

Drawing Arduino

Bárbara Mattioli

Sistemas Embebidos – Universidad Nacional Del Sur

Resumen

Este proyecto consiste en leer una imagen en la computadora y dibujar en papel real lo que aparece en ella. Se utilizan conceptos de procesamiento de imágenes, como el de obtener la intensidad de un píxel para diferenciar a aquellos que pertenecen al fondo de la imagen los que pertenecen a un dibujo dentro de ella. Para mover el lápiz, fibrón, lapicera, etc. se utiliza la estructura de un brazo robótico, el cual se mueve gracias a servomotores controlados por una placa Arduino UNO, conectados a esta mediante un shield. Para poder ubicar el lápiz en las posiciones correctas y así dibuje cada píxel, se utilizan ecuaciones trigonométricas para pasar de coordenadas cartesianas a coordenadas polares, debido a que la imagen se lee como una matriz, y los servos se mueven de forma circular.

Motivación y objetivos

El objetivo era lograr que un lápiz pueda dibujar automáticamente, sin intervención humana. Una de las motivaciones para este proyecto fue la de poder unir conceptos de materias distintas: Sistemas Embebidos y Procesamiento de Imágenes. Por otra parte, una gran motivación fue la de aprender a manejar motores, en particular servomotores, concepto visto únicamente de forma teórica en la materia y el cual me parecía muy interesante.

Descripción del problema

El problema a resolver consiste en que un software procese una imagen para descubrir cuáles son los píxeles que pertenecen a un dibujo en esta, y luego así, poder posicionar el lápiz adecuadamente sobre el papel para que pinte esos píxeles, y que de esta forma logre dibujar la imagen.

Herramientas de hardware

Con respecto al hardware, se utiliza una placa Arduino UNO para ejecutar el software que hace posible mover los servos. Estos están conectados a Arduino mediante un shield, que por cada servo provee una conexión a tierra y otra a 5V. Los servos se unen entre sí mediante reglas, que son soportadas con pequeñas cajas de cartón para mayor estabilidad de la estructura. Se utilizó pegamento para pegar los servos a las reglas y estas a las cajas que las sostienen. Además se usaron tornillos para fijar uno de los servos a un fibrofácil sobre el cual se coloca una cartulina en la que se realiza el dibujo de la imagen. Para dar alimentación a los tres servos, se conecta a Arduino una fuente de energía de 5V. Además se usa un cable USB para comunicar el Arduino UNO con la computadora.

Herramientas de software

Con respecto al software, se trabajó sobre el sistema operativo Linux. Se hizo uso del entorno de desarrollo Arduino IDE y el lenguaje C++ para el software que se ejecuta sobre la placa Arduino. Se hizo uso de la librería Servo para manejar los servomotores, y la librería LiquidCrystal para hacer posible la comunicación con el software del procesamiento de la imagen. Este último se escribió en el lenguaje Python, haciendo uso de la librería OpenCV para procesar la imagen y de la librería PySerial para comunicarse con Arduino.

Descripción de la solución

Procesamiento de la imagen

Primero que nada, se busca la imagen a partir de una ruta de una imagen que el usuario escribe en una terminal de comandos luego de ejecutar el código. Una vez que encuentra la imagen, se la comienza a leer como una matriz en la que cada elemento es un píxel. Por cada píxel, se observa su intensidad. Se asume un fondo blanco, es decir que la intensidad de un píxel que pertenezca al fondo de la imagen es de 255. Si tiene un valor distinto de este, el píxel pertenece al dibujo que deberá realizar el lápiz automático. En ese caso, se envía un paquete a Arduino con las coordenadas cartesianas de ese píxel. Además, se envía como parte del paquete, un 1 si el píxel no es vecino del píxel dibujado anteriormente, o un 0 si sí lo es. De esta forma, en Arduino se sabrá si se debe levantar o no el lápiz para llegar a su siguiente destino. Este paquete será considerado válido si el primer elemento y el último tienen el número 0. Las coordenadas se mapean de forma directa a la fila y la columna del píxel de la imagen, es decir que la coordenada Y es el número de fila, y la coordenada X es el número de columna. Luego de enviar la información a Arduino, se verificará si existen píxeles vecinos que sean parte del dibujo, antes de seguir con el recorrido general de la imagen. Además, por cada píxel que se envía a Arduino, se registra que se procesó ese píxel para no volverlo a enviar más adelante. Es decir, que por cada píxel, se verificará que no se haya procesado con anterioridad.

Pasaje de Coordenadas

En Arduino IDE, se recibe la información desde el software de procesamiento de la imagen. Antes que nada, se verifica que se haya recibido un paquete válido. Si es así, se lee la información para encontrar las coordenadas cartesianas de la siguiente posición del lápiz y para saber si tiene que levantar o no el lápiz para llegar a la nueva posición. Luego, realiza una conversión a coordenadas polares a través de ecuaciones las trigonométricas que se detallan más adelante. Si tiene que levantar el lápiz para alcanzar el nuevo punto, mueve el servo correspondiente en un ángulo constante, antes que los demás. Por último, con los resultados de las ecuaciones trigonométricas, se actualiza los ángulos de los servomotores restantes para posicionar el lápiz en la nueva ubicación.

Estructura Física

Para hacer posible el movimiento automático del lápiz, se implementó un brazo robótico. De la misma forma que un brazo humano, posee un hombro y un codo, simulados con servomotores, cuyos movimientos cambian la posición del punto final del brazo. En un brazo humano, el punto final sería la mano. En este caso, el punto final es el lápiz. El servo que simula al hombro debe estar fijo sobre una superficie. En este caso, esta atornillado a un fibrofácil. El que simula al codo no debe estar fijo para que su posición cambie según el ángulo del servo que simula al hombro.

Cinemática Inversa

La cinemática inversa es una técnica muy utilizada en robótica para calcular los movimientos de las articulaciones de un brazo para que el punto final de este alcance el destino correcto, que originalmente esta representado en coordenadas cartesianas. Esta técnica es la que se utiliza para calcular los ángulos correspondientes a los servos que simulan el hombro y el codo del brazo. El ángulo q_1 es el del servo fijo, mientras que q_2 es el del servo móvil. El ángulo del servo fijo dependerá del ángulo del móvil para posicionar el punto final en el destino correcto. Esto se demuestra en las siguientes ecuaciones, que son las utilizadas en la conversión de coordenadas.

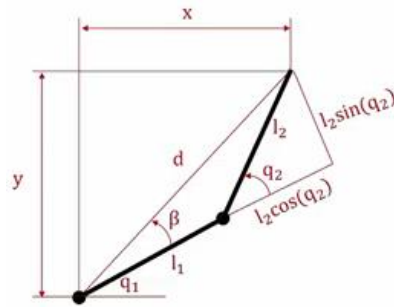
- Cálculos para obtener q_1 :

$$\tan(q_1 + \beta) = \frac{y}{x}$$

$$q_1 = \text{atan}\left(\frac{y}{x}\right) - \beta$$

$$\tan(\beta) = \frac{l_2 \sin(q_2)}{l_1 + l_2 \cos(q_2)}$$

$$\beta = \text{atan}\left(\frac{l_2 \sin(q_2)}{l_1 + l_2 \cos(q_2)}\right)$$



- Cálculos para obtener q2:

$$q_2 = \Pi - \frac{l_1^2 + l_2^2 - d^2}{(2 * l_1 * l_2)}$$

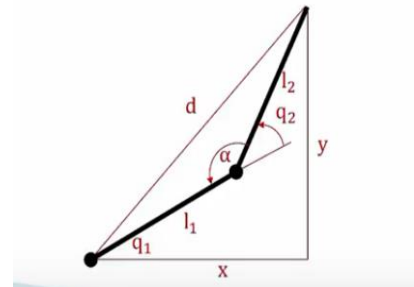
$$\cos(q_2) = \cos(\Pi - \alpha)$$

$$\cos(q_2) = \cos(\Pi)\cos(\alpha) + \sin(\Pi)\sin(\alpha)$$

$$\cos(q_2) = -\cos(\alpha)$$

$$\cos(q_2) = \frac{d^2 - l_1^2 - l_2^2}{(2 * l_1 * l_2)}$$

$$q_2 = \text{acos}\left(\frac{d^2 - l_1^2 - l_2^2}{(2 * l_1 * l_2)}\right)$$



Referencias

Idea del proyecto: <https://www.instructables.com/Drawing-Robot/>

Pasaje de cartesianas a polares: <https://www.youtube.com/watch?v=RH3iAmMsolo>

Comunicación desde Python a Arduino: <https://pypi.org/project/pyserial/>

Procesamiento de la imagen: <https://opencv.org/>