Panorama

Un Programa orientado a objetos que intenta ser ilimitado, extensible, flexible y fácil de usar.

DESCRIPCIÓN

Desafío No. 1 (nivel bajo):

Panorama logra filtros, detección de bordes y otras operaciones matriciales sobre imágenes.

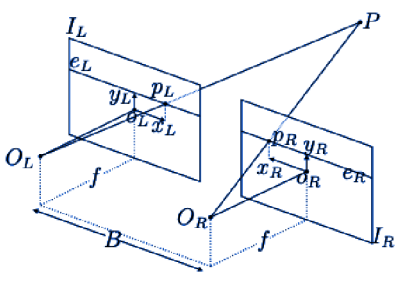
- Aritméticos
- Lógicos
- flipped and flopped
- mirror
- colorice (de blanco y negro a color)
- gray color (de color a blanco y negro)
- Threshold
- Sovel
- Morfológicos, Erosión, dilatación
- Estadísticos (Histograma)

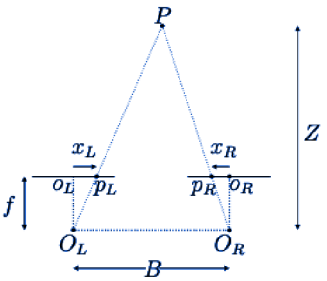
Desafío No. 2 (nivel medio):

Panorama logra el entramado verde (incrustar una imagen en el primer plano de otra). Por ejemplo, considere una foto de la ardilla de Univalle, que merece estar en un lugar agradable/emblemático o distintivo como la plazoleta. Entonces, cada píxel verde en la imagen de la ardilla será reemplazado por el píxel correspondiente en la imagen de la plazoleta y como resultado la ardilla aparece en el primer plano de la imagen resultante.

Desafío No. 3 (nivel máximo):

Panorama realiza cálculos para determinar la disparidad binocular. La estimación de la profundidad de los objetos en una escena a partir de un par estéreo de imágenes es fuente para muchas aplicaciones en el mundo de computación. Los métodos para la estimación de la profundidad mediante el cálculo de disparidad han variado mucho en las décadas que el problema está planteado. Las imágenes estéreo se obtienen desde dos puntos focales (Observadores) situados a una distancia B entre ellos. Ambos observadores capturan imágenes de un punto P que se encuentra a una





distancia Z. Las imágenes que se obtienen son similares, pero no iguales. En el esquema, se describe la captura de imágenes estéreo, los observadores (OL y OR) están separados por una distancia (B) y captura lo que ven en el punto (P). En este proyecto logra una caracterización de la profundidad utilizando cálculo de disparidad en imágenes estéreo.

Una imagen puede ser definida como una matriz de valores numéricos y de tamaño m x n. Cada valor en la posición (x, y) de la matriz es un píxel con un nivel de intensidad entre 0 y 255 (dependiendo de la escala puede ser 0 a otro valor.) Panorama utiliza un formato de archivo terriblemente ineficiente llamado "PPM simple". Algunas de estas imágenes podrán ser descargadas desde el campus virtual. La características más importante y la razón para usar este tipo de imagenes con extension (*.ppm o *.pgm) Si abre una de estas imágenes en un editor de texto (sublime, notepad, pycharm, wordpad o cualquier otro), verá algo parecido a esto:

P3
1280 960
255
0 255 0 125 255 5 0 255 0 . . .

El P3 es el "número mágico" o encabezado de archivo, que le dice a la computadora cuál es el formato, 1280 es el ancho de la imagen, en píxeles, 960 es la altura de la imagen en píxeles. 255 es el valor de intensidad, es decir, el mayor valor que tiene un pixel en la matrix (imagen).

El resto describe los valores rojo, verde y azul de cada píxel. Los valores que se encuentran en cada canal, estos números son 0 255 0, lo que significa que 0 para el canal rojo, 255 para el verde y 0 para el azul. El segundo píxel es (125, 255, 5) y así sucesivamente, hasta completar las tres matrices de tamaño 1280x960, una por cada canal. Los archivos con extensión (*.ppm o *.pgm) pueden ser visualizados como imágenes.

Tips para el éxito:

1) ¡Comienza temprano! Debido a que la programación es un progreso tan creativo, es notoriamente difícil predecir cuánto tiempo le llevará completar CUALQUIER proyecto de programación. Si comienza temprano, lo peor que puede suceder es que termine temprano y se relaje.
2) Trabaja deliberadamente, deteniéndose con frecuencia para compilar y probar los avances. Ésta es la mejor forma de detectar problemas de forma rápida y sencilla.
3) Empieza en pequeño. No vayas por el gran ejemplo de inmediato. En su lugar, use un ejemplo de depuración más pequeño que pueda seguir por completo, para que pueda verificar su trabajo.
4) Vuelva a revisar y/o compilar con frecuencia y definitivamente cuando termine cada parte.
5) Escribir buena documentación, nombres de variables, etc., a medida que avanza, siempre es más fácil que intentar agregarlos al final.

CALIFICACIÓN

El 30% de su calificación se basará en la funcionalidad (su programa funciona según lo especificado), principalmente en función de los desafíos y los casos de prueba que se han establecido, además de los casos de prueba adicionales de acuerdo con las especificaciones del proyecto. El código que envíe debe funcionar si espera obtener una calificación aprobatoria.

El 70% de su calificación se basará en el estilo de codificación, que incluye:

- 10% **Legibilidad**, se debe utilizar sangría y identación, evitar líneas largas de más de 80 caracteres.
- 15% **Documentación**, se debe aplicar la guía de diseño de programas y la especificación realizada en Doxygen para generar la documentación automática del código.
- 5% **Pruebas,** se debe crear un set de pruebas funcionales y de código.
- 10% ¡Usa tus **herramientas** sabiamente! No tenga condiciones o bucles innecesarios. Se trata del "arte" de la programación, más allá de simplemente "hacer que funcione".
- 20% Diseño **Orientado a Objetos**, definición y declaración de clases, funciones, relaciones entre clases (Herencia, Asociacion, Agrupacion, Composición), polimorfismo, manejo de excepciones entre otras.
- 10% **Propuesta de valor,** los proyectos serán similares ya que su grupo debe proponer algo diferenciador. (Interfaz de usuario, accesibilidad, Filtros o complementos a los desafíos)

ENTREGA PRELIMINAR

En el campus se encuentra una versión no Orientada a Objetos (O.O) de Panorama, revise detalladamente y aplique la guía de diseño de programas visto en clase (Nombres de clase, atributos, documentación de métodos , etc.). Con esta entrega se establecen las reglas de grupo para crear nuevas clases, atributos y métodos.

- (10%) Establecer un nombre para el equipo de trabajo. Por ejemplo. CerealKiller, Takedown, ZeroCool, CrashOverride, AcidBurn, Phantom, ThePlague o uno de su preferencia.
- (40%) Crear las carpetas:
  include: Donde se guardan los archivos header (*.h)
  src: Donde se guardan los archivos source (*.cpp)
  docs: Donde se guardan los archivos (html, doc y pdf) de la documentación asociada al proyecto.
  libs: Donde se guardan otras librerías externas/que el grupo no desarrollo
  test: Donde se guardan las clase e implementaciones que prueban.

- (30%) Completar los métodos que permiten la lectura de los diferentes formatos de imágenes, teniendo en cuenta que Panorama debe ser extensible.
- (10%) Crear una presentación formal del equipo de trabajo usando diapositivas y los avances que lograron.
- (10%) El proyecto debe ser compilado usando el comando make, usando el archivo Makefile o CMakeList.txt y el binario debe llamarse panorama.

AVANCE No. 1: DISEÑO / CONCEPTOS OO

- Identificar los atributos, los métodos y en caso de ser necesario las clases que tendrá el proyecto.
- Crear un diagrama del paso de mensajes entre las instancias de cada clase
- Configurar el generador automático de documentación.
- Crear un diagrama de clases resaltando las relaciones entre clases que se van considerar hasta esta versión.
- Incluir constructores por defecto, parametrizados, delegados y copia
- Realizar sobrecarga de métodos y/o funciones.
- Agregar los avances logrados a la presentación

AVANCE No. 2: RELACIONES ENTRE CLASES

- Revise el diagrama de clases, identifique la relaciones entre clases e implemente en su proyecto las relaciones (generalizaciones, agregaciones y asociaciones)
- Completar métodos, atributos y clases de acuerdo al diagrama
- Agregar los avances logrados a la presentación

AVANCE No. 3: POLIMORFISMO

- Revise el diagrama de clases (adjunto a este enunciado), identifique la relaciones entre clases e implemente en su proyecto las relaciones (generalizaciones, agregaciones y asociaciones)
- Se deben crear métodos virtuales que realicen operaciones sobre las imágenes sin importar el formato.
- Agregar los avances logrados a la presentación.

AVANCE No. 4: DESARROLLO DE DESAFÍOS

- Prototipo de la interfaz gráfica de usuario.
- Su versión debe ejecutarse y debe permitir trabajar con las imágenes de muestra propuestas en la carpeta *input*.
- Su versión debe permitir a través de la línea de comandos (CLI), ingresar la ruta de las imágenes a cargar y la operaciones a realizar.

ENTREGA FINAL

- Presentación formal del trabajo usando diapositivas y los desarrollos que se lograron. Revisar la sección de calificación

EJECUCIÓN

Para ejecutar el código se debe descargar la carpeta y usando la línea de comandos, debe escribir:

- cd build
- cmake ..
- make
- ./panorama

MÁS INFORMACIÓN

https://iie.fing.edu.uy/publicaciones/2005/Lec05a/Lec05a.pdf

https://courses.cs.washington.edu/courses/cse455/09wi/Lects/lect16.pdf

https://es.wikipedia.org/wiki/Disparidad_binocular