

TDP : Chiv-One

José Antonio Betances Vargas
joseantoniobetances@hotmail.com

Abstracto: En este documento, se hace una descripción breve de Chiv-One, un robot participante en la categoría “Open” de la VII Competencia Latinoamericana de Robótica. Luego, se hará una descripción de su algoritmo de reconocimiento de imágenes, y su posible aplicación en el uso doméstico.

I. Introducción

En la actualidad, los desarrollos tecnológicos de la robótica apuntan al avance de varias vertientes al mismo tiempo.

Mientras la empresa Honda realiza robots humanoides con tecnología mecánica de punta; las empresas Seiko y Epson dicen haber creado el robot volador más pequeño del mundo. El británico-canadiense Mark Tilden vive haciendo robots-insectos que mezclan de una forma extraña lo analógico con lo digital; múltiples empresas fabrican juguetes robotizados para suplir las necesidades de los niños actuales, y cada vez es más común ver empresas que desechan casi por completo la mano del hombre para reemplazarla por robots autómatas en fábricas e industrias.

Pero aun cuando la robótica parece estar muy avanzada, aun quedan varios aspectos que requieren de muchos años de investigaciones científicas, como lo es la “inteligencia” de un ente robótico ante una imagen tomada del mundo que lo rodea.

Así, aun cuando hay multiplicidad de robots que siguen líneas dibujadas en el piso, es común ver como las pequeñas fluctuaciones de la luz hacen que tales robots queden fuera de servicio, ya que estos no cuentan con un algoritmo bien elaborado para poder sobrepasar tales dificultades.

En este artículo, se presentará de una forma un tanto generalizada, el algoritmo de reconocimiento de imágenes que tiene Chiv-One, y su posible aplicación en el mundo fuera de la competencia para la cual fue creado este robot.

II. El robot

El robot que se ha implementado para resolver los problemas descritos en la categoría Open (abierta), consiste básicamente en un “carrito” con una CMUCAM 3, sensores infrarrojos, un PIC 16F876, y solo 6 servos.

Cuatro de estos servos están destinados a las ruedas (un servo por rueda); un servo se encarga de dar movimiento vertical a la cámara, y el sexto servo se encarga de mover horizontalmente un gancho, que no es más que un alambre convenientemente doblado.

La mecánica del robot es sumamente sencilla, aun cuando el problema que se tiene

que resolver en los requerimientos de la categoría sean relativamente complejos.

Se optó por hacer el robot con tracción a las cuatro ruedas por dos razones fundamentales:

1. Subir la cuesta de 45° sin problemas
2. Permitir que el robot gire sobre su propio eje ocupando un mínimo de espacio.

La razón N° 1 fue la que dio la inspiración para el nombre del robot: “**Chiv-one**”. Este nombre es una combinación de la palabra “Chivo” (el cual es un animal que acostumbra a subir cuestas muy empinadas con total confianza), y la palabra inglesa “One” (uno), ya que esta es mi primera versión de un robot que se construye para estos fines.

El robot (Figura 1) se hizo con el centro de gravedad lo más bajo posible, de forma que le sea fácil subir la cuesta sin resbalarse o voltearse hacia atrás. Este diseño permite una agilidad tremenda, ya que el robot puede subir la rampa de 45° en solo 2 segundos.

El robot, para jalar los cables, lo que hace es, básicamente, colocarse al frente del cubo, “mirar” los cables, y según un reconocimiento del color del cable, posiciona el gancho, avanza, sujeta el cable con el gancho, y retrocede. En dado caso que el cable no este ubicado en la posición correcta, el robot rodea a la bomba, hasta que consiga sacar el cable del color determinado.

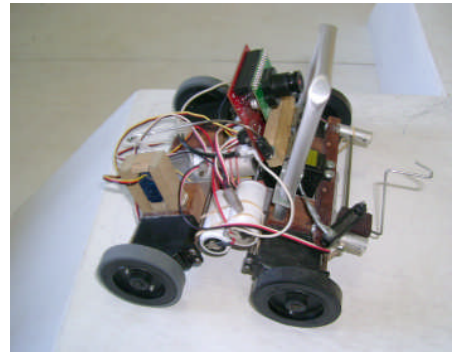


Figura 1: Foto de Chiv-One

El robot tiene una inteligencia basada en los sistemas expertos. Aun cuando su diseño no va mas allá de los requerimientos de la categoría, su algoritmo de reconocimiento de imágenes le permite hacer su trabajo muy bien.

II. La CMUCam 3

Esta cámara, fue diseñada por la Carnegie Mellon University.

Consiste básicamente en un fotorreceptor cmos de la firma Omnivision, y un microcontrolador que la maneja, con 2 entradas y salidas seriales, a demás de un puerto para 4 servomotores, y una hendidura para introducir memorias Smart-Card (vease www.cmucam.org).

Esta cámara pretende dar a los usuarios que requieren de la manipulación de imágenes, una especie de producto “todo-en-uno”, donde una persona que sepa de algo de programación en C, pueda, con un poco de ingenio, hacer sus robots con visión “inteligente”.

La CMUCam 3, muestra serias deficiencias en cuanto a su estructura, su funcionalidad y su desempeño. Por un lado, aun cuando la idea de integrar una cámara a un microcontrolador pueda ser algo bastante

útil, la CMUCam 3 requiere de muchas mejoras, si es que se les quiere integrar a un uso que valla mas allá de hacer juguetitos robotizados.

Así, uno de los grandes defectos de esta cámara, es que no tiene suficiente memoria RAM para cargar tan siquiera una cuarta parte de una imagen en baja resolución. Esto es algo muy lamentable, considerando que el dispositivo pretende precisamente esto: Manipular imágenes.

Esto trae como consecuencia que las librerías de reconocimiento de color que se entregan con la CMUCam 3 sean de pésima calidad, ya que examinan la imagen de a píxel por píxel y no por conjunto de píxeles. Por eso, si uno quisiese implementar un filtro, depararía en un problema muy grande, ya que no hay forma “evidente” de manipular la imagen tomada por el receptor Omnivision.

El uso de las librerías adjuntas a la CMUCam 3 trajo varios problemas serios a la funcionalidad del robot. La librería “track_color”, la cual se encarga de examinar la imagen, y hacer una especie de “atención” sobre un determinado color, parece funcionar muy bien en la teoría, pero cuando se le usa en la práctica, surgen varios problemas serios de funcionalidad.

Si por ejemplo, uno quisiera que la cámara prestara su atención en el cubo azul de la Figura 2, habrían varios problemas técnicos, ya que la CMUCam 3 solo examina píxeles individuales, y englobaría, muy a nuestro pesar, ese píxel azul afuera del cubo (la marca en la camisa del hombre que esta atrás del cubo) conjuntamente con este, tomando una lectura errónea (Figura 3 y 4).

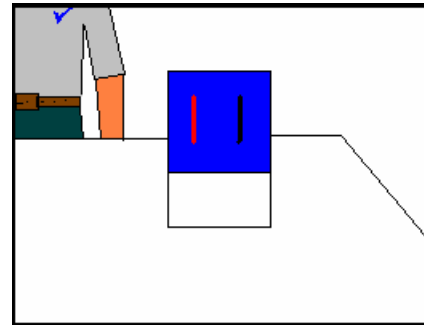


Figura 2: Imagen tomada por la cámara

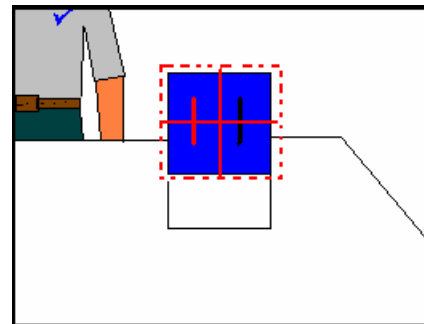


Figura 3: Comportamiento de la capción que se espera de la cámara.

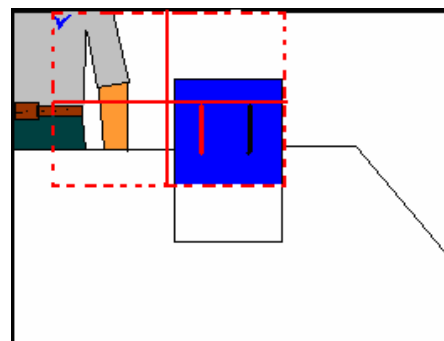


Figura 4: Comportamiento real de la capción de la cámara. Véase que engloba a la marca de la camisa.

El robot fundamenta su funcionamiento en la alineación para posicionarse en el sitio indicado al frente del cubo. Si la lectura de la atención no funciona bien, el robot fallara instantáneamente en su labor.

Para solventar esto, se tuvo que aplicar un algoritmo que usaría la Smart-Card, conjuntamente con una interfase entre la

Cámara y un PIC 16f876 que a su vez esta conectado a sensores infrarrojos de distancia, para reducir así, todos estos problemas en lo mas mínimo posible.

III. El algoritmo

En vista de que la memoria RAM escasea en la CMUCam 3, se tuvo que usar en una suerte de memoria virtual a la Smart-Card, la cual guarda la imagen, y se va modificando columna por columna mediante procesos alojados en la RAM del microcontrolador nativo de la cámara.



Figura 5. Imagen original

Partiendo de la imagen original tomada por la cámara (Figura 5), se le aplicó un gradiente, para obtener así, el borde. El resultado se muestra claramente en la Figura 6.

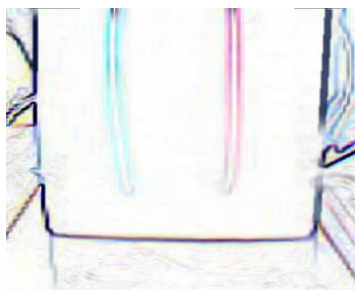


Figura 6. Imagen con gradiente

Luego, se le aplicó un filtro para resaltar las líneas oscuras y se cambió la imagen a tono de grises. El resultado se muestra en la Figura 7.

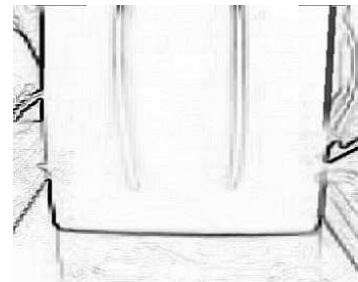


Figura 7. Imagen en escala de grises

Luego, se uso un algoritmo que rastrea la densidad de los píxeles, a través del desplazamiento de franjas de atención en el eje horizontal y vertical. Este rastreo, genera un grafico como el que sigue, donde el eje de las X representa la posición de la franja de rastreo (tanto horizontal como vertical), y el eje de las Y, representa la densidad. Se puede ver claramente en la Figura 8.

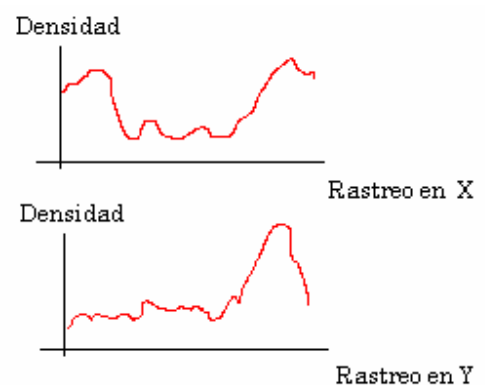


Figura 8. Grafico de densidad en el eje de X y Y de la imagen.

Como se puede observar, el grafico del rastreo horizontal tiene dos picos altos. Estos dos picos representan las dos líneas verticales que separan al cubo del resto de la

imagen. Un algoritmo de búsqueda consigue estos picos y su ubicación. De esta forma, la cámara reconoce parámetros como el ancho del cubo (y por ende, la distancia desde que se tomó la imagen). Si se combinan los picos del gráfico en X y en Y, y se hacen intersecciones, se puede deducir también la posición relativa del robot frente al cubo.

IV. Resultados

Este algoritmo tiene la ventaja de que no importa que se atravesara atrás del cubo, Siempre podrá diferenciarlo del resto. En dado caso que se atravesara alguien vestido con franjas verticales u horizontales, un algoritmo sencillo se encarga de buscar, dentro de todos los picos, los más lógicos correspondientes a un cubo de medidas perfectas.

También se puede utilizar para saber que tantos cables hay en ese lado del cubo al que se le está tomando la foto.

Los gráficos de densidad muestran ser invulnerables a los cambios de luz, cosas

cotidianas considerando que la iluminación varía dependiendo de la hora del día.

Anteriormente, había que ajustar a cada rato los rangos de la captación de la CMUCam 3. Ahora, los ajustes se reducen a un mínimo.

V. Conclusiones

Más allá de la utilidad de este algoritmo en el robot, se le puede utilizar en varias aplicaciones. Los gráficos de densidad pueden ser utilizados para todo tipo de reconocimiento de formas, además de las cámaras de vigilancia, las cuales tienden a ser susceptibles a los cambios de luz.

Nadie quiere que una alarma se active solo porque la iluminación cambie repentinamente. Los gráficos de densidad son prácticamente invulnerables a tales cambios.

Esta es una de las muchas aplicaciones que se les pueden dar a estos gráficos.