando una persona representa oficialmente a la comunidad en espacios públicos. Ejemplos de representar nuestra comunidad incluyen usar una dirección de correo electrónico oficial, publicar a través de una cuenta oficial en redes sociales, o actuar como representante designado en un evento en línea o presencial.

Aplicación

Los casos de comportamiento abusivo, acosador o inaceptable pueden ser reportados a los líderes de la comunidad responsables de la aplicación en @ciencias.unam.mx. Todas las quejas serán revisadas e investigadas de manera rápida y justa.

Todos los líderes de la comunidad están obligados a respetar la privacidad y seguridad de la persona que reporte cualquier incidente.

Directrices de Aplicación

Los líderes de la comunidad seguirán estas Directrices de Impacto Comunitario al determinar las consecuencias de cualquier acción que consideren una violación de este Código de Conducta:

1. ***Corrección***

Impacto Comunitario: Uso de lenguaje inapropiado u otro comportamiento considerado poco profesional o no bienvenido en la comunidad.

Consecuencia: Una advertencia escrita y privada de los líderes de la comunidad, aclarando la naturaleza de la violación y explicando por qué el comportamiento fue inapropiado. Se puede solicitar una disculpa pública.

2. Advertencia

Impacto Comunitario: Una violación por un solo incidente o una serie de acciones.

Consecuencia: Una advertencia con consecuencias por la conducta continua.

3. Generalidades

3.1 ¿Cómo está formado?

El proyecto está formado de dos partes, una implementada en Pascal y otra en Python, ambas versiones son funcionales.

Hicimos esto ya que muchas de las bibliotecas de Pascal solo están disponibles al usar la IDE de Lazarus, lo cual complicaba enormemente el desarrollo del proyecto, está IDE no logramos hacerla funcionar en Linux y en Windows nos daba muchos problemas.
Logramos implementarlo en Pascal, sin embargo no cumple las expectativas. La imagen con la trabaja tiene que ser en formato .bmp, un formato muy pesado, solo así logra hacerlo de la manera esperada, la ejecución nos tomó alrededor de 34 segundos.

En Python las cosas fueron mucho más sencillas, y logra cumplir con las expectativas del proyecto, maneja los formatos requeridos y lo hace de manera más rápida y sencilla

3.2 ¿A quién está dirigido?

El programa detallado en el presente manual está enfocado para el público con interés en conocer el porcentaje nuboso de cualquier imágen tomada del cielo en un formato circular, por ejemplo, meteorólogos.

3.3 Roles y responsabilidades

A continuación expondremos los roles que cada integrante del equipo tuvo en la elaboración del programa, así como sus responsabilidades.

- ***Entrada***
  El encargado de esta parte del proyecto fue responsable de:
  - Definir los parámetros de entrada.
  - Asegurar la correcta lectura de archivos de imagen.

La responsable de esta área fue Rivera Soto Aline Daniela.

- ***Algoritmo***
  El encargado de esta parte del proyecto fue responsable de:
  - Desarrollar el algoritmo para calcular el índice de cobertura nubosa.
  - Implementar las funciones de procesamiento de imágenes.
  El responsable de esta área fue Ortega Medina David.

- ***Pruebas Unitarias***
  El encargado de esta parte del proyecto fue responsable de:
  - Verificar el correcto funcionamiento del programa.
  - Documentar y hacer el readme.
  El responsable de esta área fue González Lucero Alan Uriel.

- ***Manejo de Errores***
  El encargado de esta parte del proyecto fue responsable de:
  - Implementar manejo de errores.
  -
  El responsable de esta área fue Flores Arriola Edson Rafael.

Por otro lado el líder del proyecto fue David Ortega Medina.

4. Información de Software

4.1 Software

El Software requerido para ejecutar el programa es tener Pascal y Python instalados en tu sistema operativo,

4.2 Requerimientos

Para poder ejecutar de manera exitosa el programa será necesario que el usuario tenga instalados los siguientes requerimientos:

- Python 3.
- pip (administrador de paquetes para Python).
- Free Pascal Compiler.
- Pillow (biblioteca de Python).

4.3 ¿En qué está enfocado?

El software está enfocado en analizar imágenes del cielo para calcular el índice de cobertura nubosa y proporcionar una imagen segmentada que resalta las áreas cubiertas por nubes.

4.4 Diagrama de flujo

Se encuentra en el archivo anexado llamado: Diagrama de Flujo Proyecto 2.pdf.

4.5 Pseudocódigo

cci.py

Inicio:

Verificar argumentos entrada:
Si no hay suficientes argumentos:
Mostrar mensaje de uso y terminar.
Detectar comando de Python:
Se intenta encontrar ‘python3’, ‘python’ o ‘py’.
Si no se encuentra un comando adecuado:
Mostrar error y terminar.
Verificar existencia del archivo imagen:
Si el archivo de imagen no existe:
Mostrar error y terminar.
Compilar y ejecutar ‘recortar.pas’
Si falla la compilación o la ejecución:
Mostrar error y terminar.
Verificar existencia de ‘salida-limpio.bmp:
Si ‘salida-limpio.bmp’ no existe:
Si ‘salida-limpio.bmp’ no existe:
Mostrar error y terminar.
Compilar y ejecutar ‘algoritmoIn.pas:
Si falla la compilación o ejecución:
Mostrar error y terminar.
Ejecutar ‘algoritmoIN.py’:
Si falla la ejecución:
Mostrar error y terminar.
Ejecutar ‘recortar.py’:
Si falla ejecución:
Mostrar error y terminar.
FIN

recortar.pas

```
Programa RecortarImagenCircular

Constantes:
        NOMBRE_ARCHIVO_SALIDA = "salida-limpio.bmp"
        ANCHO_IMAGEN = 4368
        ALTURA_IMAGEN = 2912
        CENTRO_CIRCULO_X = 2184
        CENTRO_CIRCULO_Y = 1456
        RADIO_CIRCULO = 1324
        DIAMETRO_CIRCULO = RADIO_CIRCULO * 2

    Procedimiento RecortarCirculoBMP(nombreArchivoEntrada, nombreArchivoSalida)
        Crear imagenJPEG como imagen
        Crear imagenBMP como imagen
        Crear lectorJPEG y escritorBMP para archivos

        Cargar imagenJPEG desde nombreArchivoEntrada

        Configurar imagenBMP para no usar paleta de colores
        Definir colorAmarillo como (rojo = máximo, verde = máximo, azul = 0)

        Para cada píxel (x, y) en imagenBMP
        Calcular distancia del píxel al centro del círculo
        Si distancia es menor o igual a RADIO_CIRCULO
        Copiar color del píxel correspondiente en imagenJPEG a imagenBMP
        Si no
        Colorear píxel en imagenBMP con colorAmarillo
        FinSi
        FinPara

        Guardar imagenBMP en nombreArchivoSalida

        Liberar memoria de lector, escritor, imagenJPEG, e imagenBMP

    FinProcedimiento

    Inicio del programa
        Si no hay archivo de entrada
        Mostrar mensaje de uso y salir
        FinSi
            Llamar a RecortarCirculoBMP con archivo de entrada y
NOMBRE_ARCHIVO_SALIDA
FinPrograma
```

algortimoIN.pas

Programa ContarPixelesNubes

Tipos:
TRGBPixel: Registro que representa un píxel con componentes R, G y B (rojo, verde, azul)

Variables:
BMPFile, OutFile: Archivo de entrada y salida BMP
Pixel, OutputPixel: TRGBPixel (píxel de entrada y salida)
TotalPixeles, PixelesNubes: Contadores de píxeles totales y de nubes
Width, Height, x, y: Dimensiones y coordenadas de la imagen
DataOffset: Offset donde comienzan los datos de la imagen
Header: Encabezado del archivo BMP
Filename, OutFilename: Nombre de archivos de entrada y salida
Flag: Bandera de opciones
GenerarSegmentacion: Booleano que indica si se debe generar la imagen segmentada

Función EsAmarillo(Color: TRGBPixel): Boolean
Retorna verdadero si el píxel es amarillo (R > 200, G > 200, B < 100)

Función EsBlanco(Color: TRGBPixel): Boolean
Retorna verdadero si el píxel es blanco (R > 200, G > 200, B > 200)

Procedimiento CargarDimensiones
Leer el tipo de archivo en BMPFile para verificar si es BMP
Si no es BMP, mostrar mensaje de error y salir
Leer el offset de datos de la imagen en BMPFile
Leer ancho y alto de la imagen en BMPFile

Procedimiento ProcesarImagen
Mover el cursor de BMPFile al inicio de los datos de imagen (DataOffset)
Si GenerarSegmentacion está activado, mover cursor de OutFile al mismo lugar
Para cada píxel (x, y) en la imagen
Leer el píxel en BMPFile
Si el píxel no es amarillo
Incrementar TotalPixeles
Si el píxel es blanco
Incrementar PixelesNubes
Asignar color blanco a OutputPixel
Si no
Asignar color negro a OutputPixel
Si el píxel es amarillo
Asignar el color original del píxel a OutputPixel
Si GenerarSegmentacion está activado

Escribir OutputPixel en OutFile
FinPara

Si el ancho de la imagen requiere relleno de bytes para alineación
Mover el cursor de BMPFile y OutFile según corresponda

Inicio del programa
Si no se proporciona el nombre del archivo BMP, mostrar mensaje de uso y salir
Leer el nombre del archivo BMP de entrada (Filename) y la bandera (Flag) si se proporciona
Establecer GenerarSegmentacion si Flag indica 'S' o 's'
Definir OutFilename basado en Filename

Abrir BMPFile
Si GenerarSegmentacion está activado
Crear OutFile y copiar el encabezado de BMPFile a OutFile

Llamar a CargarDimensiones para obtener dimensiones de la imagen

Inicializar TotalPixeles y PixelesNubes a 0

Llamar a ProcesarImagen

Cerrar BMPFile y OutFile si GenerarSegmentacion está activado

Si TotalPixeles es mayor que 0
Mostrar el porcentaje de nubes calculado como (PixelesNubes / TotalPixeles) * 100
Si no
Mostrar mensaje "No hay píxeles para analizar"

FinPrograma

recortar.py

Programa CirculoATransparente

Función CirculoATransparente(ruta_entrada)
    Argumentos:
    - ruta_entrada: Ruta del archivo de imagen a procesar
    Retorno:
    - Ninguno

    Abrir imagen en ruta_entrada y convertir a modo RGBA (para permitir transparencia)

    Definir dimensiones de la imagen (ancho, altura)
    Definir centro del círculo (centro_x, centro_y) y radio del círculo

    Crear una máscara en escala de grises con el mismo tamaño que la imagen (todo en negro)
    Crear un objeto para dibujar en la máscara

    Dibujar un círculo blanco en la máscara con centro en (centro_x, centro_y) y radio del círculo

    Aplicar la máscara de transparencia a la imagen (lo que esté fuera del círculo será transparente)

    Guardar la imagen procesada en un archivo llamado "salida-limpio.png"
    Imprimir mensaje confirmando que la imagen se ha guardado

FinFunción

Inicio del programa
    Si no se proporciona archivo de entrada
    Mostrar mensaje de uso y terminar programa
    FinSi

    Obtener ruta del archivo de entrada desde los argumentos de la línea de comandos
    Llamar a CirculoATransparente con ruta_entrada

FinPrograma

algortimoIN.py

Programa CalcularIndiceCoberturaNubosa

Función CalcularIndiceCoberturaNubosa(input_path, generar_imagen)

Argumentos:
- input_path: Ruta del archivo de la imagen de entrada
- generar_imagen: Booleano, indica si se debe generar una imagen segmentada

Retorno:
- Índice de cobertura nubosa (float)
- Ruta del archivo de la imagen segmentada si se genera, None en caso contrario

Abrir la imagen en input_path y convertir a formato RGBA
Obtener la matriz de datos de la imagen en formato numpy

Obtener dimensiones de la imagen (width, height)
Definir centro de la imagen (center_x, center_y) y radio del círculo (radius)

Crear una máscara circular:
Para cada píxel en la imagen:
Si el píxel está dentro del radio, incluirlo en la máscara circular
FinPara

Obtener los píxeles dentro de la máscara circular (circular_pixels)

Calcular brillo y saturación de cada píxel en circular_pixels:
- Brillo: Media de los valores RGB del píxel
- Saturación: Diferencia entre el valor máximo y mínimo de RGB del píxel
- Determinar si es un píxel de nube según el brillo y la saturación

Calcular:
- N = número de píxeles de nube
- C = número total de píxeles en la máscara circular
- Índice de cobertura nubosa (cloud_coverage_index) = (N / C) * 100

Si generar_imagen es Verdadero:
Crear una matriz de segmentación del tamaño de la imagen
Para cada píxel en la máscara circular:
Si es un píxel de nube, asignar color blanco (255, 255, 255, 255)
Si no, asignar color negro (0, 0, 0, 255)
FinPara
Guardar la imagen segmentada en output_path

Retornar cloud_coverage_index e output_path

FinFunción

```
Inicio del programa
    Si no se proporciona archivo de entrada:
    Mostrar mensaje de uso y terminar el programa

    Obtener input_path desde los argumentos de la línea de comandos
    Definir generar_imagen como Verdadero si el segundo argumento es "S"

    Llamar a CalcularIndiceCoberturaNubosa con input_path y generar_imagen
    Mostrar el índice de cobertura nubosa

    Si output_path existe:
    Mostrar la ruta donde se guardó la imagen segmentada

FinPrograma
```

4.6 ¿Cómo funciona?

El programa cci.py coordina la ejecución de varios scripts de Pascal y Python para procesar imágenes y calcular el índice de cobertura nubosa.

El funcionamiento se puede complementar con el diagrama de flujo y con el pseudocódigo.

Si se desea saber la ejecución, leer el readme.txt para más información.

Los archivos recortar.pas y recortar.py, básicamente cortan la imagen borrando los bordes de esta, para que algoritmoIN.pas y algoritmoIN.py calculen el índice de cobertura nuboso y te den la imagen segmentada.

5. Procedimientos

5.1 Mantenimiento

Se debe dar mantenimiento para que siga funcionando en futuras versiones de Python y Pascal.

Para verificar que siga funcionando correctamente con el paso del tiempo.

5.2 Actualizaciones

La primera actualización, puede ser que cci.py compile y ejecute el comando si se le pasa una bandera. Puesto que se le puede pasar una bandera, pero la ignorará. Se intentó agregarle la bandera, si se detectaba la bandera imprimía el mensaje del comando que se debe de usar para ejecutar con bandera, pero esa implementación, arruinaba el resto, ya no se podía ejecutar. Estaría bien trabajar en eso.

Futuras actualizaciones se podría encontrar una forma de mejorar el código, tanto estructural como de optimalidad.

Se pueden añadir más funcionalidades dependiendo de las necesidades que se presenten.

Trabajar para que incluso sea compatible con diferentes dispositivos.

6. Referencias

Para los programas en Pascal usamos principalmente estos enlaces):
[1] FreePascal, "sysutils". Sin fecha [Online], disponible en https://wiki.freepascal.org/sysutils

[2] Lazarus, "Corp a jpg file and save it". Marzo 2002 [Online]. Disponible en: https://forum.lazarus.freepascal.org/index.php?topic=48795.0

[3] galfar, "PasJpeg2000". 2010 [online]. DIsponible en: https://github.com/galfar/PasJpeg2000

[4] ukayaj620, "dip-tools". 2020 [online]. Disponible en: https://github.com/ukayaj620/dip-tools

[5] jmpessoa, "tfpnoguigraphicsbridge" 2015 [online]. Disponible en: https://github.com/jmpessoa/tfpnoguigraphicsbridge

Para los programas en Python:
[1] Pillow. "ImageBraw Module", sin fecha [online]. DIsponible en: https://pillow.readthedocs.io/en/stable/reference/ImageDraw.html

[2] Pillow. "Image Module", sin fecha [online]. Disponible en: https://pillow.readthedocs.io/en/stable/reference/Image.html

[3] Numpy. "Documentation", sin fecha [Online]. Disponible en: https://numpy.org/doc/

[4] José Galaviz Casas, "Cloud Coverage", 2009. [Online]. Disponible en: https://sites.google.com/ciencias.unam.mx/jgc-modelado-y-programacion/proyectos?authuser=0

[5] Roy, G., S. Hayman y W. Julian, "Sky analysis from CCD images: cloud cover", *Lighting Research Technology*, Vol. 33, No. 4, pp. 211-222, 2001.