

Proyecto Estructuras Discretas: Sistema de detección de obstáculos usando una cámara web

Bruno Solís Diego Román
García Gutiérrez Edgar Cristóbal
López Gonzalez Kevin
Moreno Peralta Angel Eduardo

20 de Mayo 2020

1. Resumen

La geometría computacional es una rama de la ciencias de la computación dedicada al estudio de los algoritmos que pueden ser expresados en términos de la geometría. Gracias a esta rama de la computación han surgido muchas optimizaciones en diversas áreas de la vida cotidiana, tales como los mapas interactivos (como Google Maps), el poder visualizar el tráfico en vivo de alguna parte de nuestra localidad (como Waze), automóviles autónomos (Tesla), etcétera.

Debido al gran potencial que tiene esta rama de la ciencia se decidió buscar una optimización que nos ayude a solucionar un problema de la vida real usando conceptos de matemáticas discretas, y geometría euclidiana.

Un problema clásico de la geometría computacional es el llamado *envolvente convexo* (en inglés *Convex Hull*), este problema plantea que dado un conjunto de puntos, se crea un polígono (por lo general irregular) el cual en su interior tendrá almacenados a los puntos de dicho conjunto. Este es un problema clásico de la geometría computacional y existen diversos algoritmos que resuelven este problema tales como: Graham Scan, QuickHull, Fuerza Bruta etc.

Gráficamente se representa así el problema:

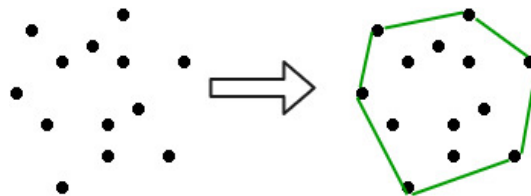


Figura 1: Representación grafica de Convex Hull

Convex Hull ha demostrado tener implicaciones importantes dentro del campo de estudio de la realidad virtual, debido a que con ayuda de los algoritmos se puede realizar reconocimiento de patrones de los objetos, o poder interactuar con una maquina y que esta reconozca al objeto con el que interactua.

2. ¿Qué hicimos?

La propuesta del proyecto fue realizar una aplicación usando la metodología de Convex Hull. La aplicación seleccionada fue un sistema que evita que algún objeto colisione con otro. El programa se realizó en Python, debido a que en dicho lenguaje de programación se cuentan con las bibliotecas necesarias para manejar el hardware necesario en este proyecto.

Para el correcto funcionamiento del programa necesitaremos tener una cámara web, los objetos para hacer las pruebas y un fondo de color. También funciona con cuarto oscuro y una lampara de escritorio, en caso de no tener el fondo uniforme. El programa mostrara una ventana con la imagen en

vivo de lo que capte la cámara web. Antes de encontrar los puntos del objeto, procesamos la imagen de entrada.

Primero se convierte en formato de blanco y negro. Después, se le aplica una umbralización para eliminar el posible ruido y aislar el objeto a detectar. Luego, aplicamos una función que detecta los bordes que encuentre en la imagen. Finalmente, aumentamos el tamaño de estos bordes para corregir posibles rupturas de contornos.

Ahora se aplica a la imagen una función de detección de contornos, que nos devuelve un arreglo de todos los puntos que conforman dicho contorno del objeto. El arreglo se transforma a una lista para que sea más sencillo de iterar y trabajar.

Con dicha lista de puntos podemos aplicar el algoritmo para construir el Convex Hull, dicho algoritmo se trata de Graham Scan debido a su facilidad de aplicar. Con el Convex Hull hecho guardamos los puntos que formen parte de las aristas de nuestro Convex Hull. Para saber si el objeto va a entrar en colisión uno de los puntos que estén contenidos en las aristas del Convex Hull tienen que ser mayores o iguales a los puntos que están contenidas en las aristas de un rectángulo, este rectángulo y la construcción del Convex Hull se pueden apreciar en la ventana que muestra lo que esta capturando la cámara web (vista normal).

La lógica de este programa se basa en la funcionalidad del ojo humano. El fundamento de esto es que cuando vemos, de manera física, muy cerca un objeto, esto implica que estamos físicamente muy cerca de ese objeto. Esto en nuestro programa se traduce en que cuando el Convex Hull salga de nuestro campo de visión (rectángulo), implicaría que este objeto esta muy cerca y por lo tanto el riesgo de colisión es muy alto.

Debido al hardware disponible por los autores, este programa tiene ciertos limites tales como el requerimiento de un buen fondo y hardware sofisticado para la detección de movimiento, sin embargo con lo hecho en este proyecto logramos lo que queremos demostrar.

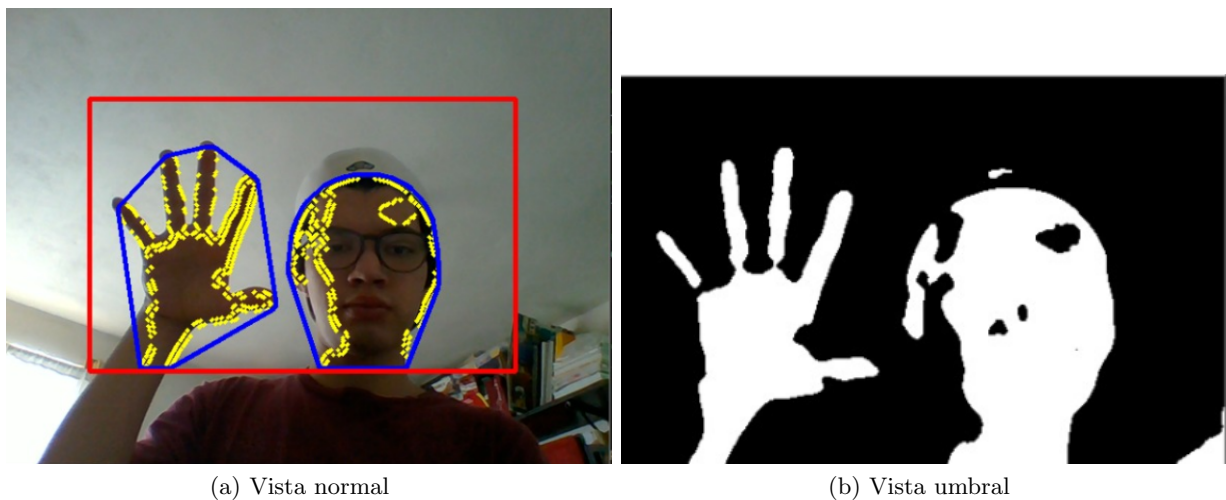


Figura 2: Funcionamiento del programa

3. Diagrama de Flujo

