

Practicas Enero Junio 2019

Eduardo Alcaraz

March 2, 2019

Contents

1	Practica 1 Aspectos matemáticos de la graficación	2
2	Practica 2 Aspectos matemáticos de la graficación	2
3	Practica 3 Modelos de Color	3
4	Practica 4 Modelos De Color	3
5	Practica 5 Primitivas de Dibujo	3
6	Practica 6 Primitivas de Dibujo Transformaciones Geométricas	4
7	Practica 7 Primitivas de Dibujo Transformaciones Geométricas	5
8	Practica 8 Primitivas de Dibujo Transformaciones Geométricas	6
9	Practica 9 Primitivas de Dibujo Transformaciones Geométricas	7
10	Practica 10 Primitivas de Dibujo Transformaciones Geométricas	8
11	Practicas 11 Filtros de Convolución	9
12	Practicas 12 Filtros de Convolución	9

1 Practica 1 Aspectos matemáticos de la graficación

En la actualidad el ambiente que prevalece en los sistemas de cómputo es el visual, teniendo cada vez más interfaces vistosas y atrayentes; tomando en consideración esta perspectiva, un estudiante de ingeniería en sistemas computacionales debe conocer los elementos fundamentales que sirven de base para la creación de este tipo de entornos, así como, las diversas herramientas disponibles en el mercado.

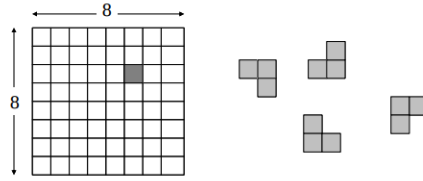
1. Programar un dibujo estilo 8 bits utilizando únicamente matrices con tipo de dato entero, donde cada casilla contiene un valor numero entre el rango de 0-255 como se muestra en la siguiente imagen.



2 Practica 2 Aspectos matemáticos de la graficación

En la actualidad el ambiente que prevalece en los sistemas de cómputo es el visual, teniendo cada vez más interfaces vistosas y atrayentes; tomando en consideración esta perspectiva, un estudiante de ingeniería en sistemas computacionales debe conocer los elementos fundamentales que sirven de base para la creación de este tipo de entornos, así como, las diversas herramientas disponibles en el mercado.

1. Dado un tablero de ajedrez programar el llenado del tablero utilizando las teselas mostradas en la siguiente imagen, es importante mencionar que el cuadro vacío puede estar en cualquier posición dentro del tablero.



3 Practica 3 Modelos de Color

En la actualidad el ambiente que prevalece en los sistemas de cómputo es el visual, teniendo cada vez más interfaces vistosas y atrayentes; tomando en consideración esta perspectiva, un estudiante de ingeniería en sistemas computacionales debe conocer los elementos fundamentales que sirven de base para la creación de este tipo de entornos, así como, las diversas herramientas disponibles en el mercado.

1. Programar 5 Operadores puntuales a la imagen de la practica 1
2. Programar el Histograma de una imagen utilizando cvLoad escala de gris.

4 Practica 4 Modelos De Color

En la actualidad el ambiente que prevalece en los sistemas de cómputo es el visual, teniendo cada vez más interfaces vistosas y atrayentes; tomando en consideración esta perspectiva, un estudiante de ingeniería en sistemas computacionales debe conocer los elementos fundamentales que sirven de base para la creación de este tipo de entornos, así como, las diversas herramientas disponibles en el mercado.

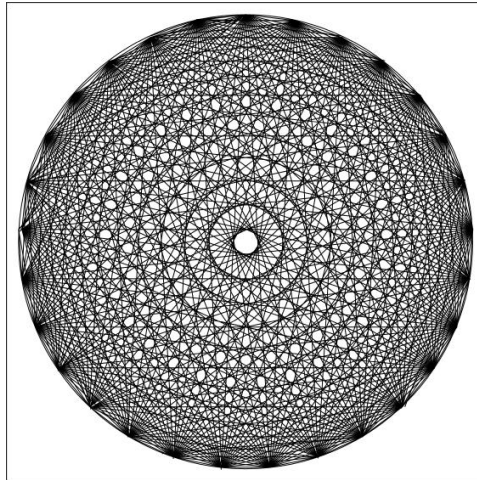
1. Programar un ejemplo por cada primitiva de dibujo de opencv.

5 Practica 5 Primitivas de Dibujo

Aprender a dibujar diferentes formas geométricas con OpenCV, utilizando las funciones: `cv2.line()`, `cv2.circle()`, `cv2.rectangle()`, `cv2.ellipse()`, `cv2.putText()`, etc.

Todas las funciones para dibujar, como las mencionadas anteriormente, tienen algunos argumentos comunes. A continuación se muestran algunos de ellos:

- `img`: La imagen donde se desea dibujar la forma geométrica
 - `color`: Color de la forma. En el caso del espacio de colores BGR, debemos pasar los valores como una tupla, ej: (255, 0, 0) para el azul. Por otra parte, para la escala de grises sólo se debe pasar un valor escalar.
 - `thickness`: Grosor de la línea, círculo, etc. En el caso de figuras cerradas como el círculo, un valor de `thickness=-1` llenará el interior de la figura. El valor por defecto de este parámetro es `=1`.
 - `lineType`: Tipo de línea, ya sea 8-conectada, suavizada, etc. Por defecto, es 8-conectada. Con `cv2.LINE_AA` se obtiene una línea suavizada que luce genial para las curvas.
1. Utilizando la primitiva de dibujo de opencv `cvline` dibujar un círculo con líneas conectando cada uno de los vértices

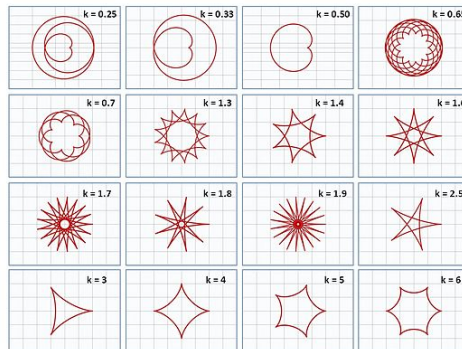


6 Practica 6 Primitivas de Dibujo Transformaciones Geométricas

Aprender a dibujar diferentes formas geométricas con OpenCV, utilizando las funciones: `cv2.line()`, `cv2.circle()`, `cv2.rectangle()`, `cv2.ellipse()`, `cv2.putText()`, etc.

Todas las funciones para dibujar, como las mencionadas anteriormente, tienen algunos argumentos comunes. A continuación se muestran algunos de ellos:

- `img`: La imagen donde se desea dibujar la forma geométrica
 - `color`: Color de la forma. En el caso del espacio de colores BGR, debemos pasar los valores como una tupla, ej: (255, 0, 0) para el azul. Por otra parte, para la escala de grises sólo se debe pasar un valor escalar.
 - `thickness`: Grosor de la línea, círculo, etc. En el caso de figuras cerradas como el círculo, un valor de `thickness=-1` llenará el interior de la figura. El valor por defecto de este parámetro es `=1`.
 - `lineType`: Tipo de línea, ya sea 8-conectada, suavizada, etc. Por defecto, es 8-conectada. Con `cv2.LINE_AA` se obtiene una línea suavizada que luce genial para las curvas.
1. Utilizando las primitivas de dibujo y ecuaciones paramétricas generar por lo menos 8 gráficas por cada función



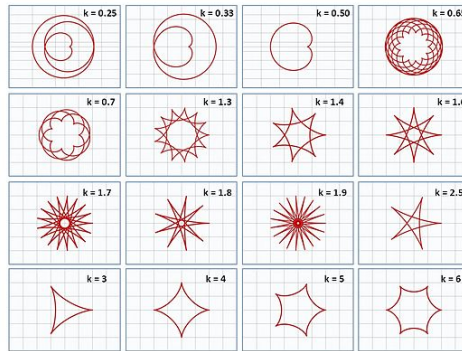
7 Practica 7 Primitivas de Dibujo Transformaciones Geométricas

Aprender a dibujar diferentes formas geométricas con OpenCV, utilizando las funciones: `cv2.line()`, `cv2.circle()`, `cv2.rectangle()`, `cv2.ellipse()`, `cv2.putText()`, etc.

Todas las funciones para dibujar, como las mencionadas anteriormente, tienen algunos argumentos comunes. A continuación se muestran algunos de ellos:

- `img`: La imagen donde se desea dibujar la forma geométrica

- **color:** Color de la forma. En el caso del espacio de colores BGR, debemos pasar los valores como una tupla, ej: (255, 0, 0) para el azul. Por otra parte, para la escala de grises sólo se debe pasar un valor escalar.
 - **thickness:** Grosor de la línea, círculo, etc. En el caso de figuras cerradas como el círculo, un valor de `thickness=-1` llenará el interior de la figura. El valor por defecto de este parámetro es `=1`.
 - **lineType:** Tipo de línea, ya sea 8-conectada, suavizada, etc. Por defecto, es 8-conectada. Con `cv2.LINE_AA` se obtiene una línea suavizada que luce genial para las curvas.
1. A partir de la practica 6, programar una esfera que se traslade por toda las curvas, de las ecuaciones antes programadas



8 Practica 8 Primitivas de Dibujo Transformaciones Geométricas

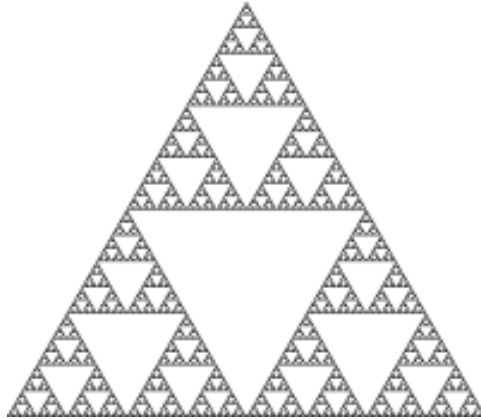
Aprender a dibujar diferentes formas geométricas con OpenCV, utilizando las funciones: `cv2.line()`, `cv2.circle()`, `cv2.rectangle()`, `cv2.ellipse()`, `cv2.putText()`, etc.

Todas las funciones para dibujar, como las mencionadas anteriormente, tienen algunos argumentos comunes. A continuación se muestran algunos de ellos:

- **img:** La imagen donde se desea dibujar la forma geométrica
- **color:** Color de la forma. En el caso del espacio de colores BGR, debemos pasar los valores como una tupla, ej: (255, 0, 0) para el azul.

Por otra parte, para la escala de grises sólo se debe pasar un valor escalar.

- **thickness:** Grosor de la línea, círculo, etc. En el caso de figuras cerradas como el círculo, un valor de `thickness=-1` llenará el interior de la figura. El valor por defecto de este parámetro es `=1`.
 - **lineType:** Tipo de línea, ya sea 8-conectada, suavizada, etc. Por defecto, es 8-conectada. Con `cv2.LINE_AA` se obtiene una línea suavizada que luce genial para las curvas.
1. Utilizando las primitivas de dibujo programar el triangulo de pascal utilizando recursividad.



9 Practica 9 Primitivas de Dibujo Transformaciones Geométricas

Aprender a dibujar diferentes formas geométricas con OpenCV, utilizando las funciones: `cv2.line()`, `cv2.circle()`, `cv2.rectangle()`, `cv2.ellipse()`, `cv2.putText()`, etc.

Todas las funciones para dibujar, como las mencionadas anteriormente, tienen algunos argumentos comunes. A continuación se muestran algunos de ellos:

- **img:** La imagen donde se desea dibujar la forma geométrica
- **color:** Color de la forma. En el caso del espacio de colores BGR, debemos pasar los valores como una tupla, ej: `(255, 0, 0)` para el azul.

Por otra parte, para la escala de grises sólo se debe pasar un valor escalar.

- **thickness:** Grosor de la línea, círculo, etc. En el caso de figuras cerradas como el círculo, un valor de `thickness=-1` llenará el interior de la figura. El valor por defecto de este parámetro es `=1`.
- **lineType:** Tipo de línea, ya sea 8-conectada, suavizada, etc. Por defecto, es 8-conectada. Con `cv2.LINE_AA` se obtiene una línea suavizada que luce genial para las curvas.

1. Programar el sistemas solar utilizando la transformada geométrica de translación

10 Practica 10 Primitivas de Dibujo Transformaciones Geométricas

Aprender a dibujar diferentes formas geométricas con OpenCV, utilizando las funciones: `cv2.line()`, `cv2.circle()`, `cv2.rectangle()`, `cv2.ellipse()`, `cv2.putText()`, etc.

Todas las funciones para dibujar, como las mencionadas anteriormente, tienen algunos argumentos comunes. A continuación se muestran algunos de ellos:

- **img:** La imagen donde se desea dibujar la forma geométrica
- **color:** Color de la forma. En el caso del espacio de colores BGR, debemos pasar los valores como una tupla, ej: `(255, 0, 0)` para el azul. Por otra parte, para la escala de grises sólo se debe pasar un valor escalar.
- **thickness:** Grosor de la línea, círculo, etc. En el caso de figuras cerradas como el círculo, un valor de `thickness=-1` llenará el interior de la figura. El valor por defecto de este parámetro es `=1`.
- **lineType:** Tipo de línea, ya sea 8-conectada, suavizada, etc. Por defecto, es 8-conectada. Con `cv2.LINE_AA` se obtiene una línea suavizada que luce genial para las curvas.

1. Programar una animación utilizando las transformada geométrica vistas en clase

11 Practicas 11 Filtros de Convolución

Un filtro de convolución, para una imagen digital, en el espacio real (X,Y) , puede representarse como una matriz cuadrada o rectangular (matriz de convolución), de dimensiones (M,N) mucho mas pequeñas que la des la imagen.

1. Programar filtro de Convolución para una matriz impulso

12 Practicas 12 Filtros de Convolución

Un filtro de convolución, para una imagen digital, en el espacio real (X,Y) , puede representarse como una matriz cuadrada o rectangular (matriz de convolución), de dimensiones (M,N) mucho mas pequeñas que la des la imagen.

1. Escalara la imagen creada en la Practica 1 por un factor de 2 y aplicar filtro de Convolución impulso.