Practica 3 Visión Artificial

Manuel Alejandro Chalarca Botero1

Santiago Valencia Cortes 2

1 Ingeniería Mecatrónica

Facultad de Ingeniería

Instituto Tecnológico Metropolitano (ITM)

[Manuelchalarca281009@correo.itm.edu.co](mailto:Manuelchalarca281009@correo.itm.edu.co)

2 Ingeniería Mecatrónica

Facultad de Ingeniería

Instituto Tecnológico Metropolitano (ITM)

[Santiagovalencia248959@correo.itm.edu.co](mailto:Santiagovalencia248959@correo.itm.edu.co)

**INSTITUTO TECNOLOGICO METROPOLITANO**

**5 DE MARZO DEL 2022**

1. **Metodología**

**1.1 Métodos de Iluminación usados en visión artificial.**

Se investigaron las diferentes formas de iluminar un objeto con el objetivo de identificar las utilidades, usos y desventajas particulares de cada método dependiendo de la aplicación y circunstancia que se requiera para conseguir las condiciones óptimas de visualización para obtener la mayor cantidad posible de información, o también para filtrar de acuerdo a las necesidades del proyecto en cuestión.

**1.2 Longitud de código de barras y método otsu.**

Con la información anterior se experimentaron con 3 tipos de iluminación los cuales fueron iluminación frontal, iluminación lateral e iluminación por campo oscuro. Luego con el método más apropiado para el lector de codigo de barras, el cual fue la iluminación lateral, se desarrolló un código que permite contabilizar el grosor de todas las longitudes de las líneas en un código de barras, una vez conseguido esto, se investigó sobre el método Otsu para umbralizaciones y el cómo podía mejorar la umbralización de ciertas imágenes.

**1.3 Algoritmo Findcontours**

La función findContours de opencv es muy utilizada como su nombre lo dice nos sirve para encontrar los contornos de una imagen. lo que realmente hace esta función es detectar el cambio de color de una imagen y este cambio de color o de intensidad lo marca como un contorno, esta operación se cumple mejor en imágenes binarizadas ya que hay cambios de intensidad altos de 255 o 0 y esto hace que detecte mucho más fácil el borde de la figura la parte que deseas detectar debe ser blanca, los parámetros que recibe esta función son una imagen que debe ser binarizda luego mode que son los modos de recuperación de contorno el tercer parámetro es method que es el método de aproximación de contorno luego offset que es el desplazamiento opcional mediante el cual se desplaza cada punto del contorno y las salidas modificadas son la imagen los contornos y la jerarquía los contornos son una lista de Python de todos los contornos de la imagen. Cada contorno individual es una matriz Numpy de coordenadas (x, y) de puntos límite del objeto.

hay varias formas de métodos de aproximación esta cv.CHAIN\_APPROX\_NONE, y también esta cv.CHAIN\_APPROX\_SIMPLE siendo esta última mucho mejor ya que ahorramos memoria y también evitamos que haya puntos redundantes

**1.4 Segmentacion de color**

Se realizo la segmentación en tiempo real en dos espacios de color diferentes uno es HSV y el otro es CIELAB para el caso de HSV a diferencia del RGB no sigue coordenadas euclidianas este modelo HSV sigue su modelo más parecido a las coordenadas cilíndricas donde H representa tonalidad o tinte S hace referencia a la saturación y por último tenemos V que se refiere al brillo o valor para el modelo CIELAB tenemos que este funciona con coordenadas esféricas donde L representa la luminosidad de un objeto la A la variación de verde a rojo y B la variación de azul a amarillo.

**1.5 Detector de geometría**

Se procedió a crear un algoritmo capaz de detectar en este caso una circunferencia, sin utilizar ninguna de las herramientas aquí consultadas como el findcontours, lo que complicó la tarea de gran manera

**1.6 Operaciones Morfológicas**

Se partía de Fig. 4 y la Fig. 5 a las cuales se les deben aplicar diferentes operaciones morfológicas para acercarse a la forma de la Fig. 6 y Fig. 7 respectivamente.



Fig. 4 (Recurso del curso)



Fig. 5 (Recurso del curso)



Fig. 6 (Recurso del curso)

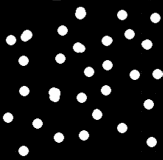


Fig. 7 (Recurso del curso)

1. **Análisis y Conclusiones**

**2.1 Métodos de Iluminación**

Las técnicas de iluminación permiten aprovechar al máximo las aplicaciones de visión artificial. Son fundamentales a la hora de determinar la imagen que se obtiene ya que, en última instancia, la luz es la materia prima con la que trabajará cualquier cámara de visión artificial, pertenezca a un rango y otro del espectro de luz. Una técnica de iluminación utiliza una fuente de luz y su ubicación con respecto a la pieza y la cámara. Algunos de los métodos son los siguientes.

1. **Iluminación Frontal**

La cámara se ubica mirando al objeto y en la dirección de la luz. Esta técnica de iluminación es especialmente útil en superficies con pocos reflejos y se logra mediante anillos de luz, iluminadores puntuales de área y lineales.

Aplicaciones: Indicada para superficies con pocos reflejos: papel, tela… para la detección de marcas de diferentes colores, caracteres y detección de todo lo que suponga un cambio de color en prácticamente cualquier superficie.

Ventajas: Elimina sombras, se puede utilizar a grandes distancias cámara/objeto.

Inconvenientes: Intensos reflejos sobre superficies reflectantes.

1. **Iluminación Lateral**

Esta técnica se utiliza para resaltar los detalles de las piezas. Estos solo serán visibles en el caso de que la luz esté orientada de forma lateral a la posición de la cámara. De este modo, se define la iluminación lateral como aquella en la que la fuente de luz se sitúa en un lado del objeto que se quiere fotografiar y que crea, necesariamente, un ángulo de 90º entre la cámara y el objeto en cuestión.

Aplicaciones: Indicada para resaltar bordes, rayas y fisuras en una dirección determinada.

Ventajas: Resalta los relieves por pequeños que sean de los objetos, resultando una sombra muy definida.

Inconvenientes: Con ángulos pequeños respecto a la horizontal, la luz producirá sombras en todos los relieves y en el contorno de la pieza.

1. **Iluminación Por campo Oscuro**

En esta técnica, el objetivo es resaltar los defectos superficiales (tales como grietas, surcos o manchas). Para poder detectarlos, se deberán detectar los códigos matriz u otros caracteres impresos en la superficie. En este caso, lo más habitual será utilizar anillos luminosos que emitan luz en dirección perpendicular a la cámara.

Aplicaciones: Indicada para resaltar incrustaciones y códigos alfanuméricos con poco contraste en metal sobre metal o gris sobre gris. Muy utilizada en la verificación de grabados tipo láser o troquel.

Ventajas: Destaca los detalles en superficies con muy poco contraste.

Inconvenientes: No es recomendable en superficies que absorban la luz.

1. **Iluminación Por Contraste (Backlight)**

Esta técnica sitúa el objeto entre la iluminación y la cámara, con lo que se consigue reconocer la silueta del objeto por contraste.

Aplicaciones: Indicada para la inspección de la silueta del objeto. Utilizada también en materiales translúcidos o transparentes para detectar manchas, rayas, grietas, etc.

Ventajas: Permite inspecciones de siluetas con mediciones muy precisas y de impurezas en los objetos transparentes o translúcidos.

Inconvenientes: No permite reconocer los detalles superficiales del objeto, códigos, inscripciones, etc.

1. **Iluminación sobre el mismo eje o iluminación coaxial**

Se usa para iluminar los objetos reflectantes. Estas técnicas de iluminación funcionan emitiendo luz de manera lateral sobre un espejo semitransparente y que se desvían en la misma dirección al eje de la cámara, lo que consigue una luz difusa y homogénea.

Aplicaciones: Indicada para la inspección superficies planas reflectantes, como PCB, etiquetas reflectantes, inspección de impresión sobre aluminio o cavidades profundas.

Ventajas: Permite inspecciones de códigos en materiales altamente reflectantes.

Inconvenientes: No permite reconocer relieves en el objeto.

1. **Iluminación Difusa tipo domo plano**

Se usa para iluminar los objetos reflectantes. Estas técnicas de iluminación funcionan emitiendo luz de manera lateral sobre un espejo semitransparente y que se desvían en la misma dirección al eje de la cámara, lo que consigue una luz difusa y homogénea.

Aplicaciones: puede aplicarse a todas las iluminaciones direccionales (barras, domos, anillos) para brindar una propagación de la luz más uniforme en todo el objetivo.

Ventajas: Esta técnica de iluminación es útil para disminuir el ruido de la iluminación como el brillo y los puntos importantes de luces direccionales en el eje, tales como luces de anillo.

1. **Iluminación Difusa tipo domo**

La luz es emitida dentro de una cúpula esférica resultando una luz difusa desde todas direcciones, eliminando sombras y reflejos, suavizando texturas y minimizando la influencia de rayas, polvo, relieves y curvaturas que pueda tener el objeto inspeccionado. A este tipo de iluminación también se le denomina iluminación de día nublado por no producir ningún tipo de sombra al objeto.

Aplicaciones: Indicada para la inspección de superficies tales como: instrumental médico, espejos, compact disk, latas, etc.

Ventajas: Eliminación de sombras y minimización de arrugas, polvo y relieves.

Inconvenientes: Coste elevado.

**2.2** **Longitud de barras y método otsu**

Se programó un algoritmo que bajo las condiciones de luz adecuada, es capaz de decir la longitud en píxeles de cada una de las líneas verticales que componen un código de barras como se muestra en la figura 8

Fig. 8 Resultado del algoritmo (recurso propio)

La línea roja central es la encargada de hacer la lectura en el área delimitada por las líneas verdes mostrando la longitud de cada línea respectiva en laparte inferior de la imagen mostrada por la cámara.

Para el ejemplo de prueba se utilizó un video de 24 segundos el cual consta de 3 códigos de barras de diferentes productos que se van retirando dejando ver el siguiente. La imagen binarizada es la utilizada por el programa para hacer la lectura.

El método de iluminación utilizado fue la iluminación lateral (Fig. 9), ya que esta era la que menos interferencia generaba al momento de binarizar la imagen (Fig. 10), permitiendo distinguir con claridad las líneas lo suficiente para generar una lectura estable (sin contar el balanceo de cámara) de la longitud de las barras.

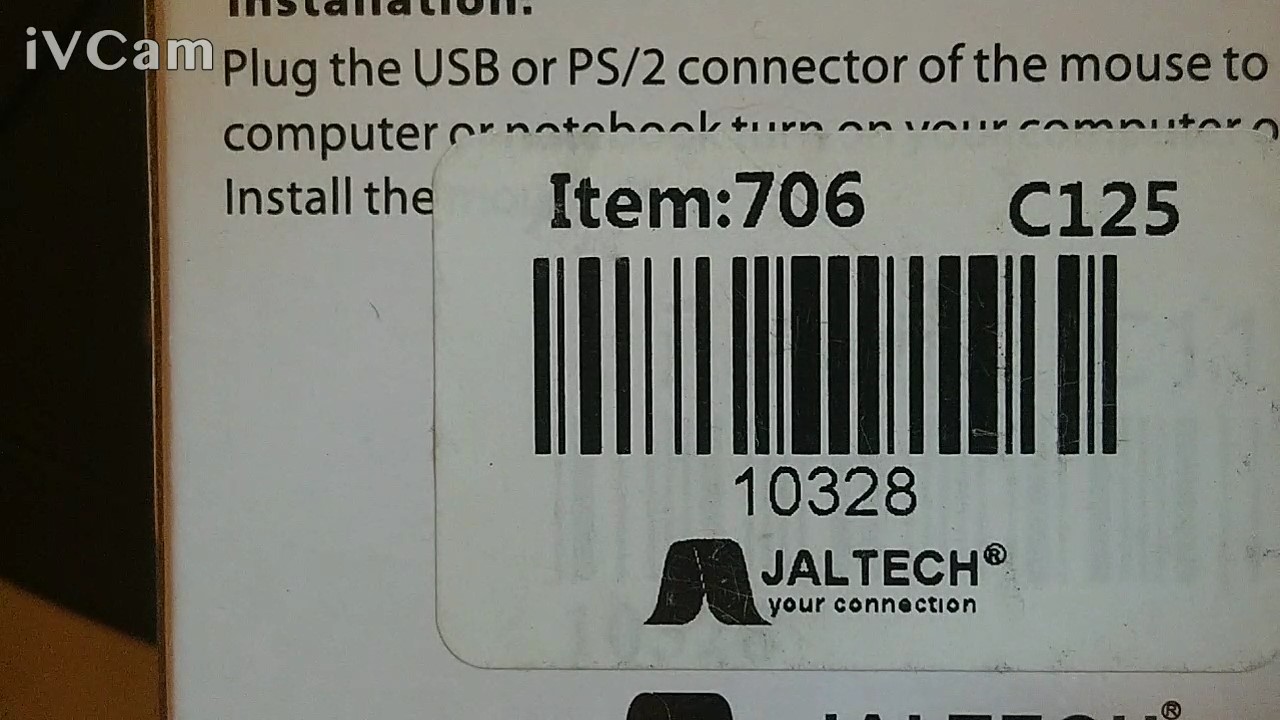


Fig. 9 Luz Lateral (Recurso propio)



Fig. 10 Binarizado Luz Lateral (Recurso propio)

También se probó la iluminación frontal o directa (Fig 11), pero esta causaba la mayor interferencia a causa del reflejo de la luz sobre el material del código de barras, la cual se aumentaba dependiendo de la reflectancia del material causando una binarización ilegible.

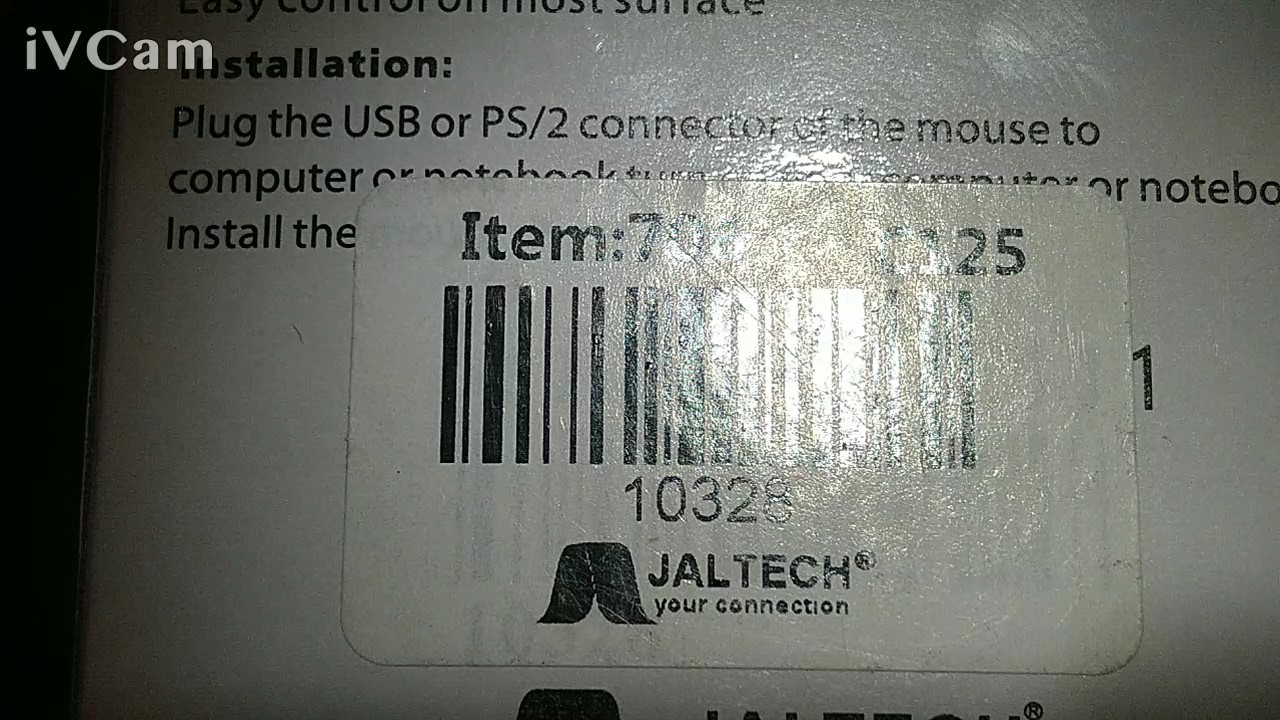


fig.11 Luz Frontal (recurso propio)



fig.12 Binarización Luz Frontal (recurso propio)

Por último se utilizó la iluminación por campo oscuro (Fig. 13) la cual es excelente para detalles de la superficie, pero como era de esperarse de una etiqueta solo se podía captar la línea de relieve de la calcomanía que contiene el código de barras en el binarizado (Fig. 14)

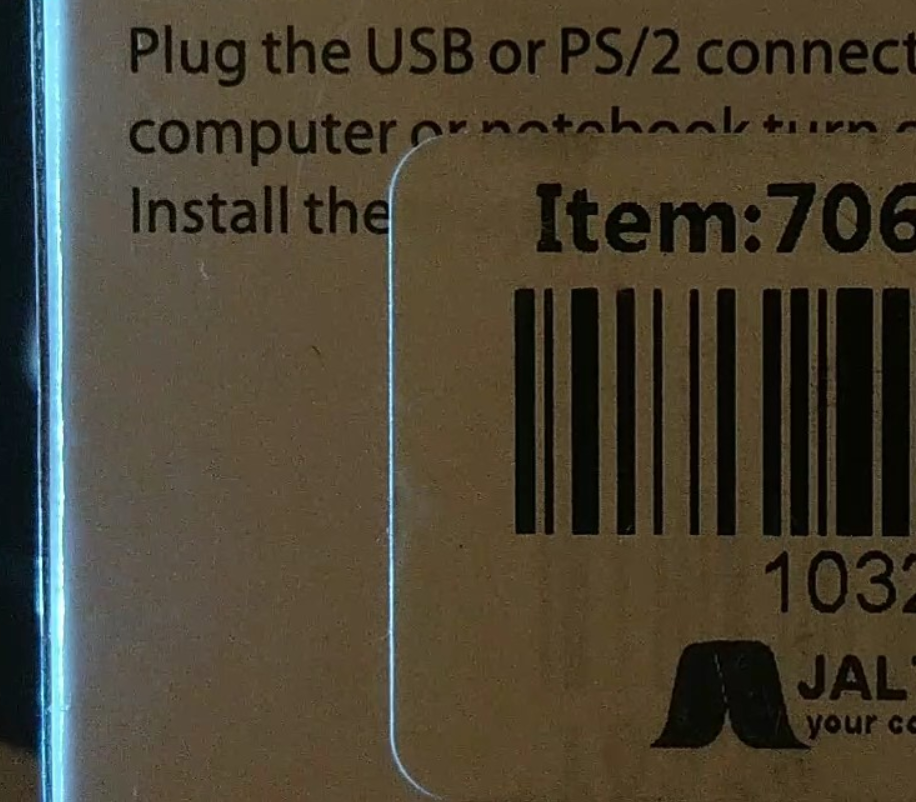


Fig. 13 Campo oscuro (recurso propio)

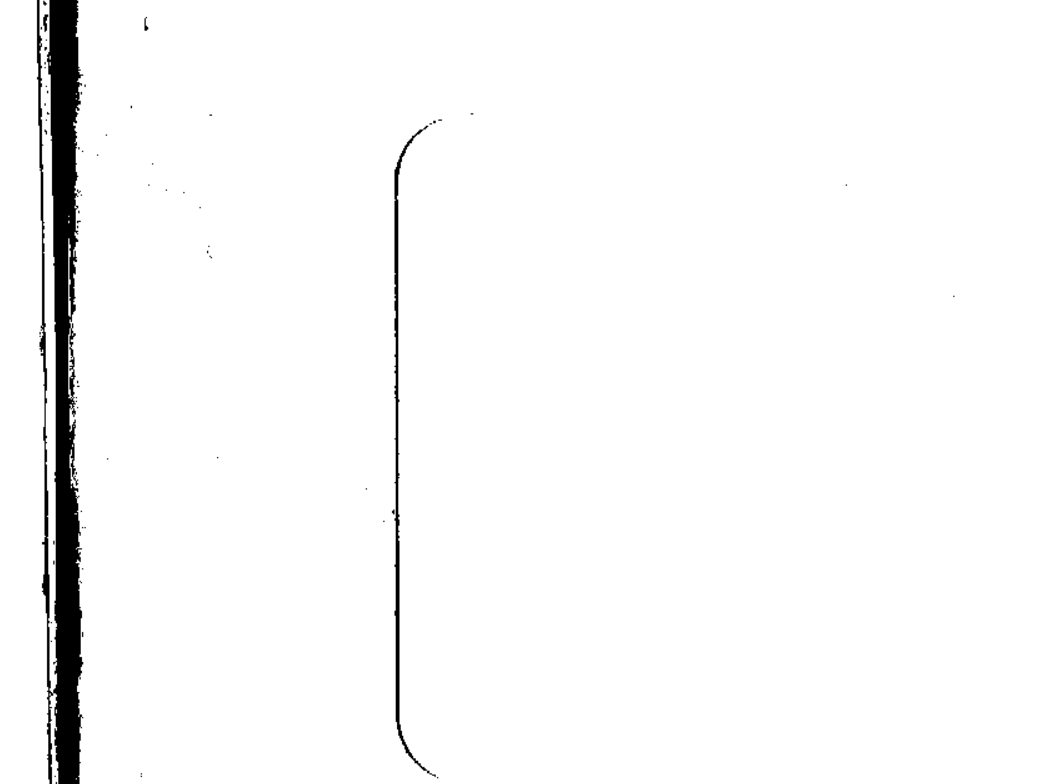


Fig. 14 Binarización Campo oscuro (recurso propio)

**Método Otsu**

El método de varianza máxima entre clases fue propuesto por el erudito japonés Otsu en 1979. Es un método de determinación de umbral adaptativo, también llamado método Otsu, o OTSU para abreviar. Es un algoritmo de binarización global, que se basa en la escala de grises de la imagen. Características del grado, la imagen se divide en dos partes: primer plano y fondo.

OTSU se utiliza para imágenes bimodales, y el histograma en escala de grises de la imagen que suele aplicarse para el umbral fijo es unimodal, el método otsu es muy útil ya que puede encontrar el umbral óptimo por sí mismo para imágenes bimodales

Ventajas: El algoritmo es simple y la imagen se puede segmentar de manera efectiva cuando el área del objetivo y el fondo no son muy diferentes.

Desventajas: cuando el área del objetivo y el fondo en la imagen es muy diferente, parece que el histograma no tiene picos dobles obvios, o el tamaño de los dos picos es muy diferente, el efecto de segmentación no es bueno o la escala de grises del objetivo y el fondo son relativamente diferentes Cuando hay una gran superposición, el objetivo y el fondo no se pueden separar con precisión.

El método Otsu fue utilizado en el programa medidor de barras, pero no se notaron diferencias grandes al respecto, esto puede ser porque se trabajó desde el principio con una umbralización adecuada para la tarea teniendo en cuenta el tipo de iluminación que se decidió utilizar.

**2.3 Algoritmo Findcontours**

El ejemplo seleccionado es un detector de movimiento, en este ejemplo lo que realizamos primero es llamar las librerías opencv y numpy luego empezamos la captura de video pasamos esto a escala de grises y con un contador vamos guardando los videos en escalas de grises para posteriormente restarlos a este resultado de la resta le aplicamos una binarización y aquí es donde utilizamos la función findcontours para hallar el contorno de la imagen binarizada hallamos el área de ese contorno y en base a eso dibujamos un rectángulo en color verde que cubra todo el contorno como se muestra en la siguiente imagen

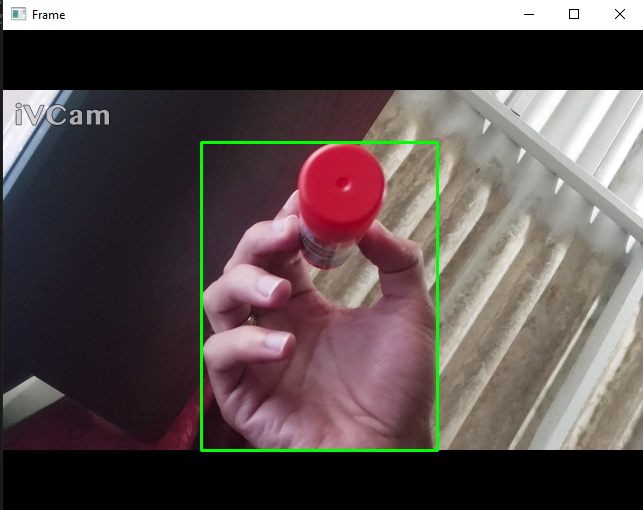


Fig.1 Ejemplo findcontours

**2.4 Segmentacion de color**

para la implementación del código se establecieron ciertos rangos para cada espacio de color para detectar los colores rojo, naranja, amarillo, verde, azul claro, azul más oscuro, violeta, luego se crea varias mascaras para la segmentación de cada uno de los colores y hacemos con estas mascaras que en el video en tiempo real en cada uno de los espacios de color mencionados anteriormente solo se tomen los 7 colores que escogimos

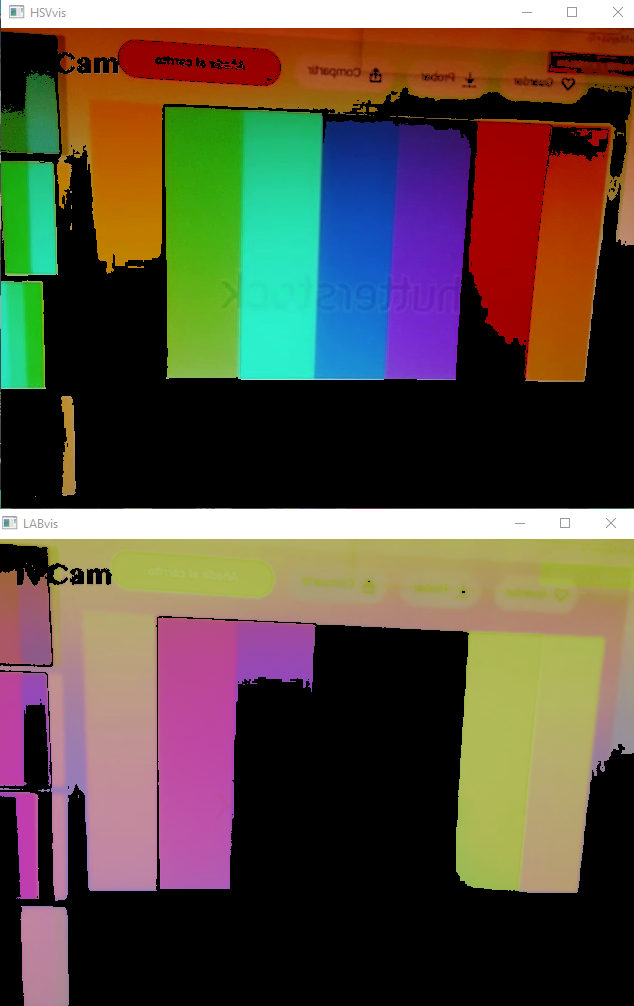
****

Fig.2 Segmentación HSV y CIELAB

**2.5 Detector de geometría**

Para la detección de la figura sin la función findcontours realmente fue algo complicado ya que era complejo poder hallar el contorno de la figura entonces decidimos utilizar la transformada de hougs esta función es utilizada para la detección de líneas y círculos la implementamos para detectar un círculo en video de tiempo real variando un poco sus parámetros logramos que no aparecieran demasiados círculos en la pantalla y lograr una buena detección del círculo en la siguiente imagen se ejemplifica la función

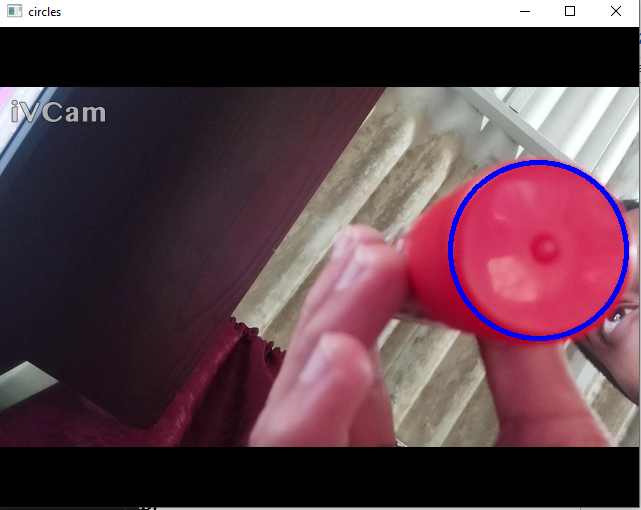


Fig.3 Detección de figura

**2.6 Operaciones Morfológicas**

Para la fig. Se utilizó la operación morfológica de Opening, con un kernel elíptico de 3x3 se consiguió el siguiente resultado.



Fig. Resultado (recurso propio)

la diferencia de píxeles blancos es inferior al 5% por lo que se cumplía con el requisito.

Para la fig. se utilizó nuevamente un kernel elíptico pero esta vez se hizo de tamaño 7x7 (fig. )para que la diferencia en cantidad de pixeles blancos no excediera el 5%, visualmente se ve peor que la imagen de referencia, esto se arregla poniendo un kernel de 9x9 (fig )pero la diferencia de pixeles es superior al 5%

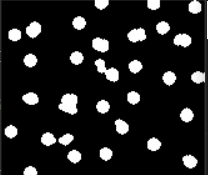


Fig. Kernel 7x7 (imagen propia)

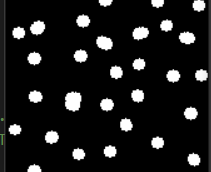


Fig. Kernel 9x9 (imagen propia)

En conclusión las operaciones morfológicas de Opening, son excelentes para eliminar las interferencias o puntos blancos que rodean una imagen, dependiendo del tipo de kernel pueden filtrarse formas enteras.

**3. Referencias**

[1] Sistemas de iluminación para aplicaciones de visión artificial [Online] Disponible en <https://www.bcnvision.es/blog-vision-artificial/iluminacion-vision-artificial2/>

[2] Las 6 mejores técnicas de iluminación para la visión industrial [Online] Disponible en: <https://www.ingelcom.com.ec/blog/blog-1/post/las-6-mejores-tecnicas-de-iluminacion-para-la-vision-industrial-17>

[3] ILUMINACIÓN DE VISIÓN ARTIFICIAL [Online] Disponible en: <https://www.cognex.com/es-mx/what-is/machine-vision/components/lighting>

[4] Image Thresholding [Online] Disponible en: <https://docs.opencv.org/4.x/d7/d4d/tutorial_py_thresholding.html>

[5] Detección de círculos con Python y Opencv [Online] Disponible en: <https://www.youtube.com/watch?v=L9kjv3Ixnv8>

[6] Detección de contornos con Opencv Python [Online] Disponible en: <http://acodigo.blogspot.com/2017/08/deteccion-de-contornos-con-opencv-python.html>

[7] Detección de colores [Online] Disponible en:

<https://omes-va.com/deteccion-de-colores/>