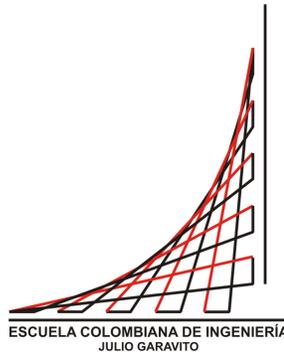


# ESCUELA COLOMBIANA DE INGENIERÍA JULIO GARAVITO

DECANATURA DE INGENIERÍA DE SISTEMAS



PROYECTO DE GRADO:

## MÓDULO DE VISIÓN ARTIFICIAL PARA UNA APLICACIÓN DE AGRICULTURA DE PRECISIÓN

Autores:

Guillermo León Álvarez Salamanca  
Juan Carlos Martínez Muñoz

---

Director:

Héctor Fabio Cadavid Rengifo  
Magister en Ingeniería de Sistemas 2016



# Índice general

<b>1. Problemática</b>	<b>5</b>
<b>2. Objetivos</b>	<b>7</b>
2.1. General . . . . .	7
2.2. Específicos . . . . .	7
<b>3. Justificación</b>	<b>9</b>
<b>4. Marco teórico</b>	<b>11</b>
4.1. Visión del computador . . . . .	11
4.2. Definición del color . . . . .	11
4.3. Modelo RGB . . . . .	12
4.4. Modelo CMY y CMYK . . . . .	13
4.5. Modelo HSI . . . . .	14
4.6. Modelo de Cámara Pin-hole . . . . .	14
4.7. Calibración de cámara . . . . .	17
4.8. Visión Estereoscópica . . . . .	18
4.9. Detección de objetos . . . . .	20
4.10. Corner detection . . . . .	24
4.11. FAST Algorithm for Corner Detection . . . . .	25
4.11.1. ORB Oriented FAST and Rotated BRIEF . . . . .	26
<b>5. Módulo de visión artificial</b>	<b>27</b>
5.1. Diagrama de comunicación . . . . .	27
5.2. Arquitectura . . . . .	28
5.3. Diagrama de clases . . . . .	30
5.4. Proyección futura . . . . .	31

<b>6. Pruebas y resultados</b>	<b>33</b>
6.1. Detección de Caras y Ojos . . . . .	33
6.2. Propiedades básicas de las imágenes . . . . .	33
6.3. Hallar contornos con OPENCV . . . . .	34
6.4. Suavizado de Imágenes . . . . .	35
6.5. Segmentación de Imágenes . . . . .	36
6.6. Calibración de la Cámara . . . . .	37
6.7. Detección por plantillas . . . . .	39
6.8. Detección por color . . . . .	42
6.9. Detección, correlación y selección de características y esquinas . . . . .	45
6.9.1. Harris Corner Detection . . . . .	46
6.9.2. Sub-Pixel Accuracy . . . . .	47
6.9.3. Shi-Tomasi Corner Detector . . . . .	48
6.9.4. SIFT (Scale-Invariant Feature Transform) . . . . .	48
6.9.5. SURF (Speeded-Up Robust Features) . . . . .	50
6.9.6. FAST (Features from Accelerated Segment Test) . . . . .	51
6.9.7. ORB(Oriented FAST and Rotated BRIEF) . . . . .	52
6.10. Detección según características y descriptores, análisis de información	53
6.10.1. SIFT (Scale-Invariant Feature Transform) . . . . .	53
6.10.2. SURF (Speeded-Up Robust Features) . . . . .	55
6.10.3. ORB(Oriented FAST and Rotated BRIEF)) . . . . .	56
6.11. Detección según características con filtro de color . . . . .	58
6.12. Estimación de distancias usando método de objeto conocido . . . . .	62
6.13. Rendimiento del módulo en la tarjeta Intel Galileo . . . . .	64

# Capítulo 1

## Problemática

En vista de la alta demanda de comida que se tiene, y el uso ineficiente de recursos para una producción que permita tener una oferta adecuada, se busca tener una tecnología que permita tener un mayor conocimiento del terreno de producción para así tomar mejores decisiones al respecto, por lo que se desea desarrollar un módulo de interpretación de video en tiempo real, con el cual se busca manejar ciertas funcionalidades requeridas para muchas tareas de agricultura de precisión, como es el reconocimiento de los distintos elementos del entorno, junto con la distancia relativa a los mismos.

La agricultura de precisión se puede definir como “la reunión de varias técnicas y tecnologías de producción agrícola y que tiende a una gestión inteligente de las parcelas partiendo de la base de que las condiciones del campo son altamente variables” según Santiago Franco, gerente de Negocio Agrícola y Energía de Eduardoño [1].

Como se mencionó previamente la agricultura de precisión hace uso de múltiples tecnologías para tener un conocimiento completo del entorno de producción, siendo una tecnología necesaria la visión artificial, ya que esta permite extraer información multimedia del medio para si poder reconocer las características visuales del mismo, permitiendo así tener una información más clara del estado del terreno, lo que significa poder mejorar la toma de decisiones.

No se desea demeritar los beneficios de las demás tecnologías aplicadas a la agricultura, solo se resalta la necesidad de la visión artificial en el tema de la agricultura de precisión.

Si bien la aplicación de la visión artificial en los temas de la agricultura de precisión es muy prometedora, todavía presenta algunos problemas en su aplicación, como son la obtención de las imágenes de muestra en caso de estar apoyado en imágenes aéreas y de satélite, o incluso en la dificultad para detectar las características buscadas debido a su reducido tamaño y baja resolución [2]. La variabilidad del terreno obliga muchas veces a dar soluciones específicas y rígidas. [3]

El caso de estudio en el que se aplica este proyecto, es una solución de automatización de agricultura de precisión que consta de un robot controlado a través de Plexil, el cual corre en una tarjeta SoC (System on a chip) Intel Galileo.

A continuación se muestra el marco teórico relacionado con las temáticas relacionadas al proyecto, en cual se tratarán los temas de visión por computador, color, cámara Pin-Hole, calibración de cámara, visión estereoscópica, detección de objetos, detección de esquinas y características.

# Capítulo 2

## Objetivos

### 2.1. General

Desarrollar un módulo de visión artificial para la toma de decisiones de un robot autónomo, el cual está diseñado para tareas de agricultura de precisión.

### 2.2. Especificos

- Modularización de la implementación de los algoritmos de detección de objetos y estimación de distancias.
- Implementación de comunicación entre procesos mediante memoria compartida.
- Pruebas de campo para detección de objetos usando ORB y color.
- Pruebas de campo para estimación de distancias usando un objeto conocido y un enfoque estereoscópico.



# Capítulo 3

## Justificación

El proyecto ‘evaluación de la tecnología Plexil para aplicaciones en el contexto nacional’, desarrollado conjuntamente por los programas de Ingeniería de Sistemas e Ingeniería Electrónica, requiere de un componente que permita a un robot autónomo (controlado por la plataforma experimental PLEXIL, desarrollada en la NASA), realizar tareas tales como la prevención de colisiones, identificación de objetos del entorno y el cálculo de las posiciones relativas de los mismos. Con el fin de asistir dichas tareas, este proyecto plantea el desarrollo de un módulo de visión artificial que haga uso de las mejores técnicas disponibles a la fecha.



# Capítulo 4

## Marco teorico

### 4.1. Visión del computador

Cuando se habla de la visión por computador se debe aclarar en primer lugar que dicho tema consiste en la obtención de información a partir de archivos multimedia como las imágenes, de forma en se puede entender como una forma de preparación del computador para reconocer las características del mundo real, por lo cual se deben conocer aspectos como la detección y seguimiento de objetos, manejo de profundidad o tridimensional en imágenes (estereoscopia), etc. [4]

### 4.2. Definición del color

El color es una percepción visual que se genera en el cerebro de los humanos y otros animales al interpretar las señales nerviosas que le envían los fotorreceptores en la retina del ojo, que a su vez interpretan y distinguen las distintas longitudes de onda que captan de la parte visible del espectro electromagnético. Todo cuerpo iluminado absorbe una parte de las ondas electromagnéticas y refleja las restantes. Las ondas reflejadas son captadas por el ojo e interpretadas en el cerebro como distintos colores según las longitudes de onda correspondientes. [1]

El espectro electromagnético abarca todas las longitudes de onda que la luz puede tener, aunque de todo el espectro, solo una pequeña porción es visible para el ser humano, denominándose esta región como espectro visible. Las longitudes que comprende el espectro visible van desde los 380 nm hasta los 780 nm. La luz de cada una de estas longitudes de onda es percibida en el cerebro humano como un color diferente. Figura 4.1. [1]

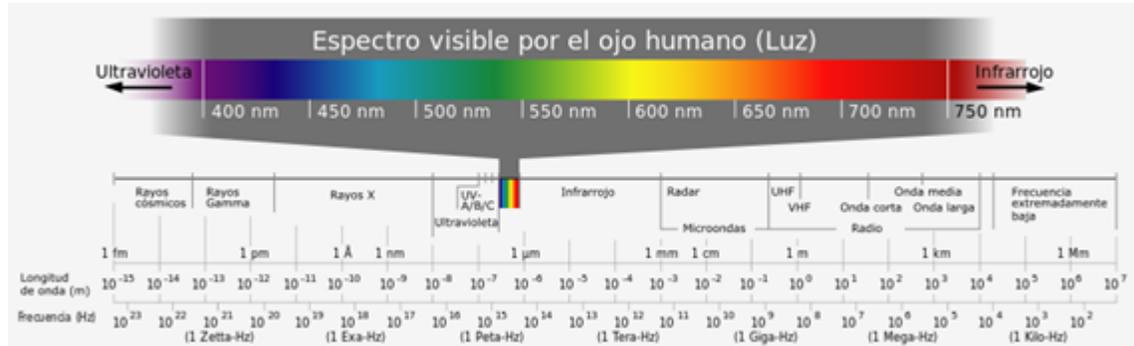


Figura 4.1: Espectro visible de la luz.

### 4.3. Modelo RGB

En este tipo de modelo, cada color aparece con sus componentes primarios espectrales de rojo, verde y azul. Se basa en el sistema cartesiano de coordenadas. Se muestra una imagen que ejemplifica esto:

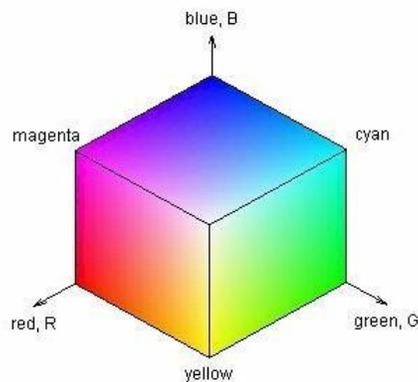


Figura 4.2: Esquema del cubo de colores RGB.

Los diferentes colores en este modelo son puntos que se encuentran sobre o en el interior del cubo, y son definidos por vectores que se extienden desde el origen. Ejemplo son los colores secundarios magenta, cian y amarillo que son otras esquinas

del cubo. [1]

Las imágenes que se representan en este modelo se componen de tres imágenes, una para cada una de los colores primarios. Estas tres imágenes se combinan para producir una imagen a color. Otro punto importante a resaltar es la cantidad de bits usadas para representar un píxel en un espacio RGB se denomina como la profundidad del píxel. Por ejemplo una imagen RGB en la cual cada una de las imágenes de azul, rojo y verde son una imagen de 8-bits, se denomina full-color.[1]

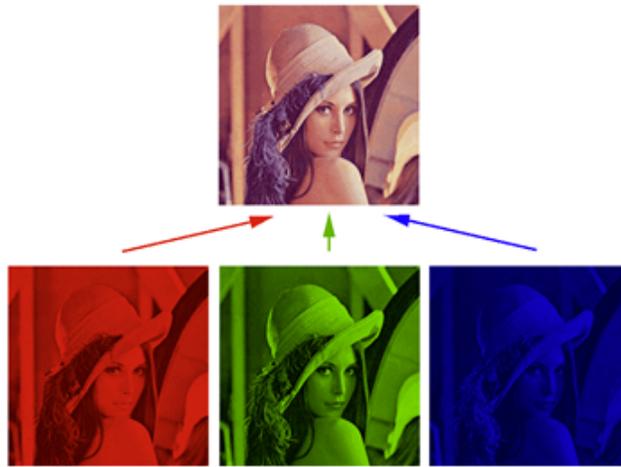


Figura 4.3: Los tres planos RGB que componen una imagen a color

## 4.4. Modelo CMY y CMYK

Cian, magenta y amarillo son los colores secundarios de la luz o, los colores primarios de los pigmentos. Un ejemplo es que cuando una superficie cian es iluminada con luz blanca, ninguna luz roja es reflejada por la superficie. Esto quiere decir, que el cian subtrae la luz roja de la luz blanca reflejada. [1]

Muchas máquinas como impresoras o fotocopiadoras necesitan la conversión de una entrada RGB a un modelo CMY para realizar su labor. Esta conversión se realiza por medio de la siguiente fórmula:

$$[C \ M \ Y] = [1 \ 1 \ 1] - [R \ G \ B]$$

De la ecuación se puede observar que una superficie con solo cian no contiene rojo,

una superficie puramente magenta no reflejará verde, y una superficie puramente amarilla no reflejará azul. Con esta combinación de colores no es posible de forma económica crear el negro, por esta razón se añade muchas veces este color dando como resultado el modelo CMYK. [1]

## 4.5. Modelo HSI

Los modelos RGB y CMY son excelentes para implementaciones de hardware, pero no dan describen los colores de una forma práctica para la interpretación humana. Un ejemplo es que las personas no se refieren al color de un automóvil dando un porcentaje de los tres colores primarios que componen a este, es más ni siquiera se piensa el color como una combinación de colores. [1]

Cuando un ser humano ve un color, este lo describe en términos de su saturación, matiz y brillo. El matiz es un atributo del color que describe un color puro (amarillo, tojo o naranja puro), la saturación da la medida del grado en que un color puro esta diluido por la luz blanco. El brillo es un descriptor subjetivo que es imposible de medir, pero se puede estimar por medio de la intensidad, esta es calculable usando un modelo de imágenes en escala de grises. [1]

EL modelo HSI (matiz, saturación, intensidad) desacopla la intensidad de la información portadora de color. Como resultado este modelo es una herramienta ideal para desarrollar algoritmos de procesamiento de imágenes basados en descriptores de color que son naturales e intuitivos para las personas. [1]

## 4.6. Modelo de Cámara Pin-hole

Es un modelo simple y preciso para la mayoría de aplicaciones en visión artificial, en este tipo de cámara, la luz pasa a través de un solo punto, denominado el centro de la cámara, antes de ser proyectado al plano de la imagen. La geometría de este tipo de cámaras se resume en la siguiente imagen.

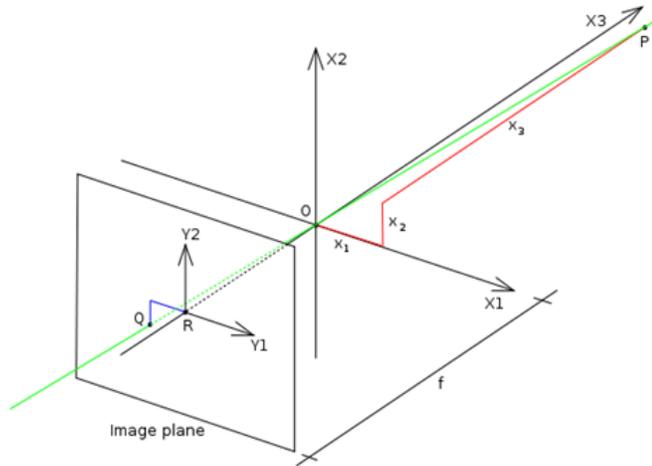


Figura 4.4: Geometría de una cámara pin-hole.

- Un sistema coordenado de 3 coordenadas con origen en el punto O. Este punto es donde se ubica la apertura de la cámara. El eje denominado X3 se llama rayo principal. Por otra parte el plano que intersecta los X1 y X2 es el frente de la cámara o plano principal.
- Un plano de la imagen del mundo en 3 dimensiones es proyectado a través de la apertura de la cámara. La imágenes es paralela a los ejes X1 y X2 y está localizada a una distancia f del origen O. Esta distancia f se denomina longitud focal.
- Un punto R es la intersección entre el rayo principal y el plano de la imagen se conoce como centro de la imagen.

El punto P tridimensional con coordenadas  $(x_1, x_2, x_3)$  se proyecta en el plano de la imagen como un punto Q. En este plano existe un sistema de coordenadas relativo con origen en R y ejes  $Y_1$  y  $Y_2$  que son paralelos a  $X_1$  y  $X_2$  respectivamente. Las coordenadas del punto Q relativas a este sistema de coordenadas son  $(y_1, y_2)$ .

Para ver la relación entre los puntos en el mundo real y los puntos en el sistema de coordenadas relativo, se utilizara las relaciones de triángulos semejantes que se pueden apreciar mejor en la siguiente imagen.

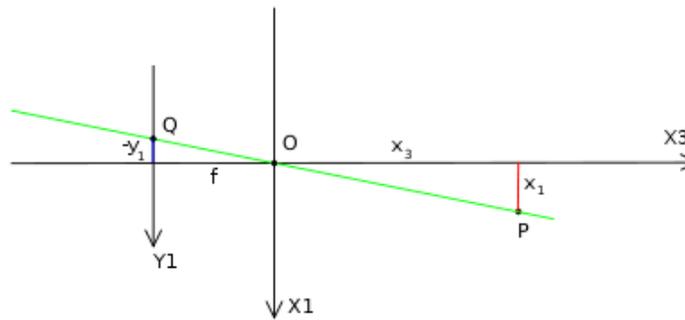


Figura 4.5: Geometría de una cámara pin-hole vista desde el eje X2

Viendo la relación de triángulos semejantes, se puede obtener la siguiente expresión:

$$y_1 = -\frac{f(x_1)}{x_3}$$

Con un análisis similar en el eje  $X_1$  se obtiene:

$$y_2 = -\frac{f(x_2)}{x_3}$$

Estas dos ecuaciones se pueden resumir en:

$$(y_1 y_2) = -\frac{f(x_1 x_2)}{x_3}$$

En coordenadas homogéneas la relación que acabamos de hallar se puede reescribir como:

$$y = Cx$$

Donde C se denomina la matriz de la cámara y es igual a:

$$[f \ 0 \ c_x \ 0 \ f \ c_y \ 0 \ 0 \ 0 \ 1]$$

En donde f sigue siendo la distancia focal y los otros dos valores, son las coordenadas del centro de la imagen. Todas estas características son intrínsecas de la cámara a trabajar.

## 4.7. Calibración de cámara

Calibrar una cámara consiste básicamente en determinar los parámetros internos de la cámara, en este caso particular es hallar la matriz de la cámara. La estrategia general para realizar la calibración se enuncia a continuación:

- Utilizar un objeto como plantilla de calibración, este puede ser un libro, tablero de ajedrez, etc.
- Identificar algunos puntos particulares del objeto en la imagen como pueden ser bordes o esquinas.
- Estimar la matriz de la cámara. Para este punto es necesario saber que objeto se usó o que aproximaciones se utilizarán.

Este proceso se realiza como un paso en el proceso de eliminación de la distorsión radial que las cámaras introducen a las imágenes. En la distorsión radial las líneas rectas tienen a curvarse, siendo más pronunciado cuanto más se aleja uno del centro de la imagen. Un ejemplo de esa distorsión se muestra en 4.6.



Figura 4.6: Distorsión radial.

## 4.8. Visión Estereoscópica

Dada una serie de imágenes bidimensionales es posible extraer una cantidad de información auxiliar acerca de la escena capturada. Una de las más útiles piezas de información es el conocimiento acerca de la profundidad relativa de los objetos en la imagen. Es sabido, que dadas dos imágenes de una sola escena es posible extraer la profundidad de varios objetos en la escena de la disparidad entre las dos imágenes. Se debe indicar que sin una cantidad significativa de información adicional y cálculos, no es posible asegurar una medida de la profundidad exacta. [5]

Para estimar la disparidad de las imágenes, tradicionalmente se usan dos imágenes referidas como imagen izquierda y derecha. Como las imágenes izquierda y derecha están viendo el plano desde diferentes lugares, habrá una notable disparidad entre las dos imágenes. EL punto vital para realizar esto es que puntos con niveles de profundidad similar en el mundo real tendrán niveles de disparidad similares entre las imágenes estereoscópicas. Esto quiere decir que dado un conjunto de puntos con sus respectivas correspondencias entre las imágenes derechas e izquierda, es posible determinar el mapa de profundidad de la escena. [5]

Para realizar el cálculo de estas correspondencias, la idea de un plano epipolar se torna fundamental. Esencialmente esta es una técnica para limitar el espacio de búsqueda cuando se está buscando la correspondencia de los puntos entre las dos imágenes. Sencillamente el plano epipolar es el plano formado por el punto en la escena y los centros de las cámaras en cada uno de los planos. Donde el plano interseca los dos planos de las imágenes se forma una línea epipolar en cada una de las imágenes. Es esta línea la que une la imagen del punto de la escena en cada una de las imágenes con el epipolo de las imágenes. Los epipolos son los puntos en cada una de las imágenes donde la línea entre los dos centros de las cámaras intersecan su plano de imagen. La siguiente imagen muestra un esquema de Figura 4.7

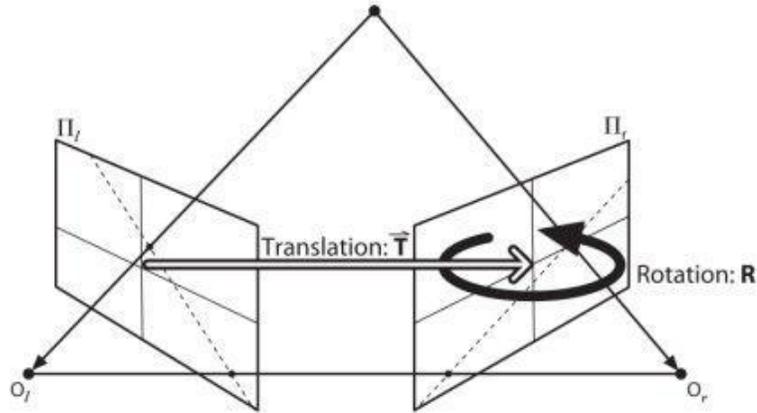


Figura 4.7: Esquema de geometría epipolar.

Todas las líneas epipolares pasan a través del epipolo, entonces para hallar la localización del epipolo, se puede hallar varias epilineas y hallar su punto de intersección. Para hallar esas líneas epipolares y epipolos se necesita hablar de la matriz fundamental (F) y la matriz Esencial (E). La matriz esencial contiene la información acerca de la traslación y rotación, la cual describe la localización de la segunda cámara relativa a la primera en coordenadas globales.

Para realizar el cálculo de la matriz fundamental se usa el algoritmo de los ocho puntos. La relación entre los puntos de una imagen y la otra es equivalente a:

$$x_2^T F x_1 = 0$$

Esta ecuación puede reescribirse como:

$$\begin{bmatrix} x_2^1 x_1^1 & x_2^1 y_1^1 & x_2^1 w_1^1 & \dots & w_2^1 w_1^1 \\ x_2^2 x_1^2 & x_2^2 y_1^2 & x_2^2 w_1^2 & \dots & w_2^2 w_1^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_2^n x_1^n & x_2^n y_1^n & x_2^n w_1^n & \dots & w_2^n w_1^n \end{bmatrix} \begin{bmatrix} F_{11} \\ F_{12} \\ \vdots \\ F_{33} \end{bmatrix} = A f = 0$$

Donde  $f$  contienen los elementos de  $F$ ,  $x_1^i = [x_1^i y_1^i w_1^i]$  y  $x_2^i = [x_2^i y_2^i w_2^i]$  son parejas de correspondencia y son  $n$  puntos de correspondencia en total. La matriz fundamental tiene en total 9 elementos, pero como la escala es arbitrario, solo son necesarias ocho ecuaciones. Ocho puntos de correspondencia son necesarios para computar  $F$ , de aquí proviene el nombre del algoritmo.

Para estimar la densidad de profundidad utilizando un enfoque estereoscópico se utiliza la triangulación como enfoque, la imagen Figura 4.8 resume la intuición

detrás de esta metodología. [6]

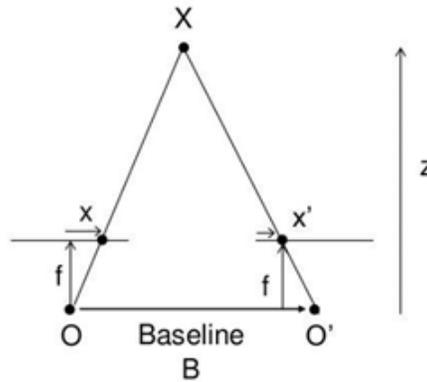


Figura 4.8: Esquema de visión estereoscópica con cámaras paralelas.

Como se aprecia de la relación de los triángulos, se puede calcular la disparidad como:

$$disparidad = x - x' = \frac{Bf}{Z}$$

Donde B es la distancia entre las dos cámaras y f es la distancia focal de la cámara. Además, x y x' es la distancia entre los puntos en el plano de la imagen correspondientes al punto de la escena en 3D y los centros de la cámara.

## 4.9. Detección de objetos

La detección de objetos tiene como objetivo principal, lograr seguir la ubicación un mismo objeto en múltiples imágenes, sin importar si cambió su tamaño o posición, de manera que se han definido múltiples algoritmos para el cumplimiento de esta tarea, siendo necesario en un inicio la implementación de filtros para la detección de objetos, los cuales aplican diferentes técnicas como redes neuronales por entrenamiento, texturas, detección de esquinas, uso de descriptores, etc. Aquellos filtros que hacen uso de redes neuronales, usan muestras previas del objeto que se desea buscar. [7]

Al momento de formular un algoritmo que permita la detección de objetos se debe tener en cuenta aspectos como ruido en la imagen, obstrucciones parciales o

totales, cambios de iluminación, requerimientos de procesamiento en tiempo real, articulación del objeto, etc., el tratamiento para dichos objetos se puede simplificar teniendo información previa, como información del escenario y velocidad movimiento del objeto a seguir. [8] [9]

La primera tarea que se debe hacer para la detección de objetos, es definir la descripción de los mismos, para lo cual se pueden determinar por puntos específicos que caracterizan al objeto, la forma geométrica del mismo, la articulación del mismo mediante la definición propia de ciertas partes, como las extremidades de un ser vivo, también se encuentra el contorno y silueta del mismo o el cambio de forma de la silueta; este último aspecto ha tenido múltiples aplicaciones de detección como plantillas (integra formas geométricas) y modelo multivista. [8]

Luego de haber conocido la definición de características, se procede a la selección de estas, en este paso se observa que la distribución de colores comúnmente usada es RGB, la cual hace una aproximación usando los colores rojo, azul y verde; no obstante, este formato no es recomendable, debido a que no asimila correctamente la distribución de colores aproximada del objeto y el brillo que presente, por lo cual se hace uso del formato HSV, el cual maneja tono, valor y saturación, permitiendo el manejo de una coloración más acertada.

En la detección de objetos se hacen uso de múltiples técnicas como detección basada en características, texturas, kernel, flujo óptico, contexto, uso de histogramas, detección de múltiples hipótesis, las cuales muchas veces se combinan con segmentación de las imágenes, siguiendo así una orientación al uso de filtros; aunque no se deben olvidar aquellas detecciones que hacen uso de sistemas de gaussianos y lineales.[10]

El uso de la forma geométrica permite disminuir la cantidad de información a procesar y la identificación en caso de que se presente oclusión parcial del objeto, variación de la escala del objeto, cambios en el escenario imprevistos implicando la definición de nuevos puntos clave o características a seguir.[11]

La segmentación presta una gran ayuda en la detección de objetos, debido a que permite acelerar el reconocimiento de objetos, ya que al momento de buscar, se tendrá un área muy grande a evaluar y posiblemente solo se pueda reconocer aquellos objetos que se presentan con el mismo tamaño de la muestra, o por el contrario se presenten falsos positivos con objetos similares al buscado, siendo así necesario la definición de una región de interés.

En una región de interés se puede evaluar pequeñas características que probablemente se hubiesen descartado en una situación normal, ya sea por la poca o mala información con la que se disponga.

En el año 2010 [12] se publicó un método de detección basado en el contexto del objeto, lográndose definir mejor las relaciones que hay entre el objeto en detección, y la imagen en la que se encuentra, aunque la orientación fue dada a la aplicación del “Implicit Shape Model” (ISM), el cual una metodología de categorización de objetos determinada por la coherencia entre aspectos seleccionados de cada objeto.

Este método [12] define como primer paso de extracción de las características propias de la información mostrada en la imagen, las cuales se distinguen entre aquellas que son propias del objeto y aquellas presentes en la superficie del objeto que se utilizan para determinar la posición del mismo, por lo que se les llamó “Supporters”.

Como dichos “Supporters” [12] están relacionados con el movimiento del objeto en detección, se hace uso de “Generalized Hough Transform” (GHT), el cual es una técnica para la actualización constante de dichas características, la cual hace innecesario un fondo estático y permite la utilización de múltiples “Supporters” al mismo tiempo.

Como ya se había mencionado anteriormente, en la detección de objetos, se debe tener en cuenta como primera tarea manejar una forma de describir el objeto a seguir; una de muchas formas es usando el color representativo o distintivo del objeto a seguir, para que así la idea sea detectar las distintas concentraciones de color que se puedan llegar a encontrar en las partes de una imagen, de manera que encontrando el espacio en el cual se encuentra la mayor concentración se puede interpretar como que en dicho lugar se encuentra capturado el objetivo.

Para la detección por color se necesita evaluar cada uno de los píxeles que contiene la imagen, estos píxeles manejan la información de su ubicación y características de los mismos, como el color. Para esto, se hace uso de máscaras de color, filtros de erosión y filtros de dilatación, con los cuales se determina patrones de color similares al buscado para luego definir su ubicación y concentración, claro que se debe manejar un valor de concentración mínima, como medida para evitar falsas ubicaciones de posibles ocurrencias.

Una forma alternativa de detección de objetos es el uso de una plantilla, para

esto se requiere del uso de clasificadores de detección, los cuales permiten evaluar aspectos como variación en la iluminación, orientación, ruido, etc. Estos clasificadores basan su comparación en imágenes a escala de grises, para luego comparar la plantilla en distintas partes de la imagen, de manera que se pueda buscar y ubicar el objetivo en distintas partes de la imagen. Esto no necesariamente quiere decir que las múltiples comparaciones se deban hacer secuencialmente, ni buscando un solo objetivo, ya que se pueden hacer las comparaciones en paralelo para la aceleración de la búsqueda del objetivo, o en caso de que se puedan encontrar múltiples objetos similares, los reconozca a todos.

Otra forma de detección de objetos es el uso de bordes, puntos de interés, los cuales se puede definir como pequeñas secciones, en las cuales se presenta un cambio de intensidad de color diferente al campo circundante de la imagen en la que se encuentre este borde, estas características se reproducen en la imagen del objeto a seguir para luego compararlas con aquellas características obtenidas de los distintos frames que se esté generando en el video.

Aunque en algunos métodos de detección por bordes, se hace una comparación en ambos sentidos para disminuir la probabilidad de encontrar falsas ocurrencias, estas características permiten que se pueda reconocer el objeto en distintas posiciones, y con variación en el tamaño del objeto (acercamiento o alejamiento frente a la cámara); además, en algunos casos se manejan la orientación de las características, para un mejor manejo de los criterios de aceptación de puntos clave.

Las características mencionadas anteriormente se pueden distribuir en dos grupos, donde el primer grupo comprende las características generales, es decir, aquellas que permiten describir el objeto en conjunto, obteniendo un buen desempeño en la detección de objetos cuando varía el tamaño del objeto dentro de la imagen, y el segundo grupo está conformado por aquellas características locales que describen las partes del objeto a seguir, estando de esta manera orientadas al reconocimiento desde distintos puntos de vista, mediante la definición y descripción de regiones de interés dentro de la imagen.

En los últimos años se ha desarrollado múltiples algoritmos para la detección de objetos, orientados a distintos aspectos como velocidad, uso de recursos, entre otros; estos algoritmos evalúan la detección desde diferentes puntos de vista como concentración del color del posible objeto a seguir, comparación simétrica frente a una plantilla y uso de descriptores, así mismo estos están orientados a diferentes aspectos, lo cual indica que uno no es inferior a otro, sino que no se están aplicando

en el campo adecuado.[10]

No obstante, aquellos algoritmos que realizan su tarea desde el mismo punto de vista, se diferencian de varias formas, como los criterios de selección de descriptores que manejan, velocidad de procesamiento de imagen, eficiencia, como es el caso de SIFT, SURF, FAST y ORB, los cuales son algoritmos diseñados para la detección de objetos en función de descriptores propios de cada uno.[8]

En el año 2006, se presentó un estudio en el cual su objetivo era presentar una nuevo detector de características, más eficiente que los del momento, el primer criterio para su definición fue la velocidad de procesamiento, sin que esto representará un costo en el consumo de recursos; el primer resultado fue FAST (Features from Accelerated Segment Test). El cual se evalúan secciones circulares de 16 píxeles con una posible esquina para ser considerada como un punto clave. [13]

## 4.10. Corner detection

Un aspecto primordial para el reconocimiento de objetos es el manejo de la detección de esquinas, entendidas como áreas en los cuales se evalúan diferencias en los cambios de color de los píxeles abordados, estos cambios en forma individual se llaman bordes, los cuales surgen por discontinuidad en la forma del objeto que al capturarse en un frame genera un cambio en los colores o en la intensidad de alguno.

Cuando se empieza a encadenar bordes se presentan curvaturas, las cuales permiten su cuantificación fácilmente, permitiendo así su reconocimiento eficientemente, además, en las curvas se manejan ángulos; las esquinas son aquellas en la cuales se presenta la mínima curvatura, ya que en estos puntos se presentan generalmente múltiples intersecciones de otras curvas.[13]

Como ningún objeto se puede reconocer por una única característica, se vuelve necesario desarrollar algoritmos que permitan manejar múltiples características en simultáneo; de manera que se pueda reconocer un objetivo por varios aspectos; en este punto es donde se presenta el problema de su rendimiento en el procesamiento de video en tiempo real, siendo necesario bajar la complejidad sin sacrificar la calidad del detector, y una forma es la de generalizar la detección, es decir, como la velocidad a la que se generan los frames del video permite la captura de entornos similares, se requiere formular algoritmos que puedan detectar estas situaciones y

las tomen como repeticiones de frames previamente evaluados.[13]

## 4.11. FAST Algorithm for Corner Detection

Algoritmo de detección de características de alta velocidad propuesto por Edward Rosten and Tom Drummond, en el 2006 en su documento “Machine learning for high-speed corner detection” [13], el cual engloba los algoritmos de detección de características que se habían desarrollado hasta el momento, cabiendo aclarar que los primeros pasos dados para la detección, seguimiento y localización, se basaron en detectores de esquinas, de modo que se presenta en dicho documento una evolución de los modelos del momento, manteniendo siempre una buena calidad en los descriptores, sin que esto implique una mayor carga de procesamiento.

Los puntos clave o descriptores que maneja éste algoritmo, consiste en conjuntos de puntos los cuales tienen una posición contigua a los demás del conjunto, formando un círculo, en cuyo centro se encuentra el píxel aspirante a consideración de esquina, el cual se verifica comprobando que los píxeles circundantes a éste tengan una mayor diferencia de intensidad de brillo, mirándolo a nivel de escala de grises. Este principio está basado en el algoritmo detección de esquinas SUSAN, en el cual se define un área circular para evaluación, además, SUSAN se basa a su vez en la definición del “Bresenham’s circle” [14], claro que en el algoritmo SUSAN se evalúan todos los píxeles seleccionados, pero en algoritmo FAST, se evalúan sólo una parte de estos, aquellos que presenten la mayor diferencia de brillo.

Para una evaluación rápida de esquinas se toman normalmente todos los 16 píxeles, pero en el caso de FAST se toman 9; estos números indican la cantidad de ángulos que se están evaluando para la verificación contra el píxel central. El criterio completo de evaluación incluye el aseguramiento de un mínimo de píxeles que presenten una determinada diferencia en su umbral de brillo, respecto al píxel central, a esto se le agrega que se maneja un algoritmo de verificación de patrones para facilitar selección de esquinas, ya que la evaluación continua de cada uno de los píxeles en una imagen sería demasiado costoso; además, se maneja una máquina de aprendizaje en la que se usan árboles de decisión, sobre tres aspectos: más brillante, más oscuro y similar.

Durante la selección de esquinas se puede presentar ubicaciones continuas de esquinas, por lo cual FAST, presenta una herramienta para tratar éste problema,

llamada “Non-maximal suppression”, en la cual se evalúa la adyacencia entre esquinas, para así eliminar esquinas similares.[15]

#### 4.11.1. ORB Oriented FAST and Rotated BRIEF

Algoritmo de detección de características binarias, basado en BRIEF, con mejoras como la invariancia a la rotación, resistencia al ruido y orientación exacta, sin que estos cambios impliquen disminución en la calidad del algoritmo original, ya que ha demostrado buenas capacidades para el reconocimiento de objetos, claro que teniendo un gran costo computacional, lo cual lo hace inviable para un dispositivo móvil o de bajas capacidades; el rendimiento logrado permite su aplicación en tiempo real; se aclara que este algoritmo no maneja la licencia de FAST y BRIEF, lo cual permite su uso libre para investigación y afines.[16]

Fue desarrollado en los laboratorios de OpenCV y presentado por Ethan Rublee, Vincent Rabaud, Kurt Konolige y Gary R. Bradski en su artículo titulado “ORB: An efficient alternative to SIFT or SURF” en el año 2011 [17]; integra la definición de puntos clave manejada en FAST, para el escalamiento se apoya en la medida de esquinas de Harris.

En el tema de rotación se presenta un problema y es que FAST no computa orientación, de manera que se manejó como orientación el vector generado entre el centro de la esquina y el ancho del punto clave manejado; para los descriptores hace uso de BRIEF, pero para corregir su falencia en la rotación, se genera una matriz que cubra los píxeles tomados para el punto clave y se rota para así conocer la versión rotada de la característica; BRIEF maneja una varianza en la definición de las características y al usar la corrección de orientación esto se pierde lo que vuelve a los descriptores más selectivos.

# Capítulo 5

## Módulo de visión artificial

### 5.1. Diagrama de comunicación

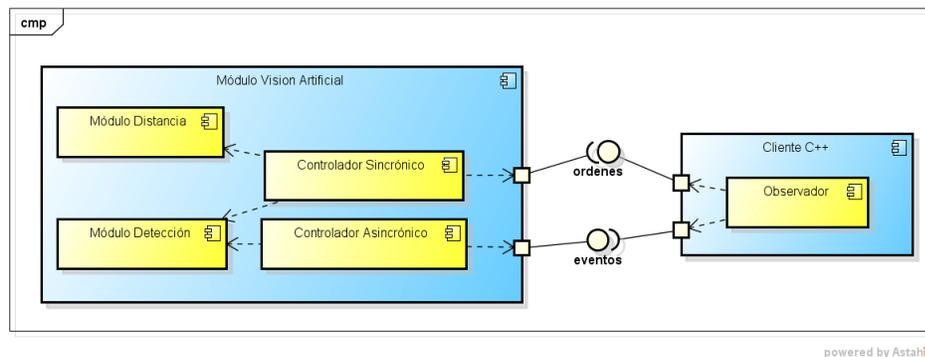


Figura 5.1: Diagrama de comunicación cliente-módulo

Al manejarse un cliente *c/c++*, éste cumple con la tarea principal de ser un observador de eventos y ordenes, de manera que si el módulo en alguna de sus funcionalidades de detección de objetos o estimación de distancias genera alguna información de algún suceso, el cliente estará en la capacidad interpretar estos eventos, para así comunicárselo al robot, para poder tomar acciones al respecto. Por el contrario, si es el robot aquel que requiere alguna información que le puede brindar el módulo, el cliente generara y enviara una orden, la cual el módulo tratará y dará respuesta al respecto. (Figura 5.1)

Se debe aclarar que se deben manejar sucesos síncronos y asíncronos, aquellos hechos síncronos se están refiriendo a las órdenes dadas por el robot ya que cada

orden que va recibiendo el módulo, se acumula en una cola y el módulo las va respondiendo inmediatamente, en cambio, aquellos hechos asíncronos, nacen en el módulo y se envían al cliente en cualquier momento y sin que este haya solicitado alguna acción. (Figura 5.1)

## 5.2. Arquitectura

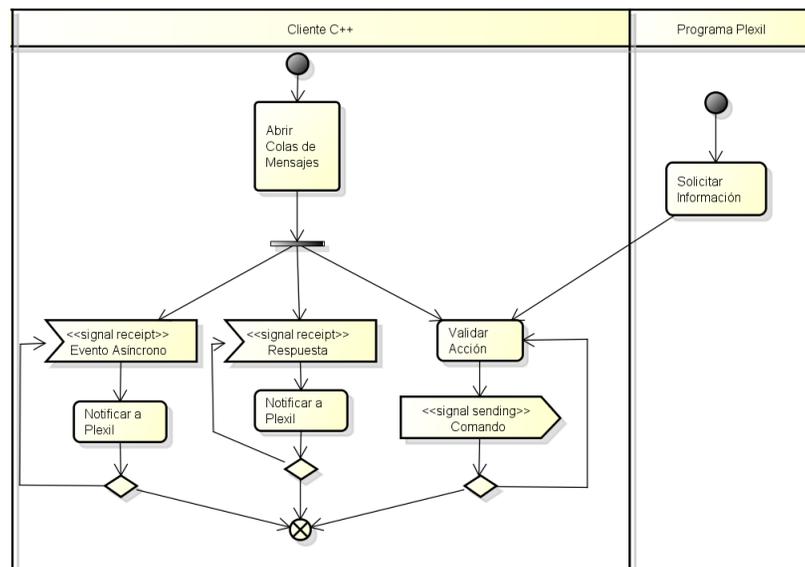


Figura 5.2: Diagrama de arquitectura del módulo

Una vez inicializado el módulo de visión artificial, se procede a cargar los módulos que éste tenga; luego, en paralelo, se verifica por un lado la recepción de alguna orden por parte de cliente, y por otro la generación de algún evento que implique una tarea asíncrona. De presentarse el primer caso se procede a ejecutar la funcionalidad requerida y luego se envía la respuesta al cliente para finalmente volver a esperar una nueva orden; en caso de presentarse el segundo caso, una vez verificado el evento, se procede a enviar la información recopilada del mismo, para luego nuevamente verificar nuevos eventos. (Figura 5.2)

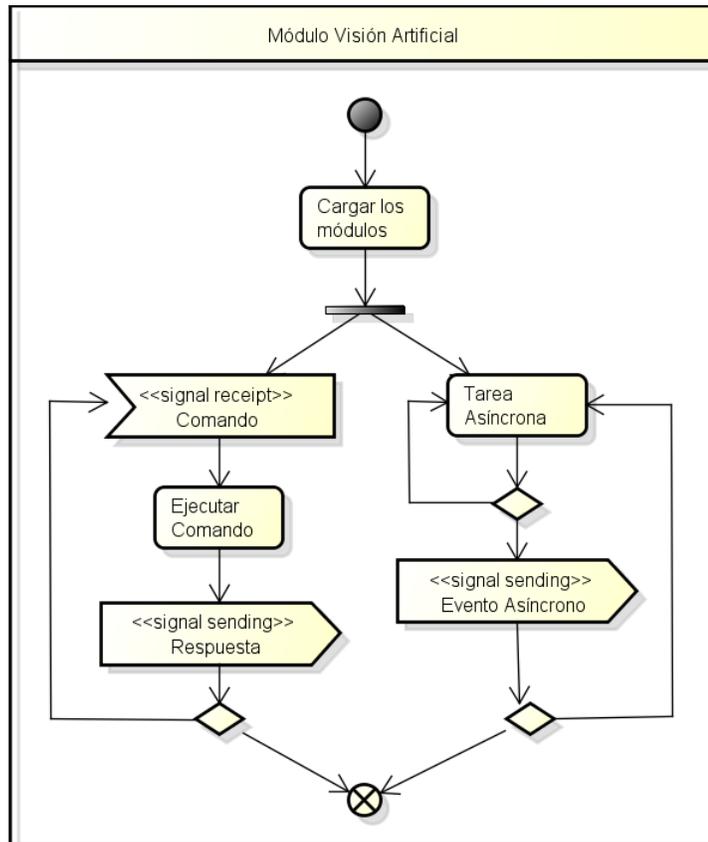


Figura 5.3: Diagrama de arquitectura cliente-Plexil

Una vez inicializado el cliente c/c++, se procede a abrir las colas de almacenamiento compartido (tipo POSIX), tanto para el módulo como para éste mismo, para luego realizar en paralelo la recepción de mensajes asíncronos del módulo, respuestas a peticiones dadas al módulo y la validación de órdenes dadas por el programa Plexil para el módulo. En caso de presentarse los primeros dos sucesos de recepción de respuesta, se procederá a notificar a Plexil, para que éste tome acciones al respecto y luego seguir esperando por nuevos sucesos, en el caso restante se valida la petición dada por Plexil, para luego notificar al módulo sobre la orden solicitada. De manera que se presenta un cliente para solicitar información de forma síncrona o recibir eventos de forma asíncrona del módulo de visión artificial, junto con un puente de comunicación del entorno PLEXIL con los eventos generados por software del módulo python. (Figura 5.3) (Figura 5.4)

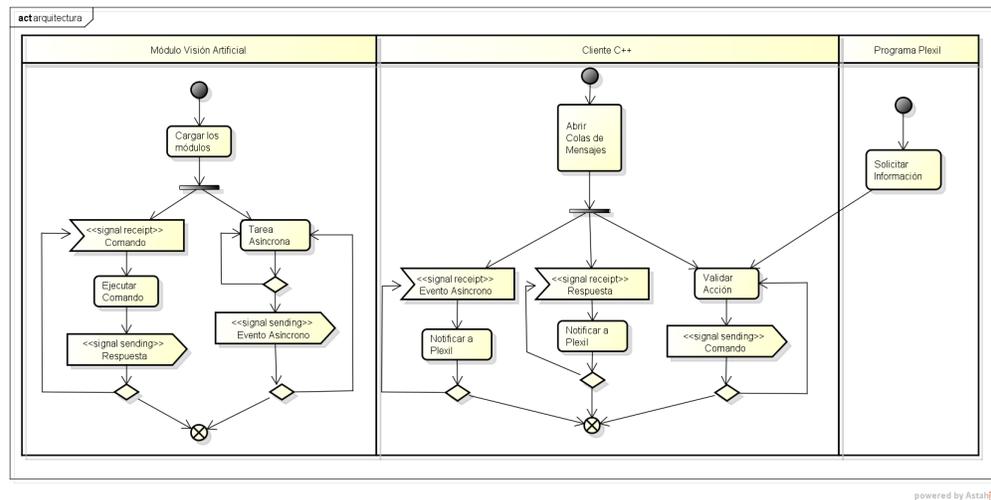


Figura 5.4: Diagrama de arquitectura

### 5.3. Diagrama de clases

El módulo cuenta con un controlador sobre las funcionalidades de detección de objetos y estimación de distancias, claro que éste primero también está acompañado por una herramienta de generación de eventos; como la idea es que el procesamiento del video se haga en forma transparente, se hace uso de interfaces, que enuncian ciertas funcionalidades a implementar por parte de cualquier clase que las implemente.

Las funcionalidades en este caso son distancia al objeto y distancia a un punto específico de una imagen del video por parte de la interfaz de distancia y esta el objeto para la interfaz de detección, que para estos casos se tienen las implementaciones de visión estereoscópica, objeto conocido, detección por color y por características (SURF, ORB).

Claro que como el módulo puede mantener en ejecución una de las tareas de la funcionalidad de detección en forma permanente, de manera que cuando ésta genere alguna información se procederá a definir y enviar el evento el cliente c++.

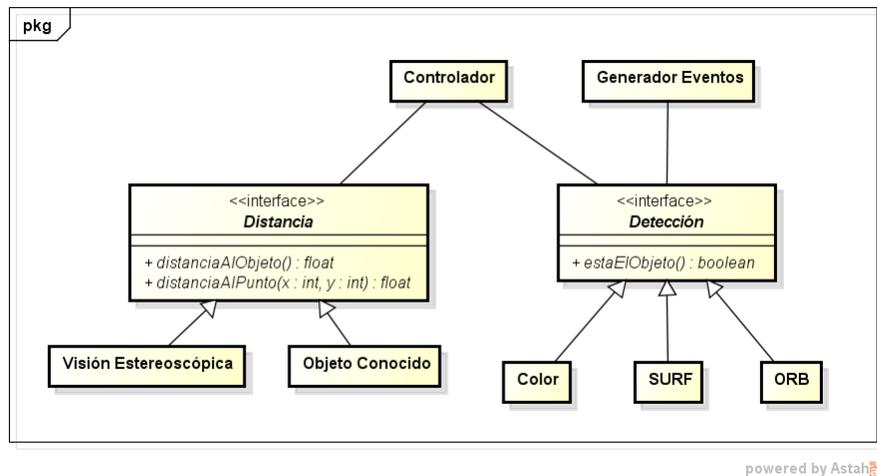


Figura 5.5: Diagrama de clases del módulo

## 5.4. Proyección futura

En vista de las bajas capacidades de la tarjeta Galileo en uso se podría esperar su aplicación como una tarjeta dedica a captura y envío de video por streaming, de manera que se pueda hacer el procesamiento del video en forma remota, con una maquina más potente; una segunda aplicación futura podría ser el refinamiento de los algoritmos de detección y estimación de objetos y distancias respectivamente; además de la integración de nuevos algoritmos.



# Capítulo 6

## Pruebas y resultados

### 6.1. Detección de Caras y Ojos

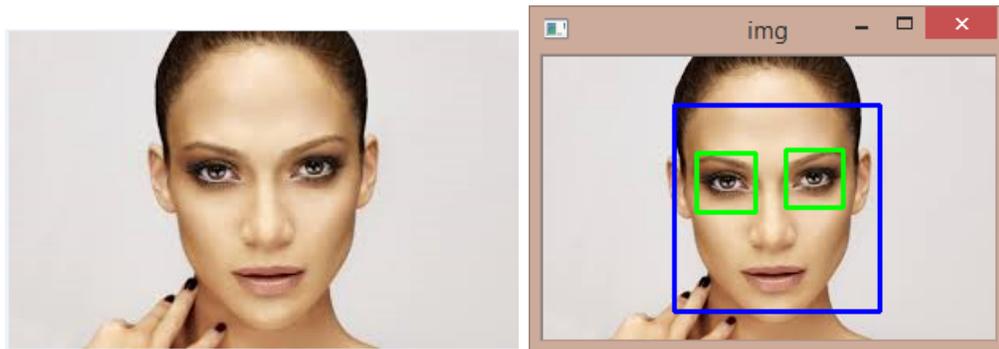


Figura 6.1: Imagen Original, imagen procesada

### 6.2. Propiedades básicas de las imágenes

Con el uso de OpenCv se puede realizar de forma sencilla cambios entre los diferentes modelos de color como los son RGB, HSV, escala de grises y demás modelos. Un ejemplo es poder extraer de una imagen a color los tres planos RGB que la componen. Se muestra el resultado de realizar esta operación de división en OpenCV en (Figura 6.2).

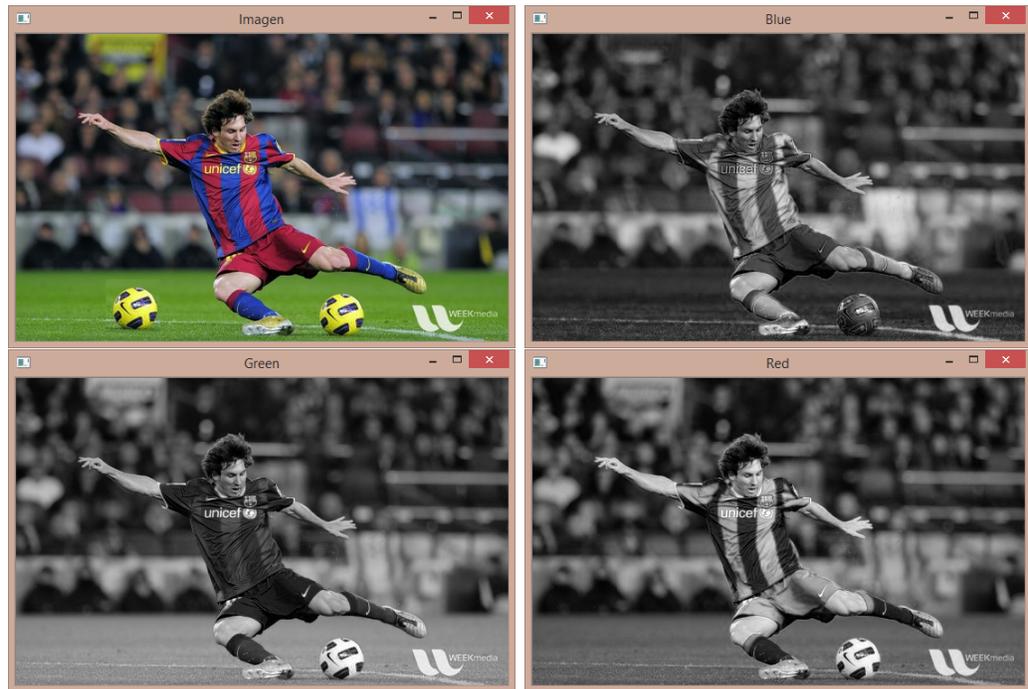


Figura 6.2: Imagen Original, plano azul, plano verde, plano rojo

### 6.3. Hallar contornos con OPENCV

Uno de las funcionalidades más importantes para la detección de objetos en el procesamiento de videos, es el poder detectar los contornos de las figuras que componen la imagen. Para esto se realiza un filtro pasa altas que deja pasar los cambios fuertes de frecuencia en la imagen, siendo estos cambios fuertes observables en los contornos de las figuras. Se presenta a continuación el filtro sobre que nos brinda openCV y los resultados arrojados (Figura 6.3).



Figura 6.3: Contornos

Con el filtro de sobre original solo se detectan los contornos de entrada a los cuadrados de la figura, es por esta razón que se usa una variación que permite detectar tanto los contornos de la derecha como los de la izquierda de la imagen.

## 6.4. Suavizado de Imágenes

En muchas ocasiones se tiene que las imágenes presentan algún tipo de ruido, el cual puede ser de sal y pimienta, gaussiano o de otra índole. Realizar el procesamiento de las imágenes en estas circunstancias puede llegar a producir resultados equivocados por la presencia de estos datos corruptos. OpenCv cuenta con varios filtros predefinidos que se pueden utilizar con el objetivo de disminuir el impacto del ruido en nuestro análisis. Destaca el filtro Gaussiano, de medio, promedio y bilateral. Lo que buscan este tipo de filtros es realizar la convolución de un filtro sobre la imagen. El filtro es una matriz cuadrada típicamente de dimensiones de  $3 \times 3$  que busca suavizar la imagen. Esto lo logra ponderando los píxeles de la vecindad de un valor y dando un valor que está relacionado con sus vecinos, por ende, si existe un píxel con valores atípicos, los demás vecinos tenderán a disminuir su impacto. Se muestra las pruebas de estos filtros a una imagen con alto contenido de ruido en (Figura 6.4).

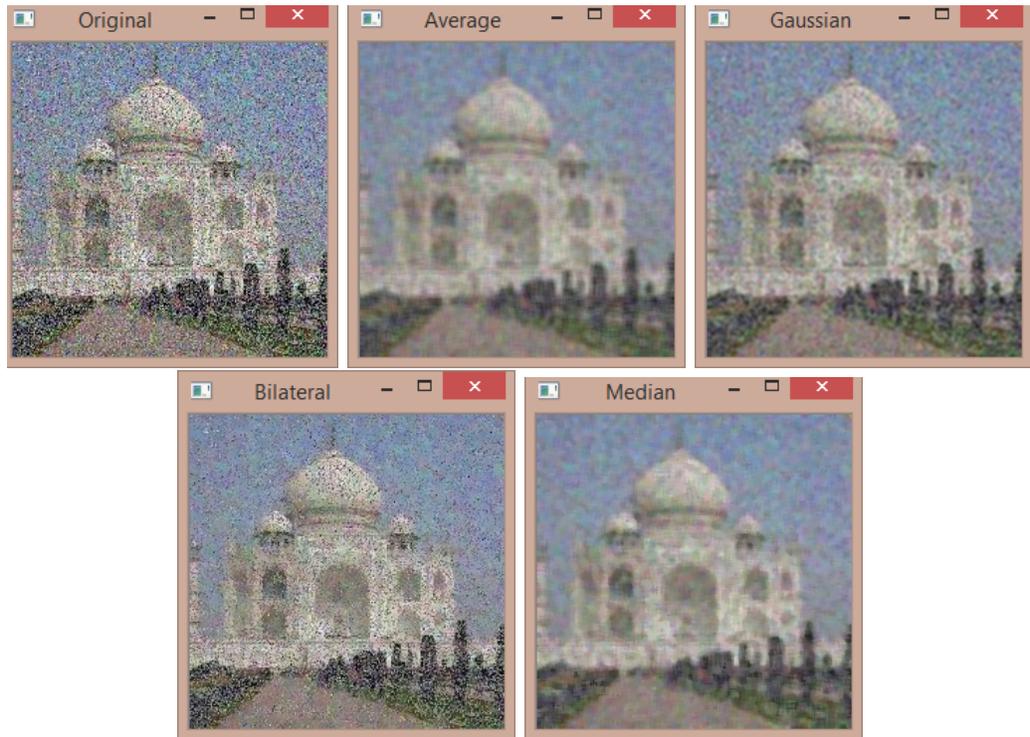


Figura 6.4: El filtro que se deba utilizar en la aplicación debe ser identificado en la práctica

## 6.5. Segmentación de Imágenes

La segmentación de imágenes consiste en el proceso de dividir una imagen digital en múltiples segmentos (conjuntos de píxeles). La meta de la segmentación es simplificar o cambiar la representación de la imagen en algo con más sentido y más sencillo de analizar. La segmentación de imágenes se utiliza básicamente para detectar fronteras (líneas, curvas, etc.) en imágenes.

Uno de los métodos más sencillos para realizar este tipo de segmentaciones es uso de umbrales para clasificar un píxel como blanco o negro dependiendo de si su valor es mayor o menor a determinado umbral. Esto genera una imagen binaria que puede ser sometida a otro tipo de técnicas como transformaciones morfológicas. Se muestra el uso de diversas técnicas de segmentación de umbrales con las que cuenta openCV en (Figura 6.5).

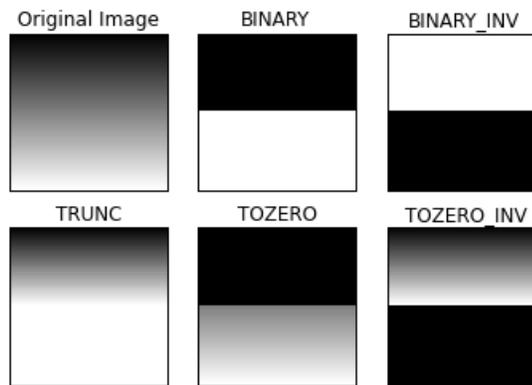


Figura 6.5: Umbrales

Por otra parte, si el valor de umbral es desconocido existen algoritmos más potentes que haciendo uso del histograma de la imagen pueden calcular un valor estimado de umbral tal que reduce el error en la segmentación, ejemplo de esta técnica es el algoritmo de otsu. Se muestra un ejemplo de este filtro a continuación:

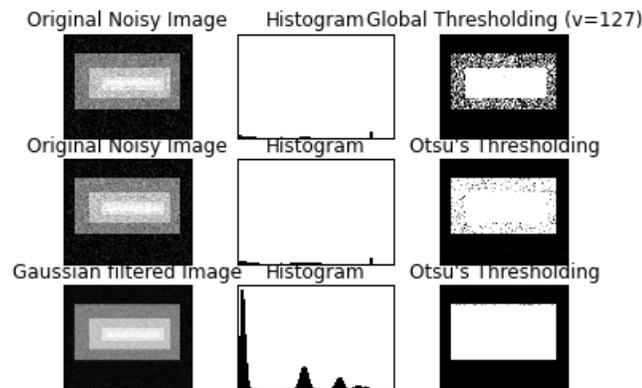


Figura 6.6: Error en la segmentación

## 6.6. Calibración de la Cámara

Para realizar la calibración de la cámara es necesario tomar varias fotografías a un patrón definido, para este caso se utiliza un tablero de ajedrez. En las imágenes en las que se puede identificar el patrón de ajedrez (Figura ??), se estima el punto en el mundo real de donde se ha tomado la foto y la coordenada equivalente en la

foto. Una vez se tiene un mínimo de 10 imágenes se para a calcular la matriz de la cámara, la cual tiene los parámetros intrínsecos de la cámara y los coeficientes de distorsión. Son estos últimos valores los que nos permitirán hacer la corrección del error radial y tangencial de la cámara.

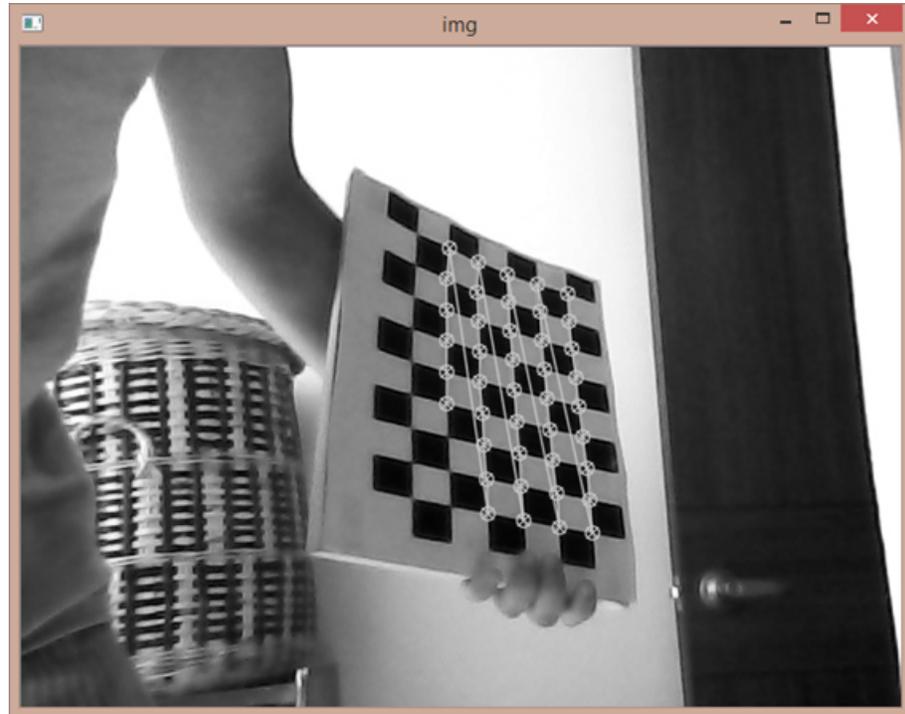


Figura 6.7: Patrón de ajedrez identificado en el tablero

Los valores que arrojo el cálculo de la matriz para la cámara del portátil fueron:

Camera matrix:

$$\begin{bmatrix} 771,70421552 & 0 & 250,24615454 \\ 0 & 769,20371322 & 302,38472516 \\ 0 & 0 & 1 \end{bmatrix}$$

Distortion coefficients:

$$[-1,52917056e - 02 \quad 4,97424197e + 00 \quad 1,69034164e - 02 \quad -7,57046164e - 03 \quad -2,3319302e - 03]$$

Una vez terminados estos cálculos se pasó a la fase de estimación de la profundidad, para esto se simuló la toma de las fotos estereoscópicas con la cámara del portátil (Figura 6.8). Inicialmente se tomó una foto del fondo, luego se hace un desplazamiento de 5cm del portátil y se vuelve a tomar la foto al mismo fondo. Una vez se tiene estas dos imágenes se hace el cálculo de la densidad de profundidad con una función del framework openCV. Figura 6.8



Figura 6.8: Mapa de profundidad

Si bien en este punto se puede apreciar que la silueta de la persona es lo más cercano a la imagen, todavía es necesario una etapa de calibración de los parámetros de la función para obtener una imagen de mayor calidad.

## 6.7. Detección por plantillas

La metodología de detección por plantillas implica la búsqueda y posterior ubicación de una determinada plantilla en una imagen (Figura 6.9), que para este caso es un frame (Figura 6.9) de un video en tiempo real. De manera que en dicho ejercicio se implementó la búsqueda de un libro en específico, el cual mostró en una primera instancia la ubicación del libro en el frame, posteriormente al desplazarse se seguía ubicando formándose así un seguimiento, incluso este tipo de seguimiento permite seguir detectando el objeto aunque una parte del mismo se encuentre obstruida (Figura 6.9), o en el ambiente se presente cambios de luz (Figura 6.10).



Figura 6.9: Plantilla, Detección normal, Detección con obstrucción  
 Imagen del libro “Doce Cuentos Peregrinos” del escritor Gabriel García Márquez

Aunque se aclara que en el caso del cambio de luz afecta en gran forma tanto la cámara en uso, como la calibración de la misma. Además, se ubica el objeto aunque éste se encuentre con cierto grado de inclinación o se presente un cambio en su rotación, lo cual se evidencia en la figura 6.10.

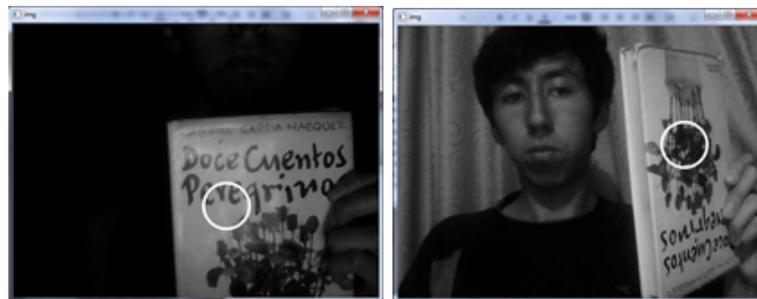


Figura 6.10: Detección con poca luz, Detección con inclinación y rotación  
 Imagen del libro “Doce Cuentos Peregrinos” del escritor Gabriel García Márquez

No obstante, cuando se presentan cambios como el acercamiento del objeto a la cámara se pierde la detección, entrando a confundirlo con cualquier otro objeto de características similares en el frame; lo que de igual forma pasa cuando se aleja el objeto de la cámara (Figura 6.11). Aunque se aclara que en objetos de características únicas dentro del ambiente de exposición se logra realizar la detección aunque parcialmente (Figura 6.11).

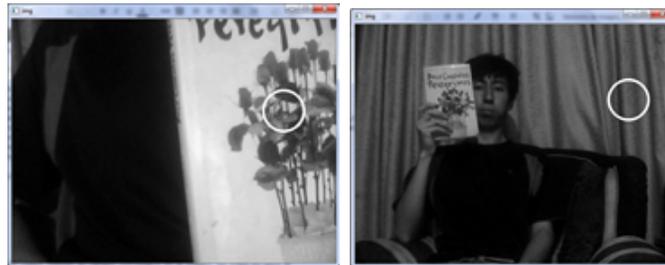


Figura 6.11: Detección cercana, No detección  
Imagen del libro “Doce Cuentos Peregrinos” del escritor Gabriel García Márquez



Figura 6.12: Segundo caso detección

Luego de varios intentos con múltiples plantillas se corroboró que el seguimiento por plantillas se desempeña mejor con objetos que poseen características únicas en el ambiente en el que se están grabando, de modo que en un ambiente con características similares se pueden presentar inconsistencias (Figura 6.12).

Los filtros utilizados para la detección del objeto mediante plantillas son:

- `cv2.TM_CCOEFF`.
- `cv2.TM_CCOEFF_NORMED`.
- `cv2.TM_CCORR`.
- `cv2.TM_CCORR_NORMED`.
- `cv2.TM_SQDIFF`.
- `cv2.TM_SQDIFF_NORMED`.



Figura 6.13: Cambio morfológico

Los cuales realizan un cambio morfológico (Figura 6.13) en la imagen del video es decir la suavizan y/o distorsionan, para así realizar la detección de la plantilla.

## 6.8. Detección por color

La búsqueda y detección de objetos por color implica el filtrado de colores específicos en la imagen, de forma que para el reconocimiento del objeto se busca la mayor concentración de color, lo cual supone que esa concentración está indicando la presencia del objeto (Figura 6.14, 6.15).

Debido a que se usa un rango de colores, esta metodología de reconocimiento se adapta a tanto a los cambios de tonalidad como a los de luz. Aunque si esos cambios son muy bruscos o cubren una gran parte del objeto, el reconocimiento falla presentando así inconsistencias.

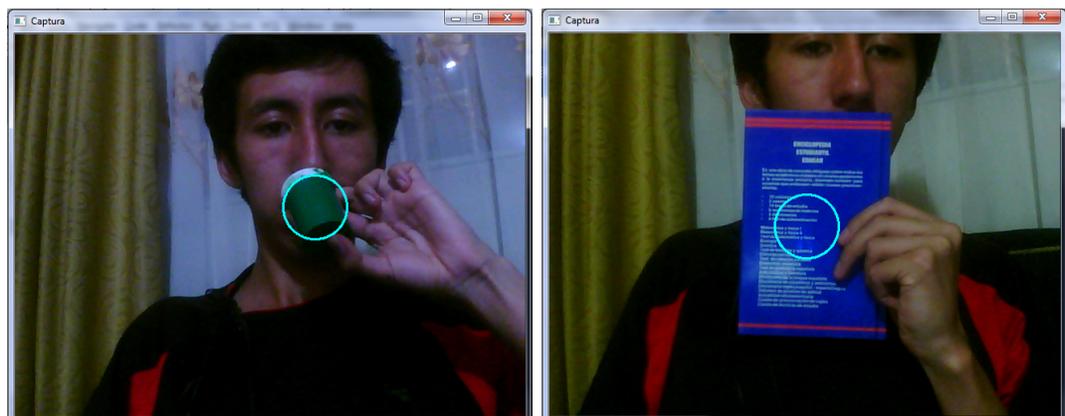


Figura 6.14: Detección en verde, detección en azul  
Imagen del libro “Arte Música y Literatura” del escritor Benhur Sánchez Suárez

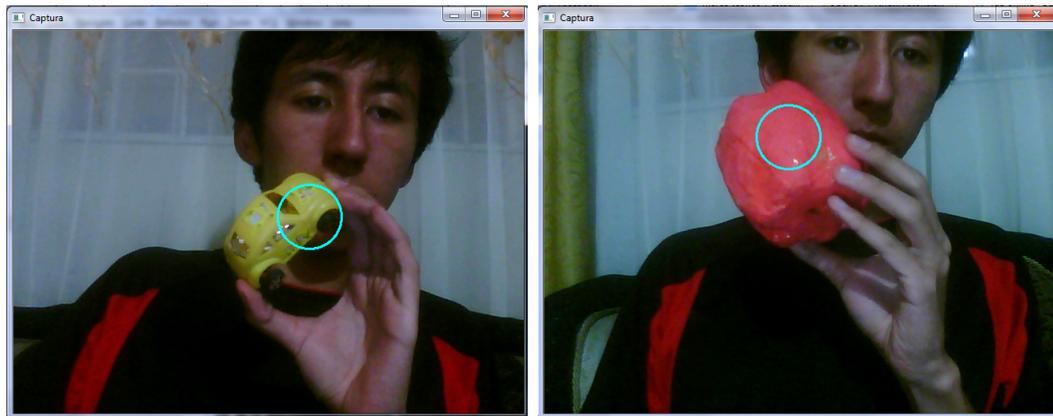


Figura 6.15: Detección en amarillo, detección en rojo

Se hacen usos de filtros los cuales solo revelan las ocurrencias de similitudes entre el rango de colores definidos y los obtenidos de la imagen del video, (Figuras 6.16 - 6.19), luego se retiran aquellas pequeñas ocurrencias evitando así objetivos erróneos, para después limpiar los pequeños errores dentro el objeto y así determinar finalmente su posición relativa.

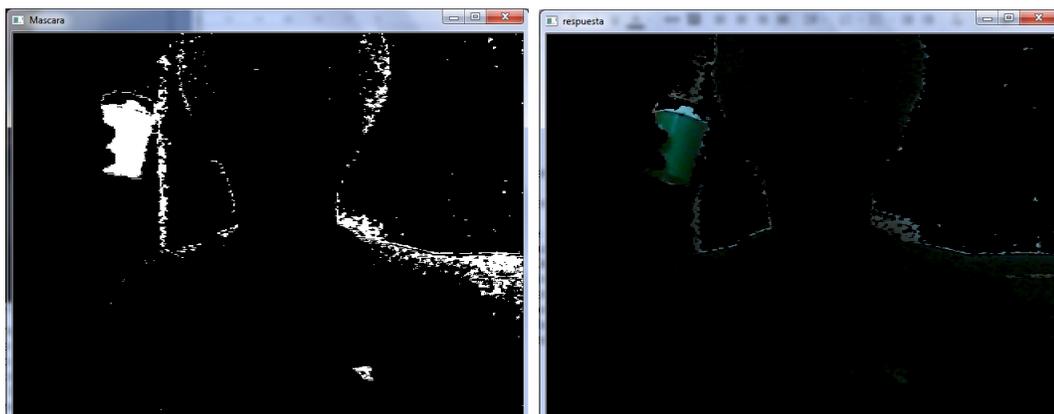


Figura 6.16: Mascara en verde, respuesta en verde

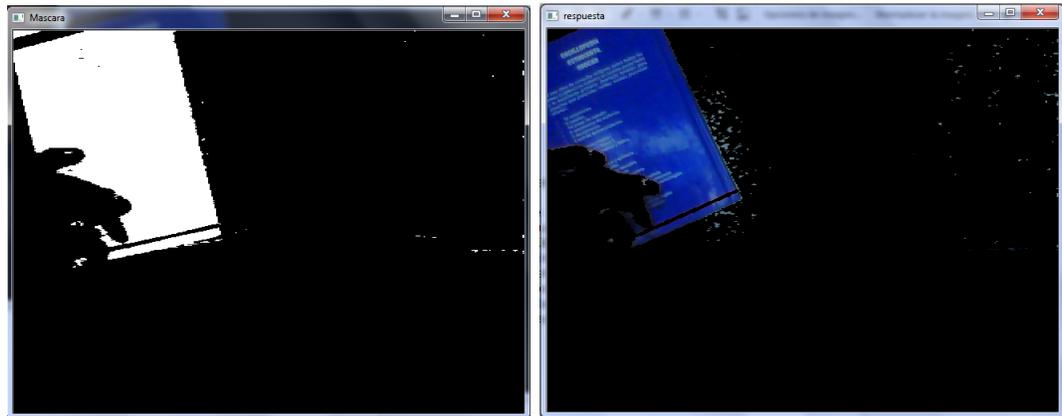


Figura 6.17: Mascara en azul, respuesta en azul  
Imagen del libro “Arte Música y Literatura” del escritor Benhur Sánchez Suárez

No obstante, la búsqueda únicamente por color está limitada solamente al reconocimiento de objetos que tengan un color similar al que se está buscando, por lo cual considera como uno solo, múltiples objetos que tengan el mismo color, además, en entornos con colores similares y en mayor concentración se presentaron inconsistencias (Figura 6.20).

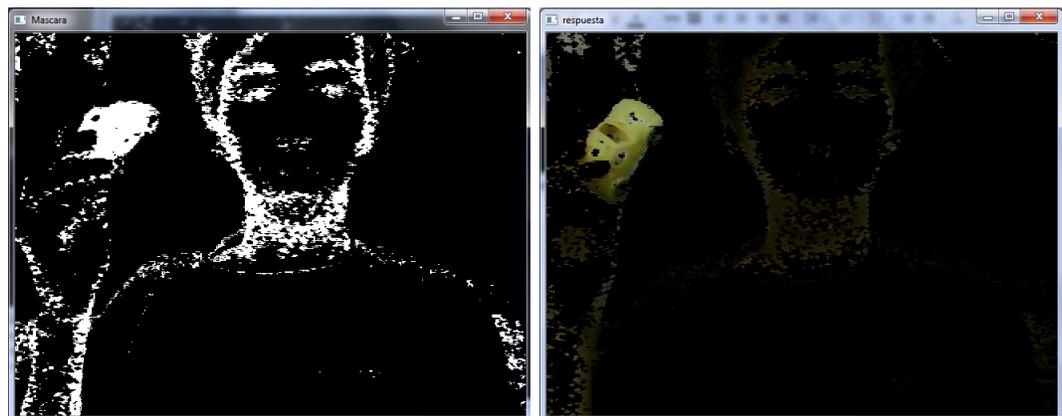


Figura 6.18: Mascara en amarillo, respuesta en amarillo

## 6.9. DETECCIÓN, CORRELACIÓN Y SELECCIÓN DE CARACTERÍSTICAS Y ESQUINAS45

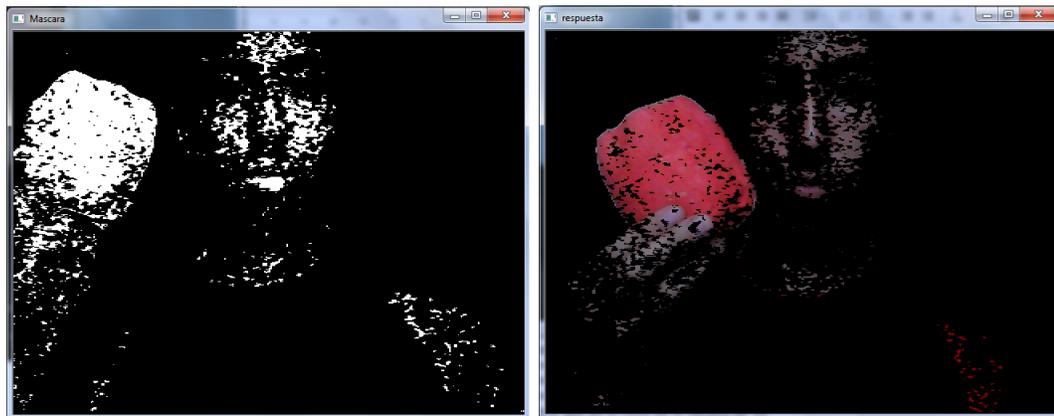


Figura 6.19: Mascara en rojo, respuesta en rojo

Esta metodología respecto a la rotación y/o inclinación del objeto es indiferente a menos que estos cambios lleguen a afectar el color expuesto por el objeto. Se aclara que en el caso del cambio de luz afecta en gran forma la cámara en uso, así como la calibración de la misma.



Figura 6.20: Inconsistencia en amarillo, inconsistencia en verde

## 6.9. Detección, correlación y selección de características y esquinas

Al momento de describir una imagen se deben definir descriptores, los cuales son puntos o segmentos específicos que definen ciertas particularidades de la imagen,

para lo cual se han definido elementos llamados “esquinas”, las cuales cumplen con dicha tarea, para así facilitar luego su búsqueda y ubicación en múltiples imágenes. El concepto de esquinas consiste básicamente en la detección de aquellos puntos o lugares en los cuales se presenta un fuerte cambio en la intensidad de los colores.

### 6.9.1. Harris Corner Detection

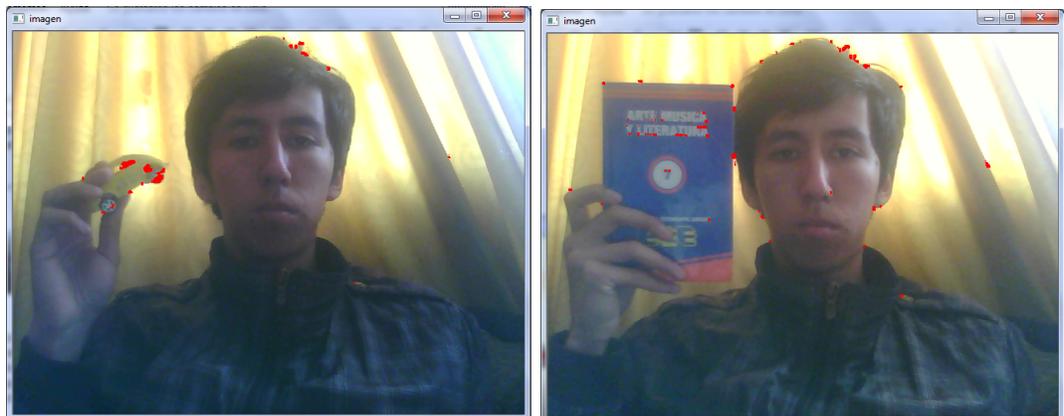


Figura 6.21: Esquinas de Harris  
Imagen del libro “Arte Música y Literatura” del escritor Benhur Sánchez Suárez

En un primer caso se presenta el uso del algoritmo de detección de esquinas conocido como “Harris Corner Detection” perteneciente a Chris Harris y Mike Stephens, en el cual se constató la ubicación de características especiales dentro de los distintos frames, orientado a resaltar aquellas características propias de ciertos objetos (Figura 6.21). Durante la realización de la práctica se verificó la influencia que tiene la luz sobre la ubicación de estas características (Figura 6.22); aunque se verificó la capacidad de detectar las mismas esquinas aunque se mueva el objeto.

## 6.9. DETECCIÓN, CORRELACIÓN Y SELECCIÓN DE CARACTERÍSTICAS Y ESQUINAS 47

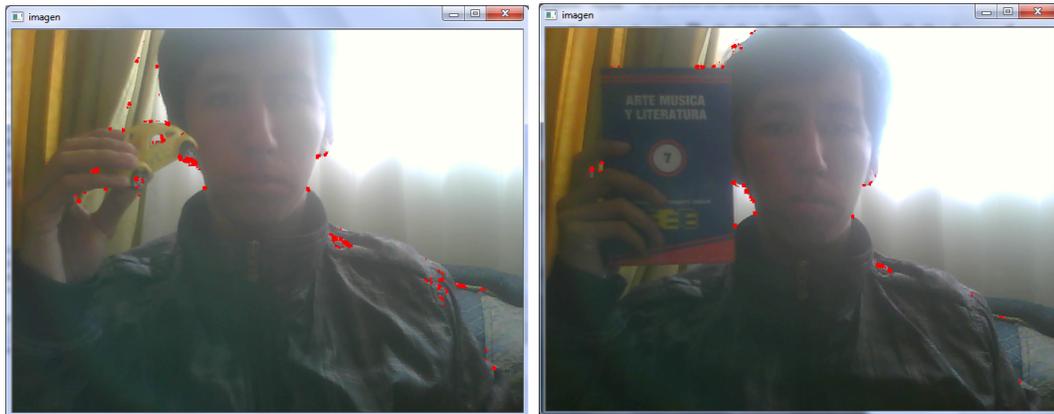


Figura 6.22: Esquinas de Harris 2  
Imagen del libro “Arte Música y Literatura” del escritor Benhur Sánchez Suárez

### 6.9.2. Sub-Pixel Accuracy

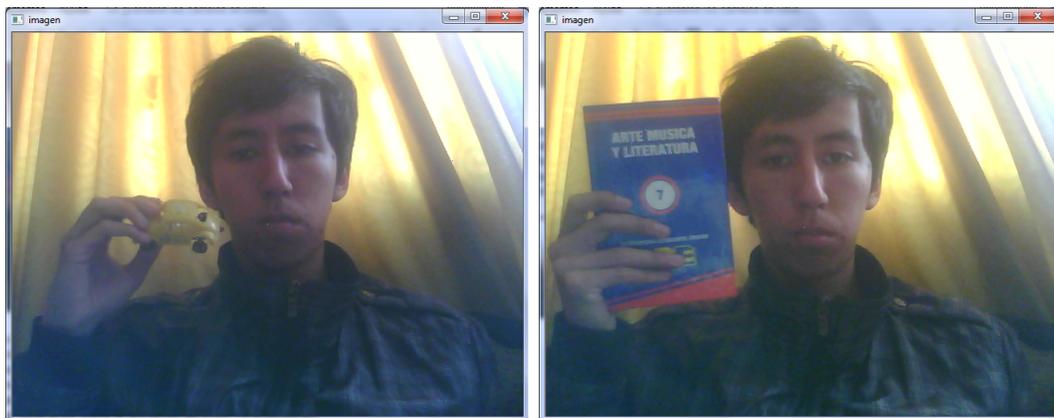


Figura 6.23: Esquinas de Sub-Pixel  
Imagen del libro “Arte Música y Literatura” del escritor Benhur Sánchez Suárez

Se hizo uso de un algoritmo de refinamiento de esquinas llamado “Sub-Pixel Accuracy” en el cual dadas las esquinas obtenidas por el detector “Harris Corner Detection”, se define un rango de píxeles vecinos específico en el cual se busca aquellos píxeles en los cuales se presenta la característica especial en el frame (Figura 6.23), aunque se presenta la misma desventaja del detector en uso.

### 6.9.3. Shi-Tomasi Corner Detector

Se usó el algoritmo de J. Shi and C. Tomasi, llamado “Shi-Tomasi Corner Detector”, el cual es una mejora de “Harris Corner Detection”, en el cual se especifica que para la selección de una esquina se filtran aquellas con el umbral de cambio más grande. Durante la realización de la práctica se verificó la influencia que tiene la luz sobre la ubicación de estas características.(Figura 6.24)



Figura 6.24: Esquinas de Shi-Tomasi  
Imagen del libro “Arte Música y Literatura” del escritor Benhur Sánchez Suárez

### 6.9.4. SIFT (Scale-Invariant Feature Transform)

A pesar de que los algoritmos previos se adaptan en distinta forma según los cambios que se presentaban en el ambiente expuesto, como luz o movimiento, estos dependen totalmente de que el tamaño de la imagen se mantenga igual o que la misma no se haya escalado, ya que en caso de que se estuvieran comparando dos imágenes de diferente tamaño o que el objeto se alejará o acercará a la cámara, aparecerán nuevas esquinas o incluso se mantengan algunas de las ya encontradas, pero lo más probable es que se pierdan la mayoría de las esquinas encontradas previamente.

## 6.9. DETECCIÓN, CORRELACIÓN Y SELECCIÓN DE CARACTERÍSTICAS Y ESQUINAS49

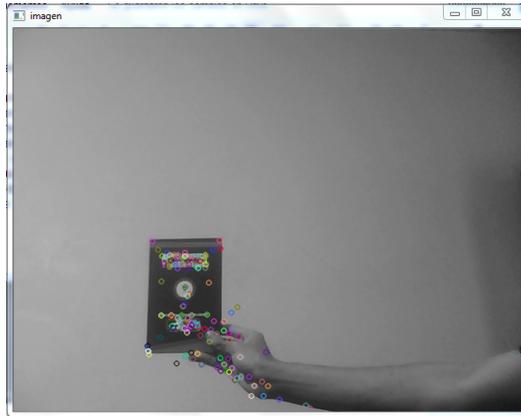


Figura 6.25: Características de SIFT 1  
Imagen del libro “Arte Música y Literatura” del escritor Benhur Sánchez Suárez

Por lo cual para cubrir este caso se hace uso del algoritmo SIFT, el cual cubre estas necesidades, mediante el uso de un detector de manchas para el escalamiento en las imágenes formalmente conocido como “Laplacian of Gaussian”, para luego utilizar “Difference of Gaussians”, en pro de obtener una aproximación de la ubicación de las esquinas, para finalmente precisar la ubicación mediante una matriz Hessian y series de Taylor.

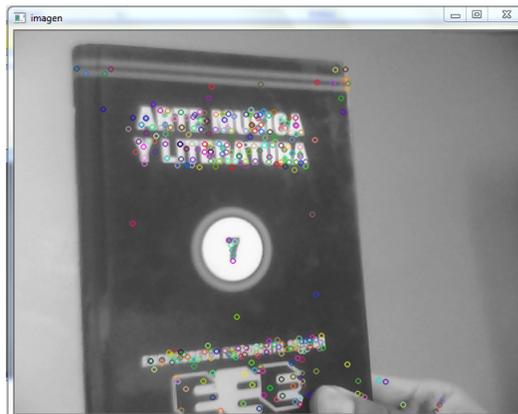


Figura 6.26: Características de SIFT 2  
Imagen del libro “Arte Música y Literatura” del escritor Benhur Sánchez Suárez

Como se esperaba al algoritmo en uso permite mantener el reconocimiento de las esquinas encontradas cuando el objeto a detectar es pequeño, respecto a cuando se encuentra cerca a la cámara, en donde se muestra una versión más grande del

mismo (Figura 6.25, 6.26), aunque se reconocieron nuevas esquinas, las mismas ya se encontraban previamente señaladas dentro de una esquina detectada en la versión pequeña del objeto, sin embargo, se presentó retraso en el procesamiento de los puntos clave, mientras se estaba probando con un video en tiempo real.

### 6.9.5. SURF (Speeded-Up Robust Features)

En vista de la velocidad de procesamiento de descriptores de SIFT, en 2006 se desarrolló un nuevo algoritmo orientado a mejorar el cumplimiento de este aspecto de SIFT [18], para ello mantuvo la técnicas características de SIFT, pero hace uso de una caja de filtro, la cual permite el procesamiento de esquinas en diferentes escalas y en paralelo. Además, SURF permite determinar la orientación de las esquinas aproximándose por concentración de ocurrencias cerca al punto de ubicación de la esquina, aunque deshabilitando esta característica la velocidad se incrementa. Se aclara que se mantiene debilidad ante los cambios de luz sobre el campo de exposición.

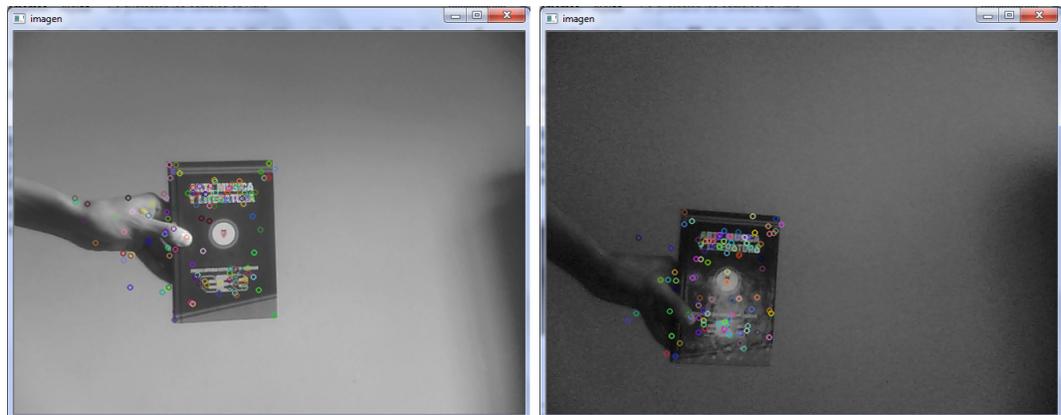


Figura 6.27: Características de SURF  
Imagen del libro “Arte Música y Literatura” del escritor Benhur Sánchez Suárez

Durante la prueba se verificó el desempeño del algoritmo en un video en tiempo real, el cual fue bueno, demostrando así su superioridad en velocidad respecto a SIFT; aunque en teoría debería presentarse algunas inconsistencias con los cambios de luz, en la prueba se mantuvo el reconocimiento de las mismas esquinas, al pasar de un ambiente iluminado a uno oscuro (Figura 6.27).

### 6.9.6. FAST (Features from Accelerated Segment Test)

El algoritmo en mención se desarrolló orientado más que todo a un bajo consumo de recursos, debido a su aplicación en la robótica. La búsqueda, determinación y ubicación de descriptores, se hace mediante la selección de una esquina, que luego se considera como el centro de una circunferencia de evaluación, en esta circunferencia se evalúan 4 puntos extremos, el superior con el inferior y luego los laterales. Lo cual se puede asemejar a un reloj, en el cual se comparan los píxeles encontrados a las 12 y a las 6, y luego aquellos que se encuentren a las 3 y a las 9, dicha evaluación está orientada al manejo de los cambios de brillo e intensidad, por lo cual para retirar aquellas esquinas innecesarias se descartan aquellas que tengan un menor cambio de intensidad respecto a las circundantes.

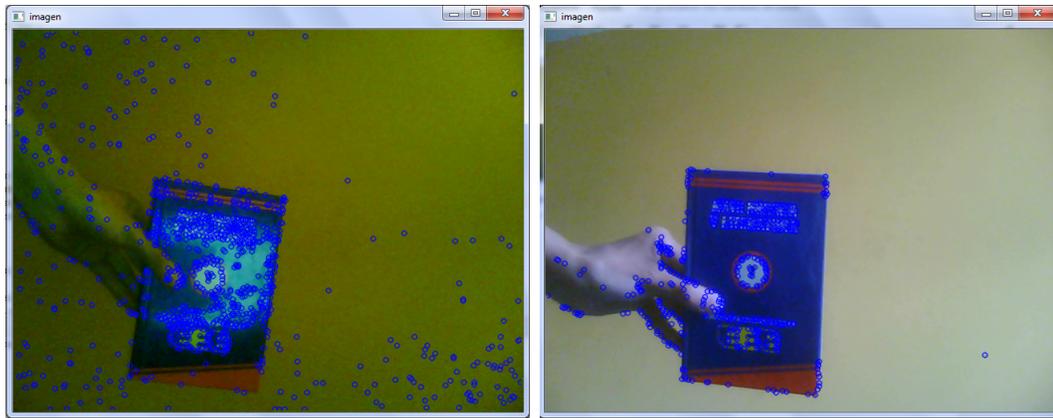


Figura 6.28: Características de FAST  
Imagen del libro "Arte Música y Literatura" del escritor Benhur Sánchez Suárez

Durante la prueba se usó el algoritmo para la detección de esquinas en un video en tiempo real, con cambio en la intensidad de la luz presente el campo de exposición, con lo cual se corroboró la dependencia que se presenta respecto al cambio en la iluminación del medio, así como la permanencia de las esquinas que se encontraron previamente, como se observa en la figura 6.28, cualquier brillo sostenido en alguna parte de la imagen o un misma tonalidad en cierto espacio, no presenta ninguna detección de esquina.

### 6.9.7. ORB(Oriented FAST and Rotated BRIEF)

El algoritmo en mención, como su nombre lo indica, está compuesto por las capacidades de FAST, pero aprovechando las ventajas de un descriptor BRIEF, los cuales surgieron en razón de la complejidad que se manejaba desde la implementación de SIFT y SURF, por lo cual este hace uso de una cadena binaria en lugar de descriptores, esta cadena binaria proviene de la identificación hash de los descriptores, de forma que el tiempo requerido para la evaluación de las múltiples dimensiones presentes en los descriptores se elimina, mejorando así la velocidad de procesamiento de descriptores.

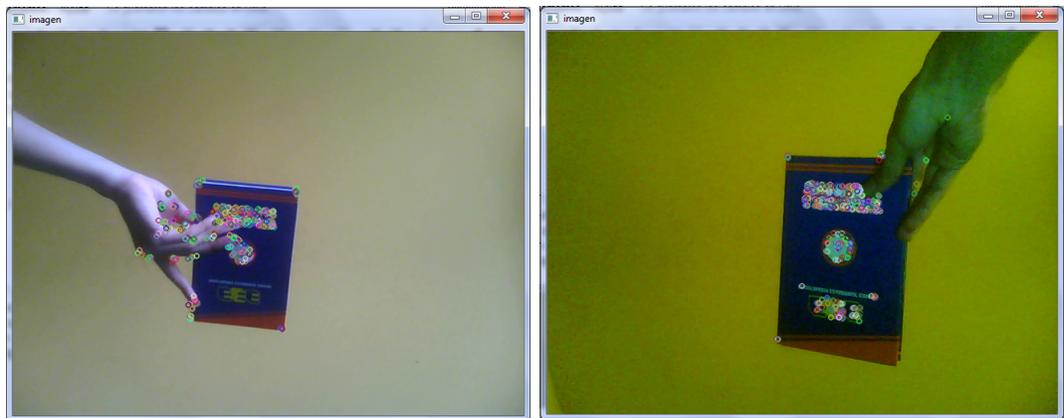


Figura 6.29: Características de ORB  
Imagen del libro “Arte Música y Literatura” del escritor Benhur Sánchez Suárez

La prueba realizada sobre un video en tiempo real, con el algoritmo ORB, permitió comprobar el desempeño prometido de FAST, además de que no se presentan esquinas en pequeños cambios de luz, aunque el cambio en la intensidad de luz de la misma afectó el reconocimiento de esquinas, entre un estado y el otro, como se expone en la imagen 6.29.

## 6.10. Detección según características y descriptores, análisis de información

### 6.10.1. SIFT (Scale-Invariant Feature Transform)

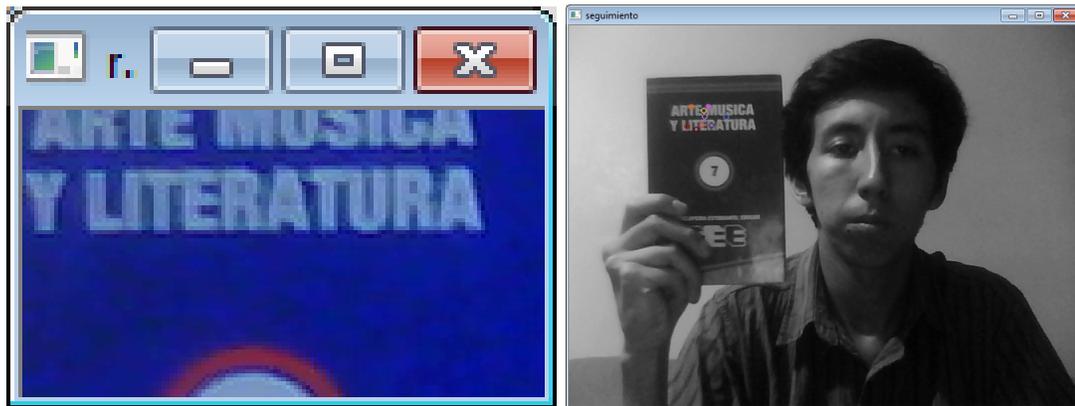


Figura 6.30: Imagen de muestra, características de SIFT  
Imagen del libro “Arte Música y Literatura” del escritor Benhur Sánchez Suárez

Luego de haber evaluado la forma en la que el algoritmo SIFT, describe determinadas características especiales en una imagen, se procederá a aplicarlo para la detección de objetos, para lo cual a partir de la imagen del objeto que se desea detectar, se buscará en los distintos frames que se generen en el video de tiempo real, la ubicación de este a partir de las comparaciones entre los descriptores generados en los frames del video y aquellos generados a partir de la imagen guía, de forma que solo se muestran aquellos que indican la posición del objeto en seguimiento (Figura 6.30).



Figura 6.31: Características de SIFT  
Imagen del libro “Arte Música y Literatura” del escritor Benhur Sánchez Suárez



Figura 6.32: Características de SIFT 2  
Imagen del libro “Arte Música y Literatura” del escritor Benhur Sánchez Suárez

## 6.10. DETECCIÓN SEGÚN CARACTERÍSTICAS Y DESCRIPTORES, ANÁLISIS DE INFORM

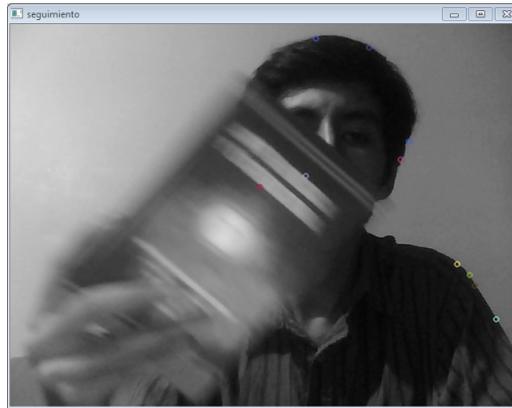


Figura 6.33: Características de SIFT con movimiento  
Imagen del libro “Arte Música y Literatura” del escritor Benhur Sánchez Suárez

Durante el proceso de prueba se permitió comprobar la debilidad o presencia de inconsistencias cuando el movimiento del objeto es mucho, de forma que se muestran descriptores con características similares a algunos de los generados a partir del objeto (Figura 6.33). Aunque se permitió el reconocimiento del objeto aun cuando este se acercara o alejara de la cámara, claro que hasta cierta distancia, además el reconocimiento continuo cuando el objeto se ha rotado o inclinado (Figura 6.31, 6.32).

### 6.10.2. SURF (Speeded-Up Robust Features)



Figura 6.34: Imagen de muestra de SURF, características de SURF  
Imagen del libro “Arte Música y Literatura” del escritor Benhur Sánchez Suárez

Así como se evaluó el desempeño del algoritmo SIFT la definición de descriptores para la detección de objetos, se procede a realizar una prueba del algoritmo SURF, en el cual se espera mejor desempeño, realizando la misma prueba anterior, es decir, que a partir de la imagen del objeto que se desea seguir, se buscará en los distintos frames que se generen en el video en tiempo real, la ubicación de éste a partir de las comparaciones entre los descriptores generados en los frames del video y aquellos generados a partir de la imagen guía, de forma que solo se muestran aquellos que indican la posición del objeto en seguimiento (Figura 6.34).



Figura 6.35: Características de SURF  
Imagen del libro “Arte Música y Literatura” del escritor Benhur Sánchez Suárez

Tal como se esperaba el rendimiento del algoritmo SURF, fue mayor, debido a que no afectó tanto la fluidez del video y se mantuvieron las mismas capacidades de reconocimiento expuestas en la prueba anterior, es decir, se logró el reconocimiento del objeto aun cuando este se acercara o alejara de la cámara, claro que hasta cierta distancia, además el reconocimiento continuo cuando el objeto se ha rotado o inclinado; pero persistieron los mismos problemas (Figura ).

### 6.10.3. ORB(Oriented FAST and Rotated BRIEF))

El algoritmo elegido para el seguimiento de objetos es ORB, de manera que a continuación se muestran las pruebas realizadas, se aclara que en esta ocasión no se selecciona el objeto a seguir desde el video, sino que se obtiene desde una fotografía obtenida por otra fuente, para ayudar en el reconocimiento de la imagen en distintos tamaños se altera el tamaño de la fotografía de muestra suministrada (Fi-

## 6.10. DETECCIÓN SEGÚN CARACTERÍSTICAS Y DESCRIPTORES, ANÁLISIS DE INFORM

gura 6.36) a la mitad del ancho de los fotogramas manejados por la cámara de video.

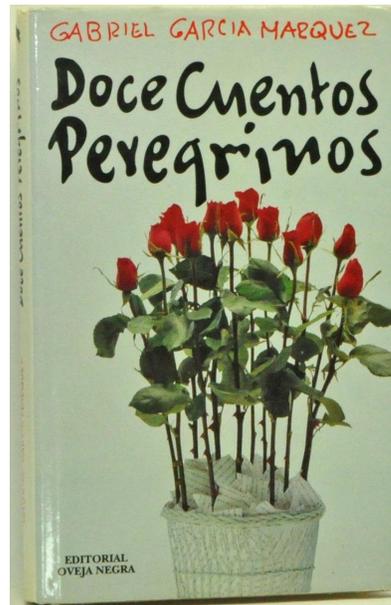


Figura 6.36: Imagen de muestra de ORB  
Imagen del libro "Doce Cuentos Peregrinos" del escritor Gabriel García Márquez

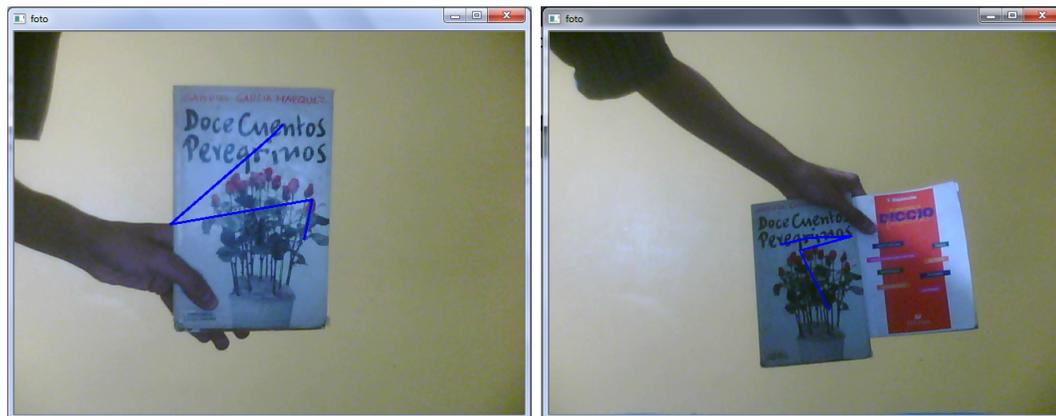


Figura 6.37: Características de ORB  
Imagen del libro "Doce Cuentos Peregrinos" del escritor Gabriel García Márquez Imagen del libro  
"DICCIOFormulas" del autor Y. Deplanche

Como se muestra en la fotografía 6.37, se realizó la búsqueda del objeto a seguir teniendo en cuenta cierto brillo proyectado por el objeto, se proyectó sombra sobre el

objeto y además se colocó al lado de otro objeto con características similares; los resultados obtenidos en las pruebas fueron satisfactorios logrando reconocimiento con presencia de brillo, sombra y posible confusión con un objeto de características similares. Aunque el desempeño disminuye conforme incrementa el número de imágenes de muestra.

## 6.11. Detección según características con filtro de color

Para la detección de plantas se había seleccionado previamente los algoritmos SURF y ORB, los cuales evalúan características o descriptores, para el reconocimiento de un objeto en específico, no obstante, al hacer pruebas de campo se observó la presencia de múltiples falsos positivos, lo cual hacía necesario afinar las definiciones de reconocimiento, por lo cual se intentó plantear la detección en forma segmentada, es decir, sesgar el espacio de búsqueda, mediante con la comprobación de correlación entre el color dominante en la imagen del objetivo y el frame en evaluación, de forma que solo se busquen las características de la planta buscada en aquellos lugares del frame de video en el cual se encuentra una concentración de color igual o similar. (Figura 6.38)



Figura 6.38: Imagen de muestra a color para características y color, caso exitoso características y color

Claro que inicialmente se está evaluando comparación de colores dominantes sólo en escala de grises (Figura 6.39), por lo cual se pueden presentar algunos falsos positivos, pero es solo en estas secciones donde se evaluarán las características definidas

### 6.11. DETECCIÓN SEGÚN CARACTERÍSTICAS CON FILTRO DE COLOR59

por los algoritmos previamente mencionados. Algunos de los resultados se pueden observar en (Figura 6.40).

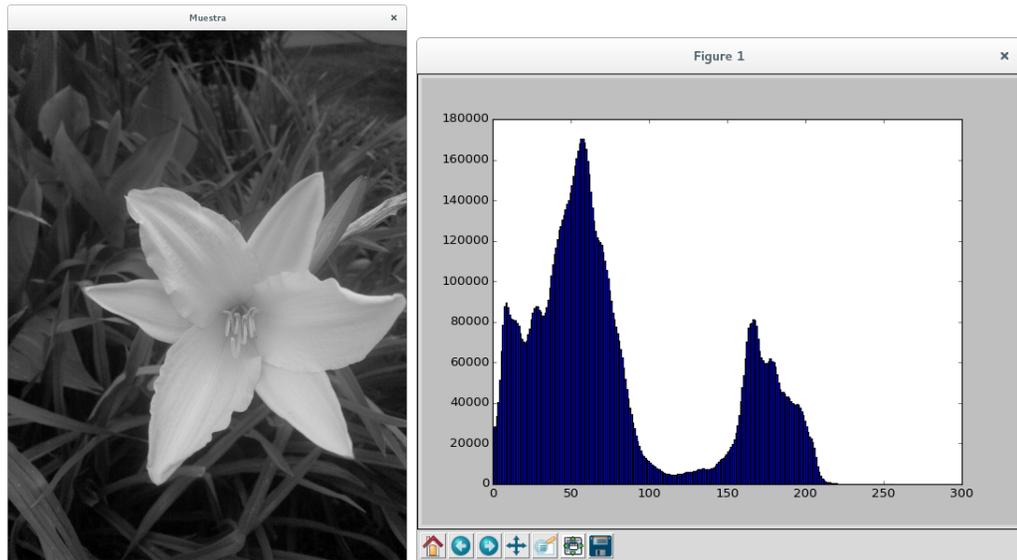


Figura 6.39: Imagen de muestra en grises para características y color, gráfico de intensidad de color

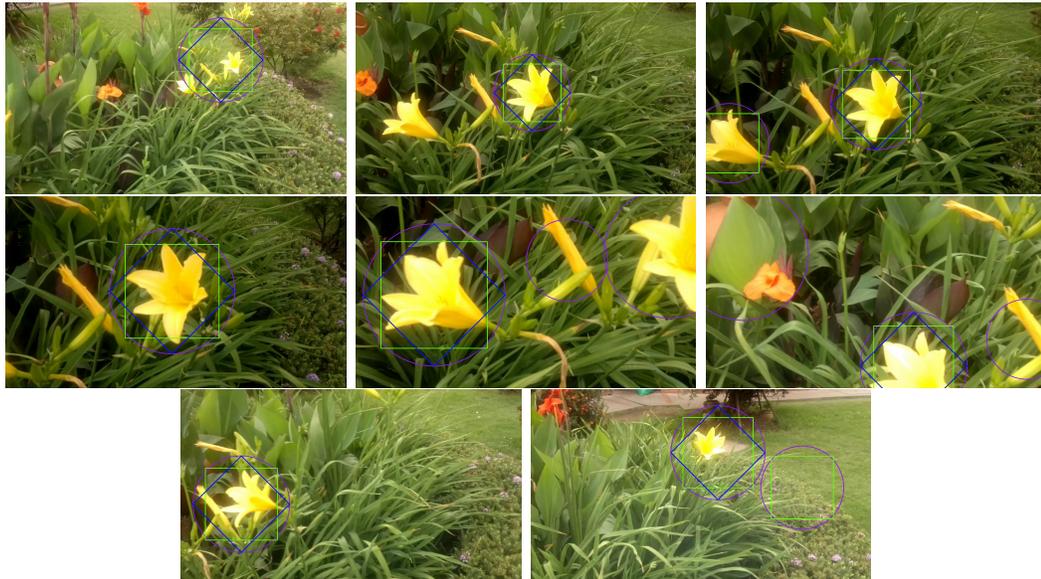


Figura 6.40: Resultados exitosos en las pruebas de lirios amarillos

En una segunda prueba se obtuvieron los resultados obtenidos en las imágenes 6.41 y 6.42

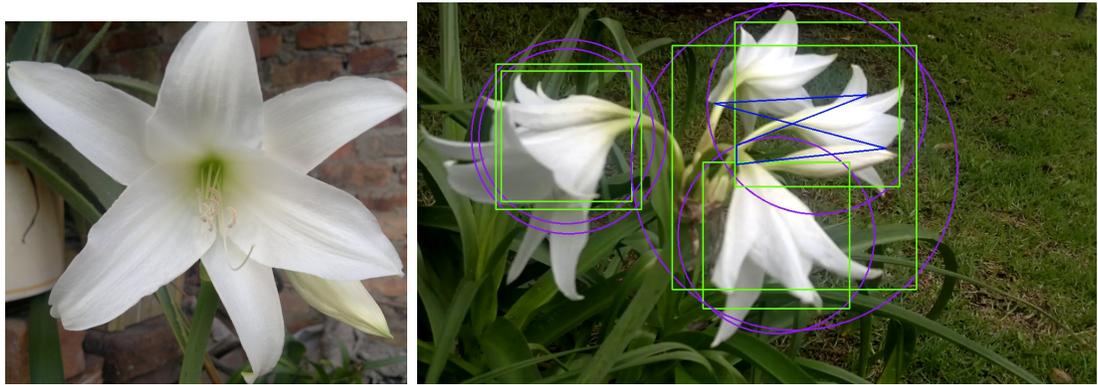


Figura 6.41: Imagen de muestra a color para característcias y color, caso exitoso característcias y color

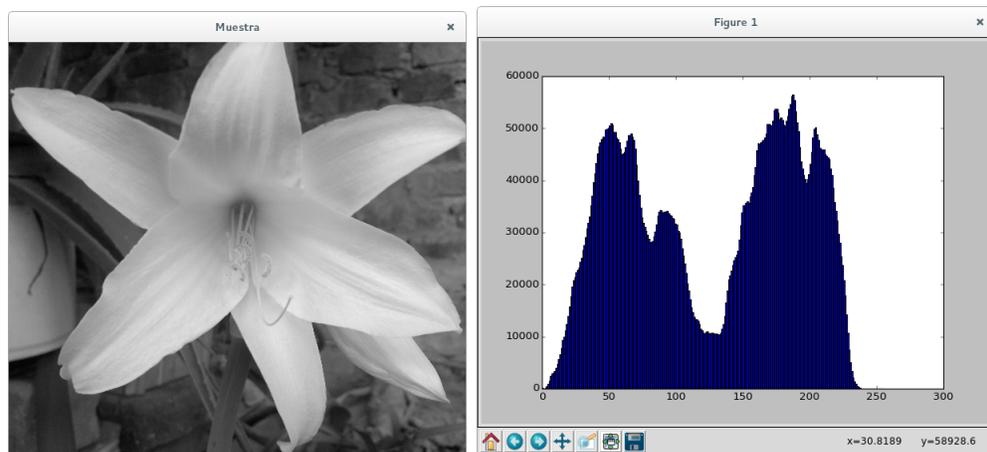


Figura 6.42: Imagen de muestra en grises para característcias y color, gráfico de intensidad de color

Se realizaron tres pruebas con un video, en el cual se grabó una planta de lirios blancos (Figura 6.42), (en un computador) en el que se buscaba encontrar 8 flores en buen estado, claro que aparecen otras en un estado previo al florecimiento y otras posteriores a éste estado, buscándose así que detecte en un inicio las flores por su color y en dicha zona se evalúan características obtenidas de la imagen del objetivo

### 6.11. DETECCIÓN SEGÚN CARACTERÍSTICAS CON FILTRO DE COLOR61

(Figura 6.41).

En la primera prueba se encontraron un total de 603 frames de flores encontradas, de las cuales 553 fueron casos exitosos y 116 falsos positivos, formándose así un total de 603 ocurrencias detectadas, de lo cual se esperaba 1420 ocurrencias de detección correctas, como se muestra en la gráfica 6.43. Lográndose una precisión del 17,34 % y una exhaustividad del 52,89 %.

$$precision = \frac{1420 - 669}{1420}$$

$$exhaustividad = \frac{1420 - 669}{1420}$$

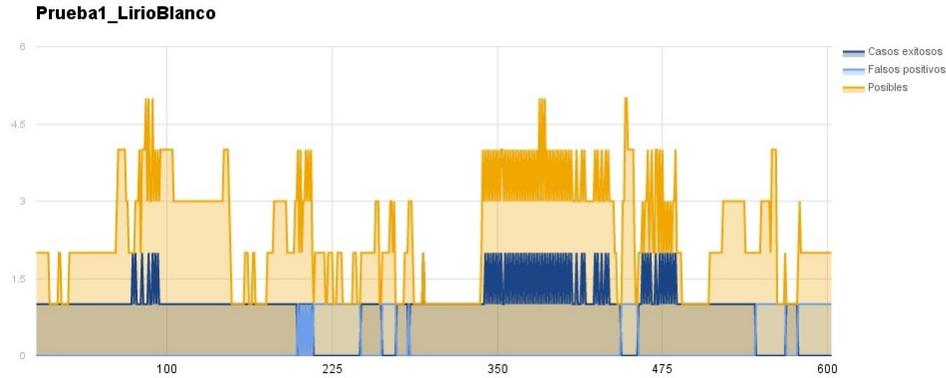


Figura 6.43: Gráfico de áreas entre éxitos, falsos positivos, y opciones posibles.

Con el mismo objetivo de la prueba anterior se realizaron 3 pruebas con un video en el cual se grabó una planta de lirios amarillos, (en un computador) en el que se buscaba encontrar 3 flores en buen estado, claro que aparecen otras en un estado previo al florecimiento y otras posteriores a éste estado. En la primera prueba se encontraron un total de 225 frames de flores encontradas, de las cuales 177 fueron casos exitosos y 48 falsos positivos, formándose así un total de 225 ocurrencias detectadas, de lo cual se esperaba 344 ocurrencias de detección correctas, como se muestra en la gráfica (Figura 6.44). Lográndose una precisión del 21,32 % y una exhaustividad del 48,55 %.

$$precision = \frac{225 - 177}{225}$$

$$exhaustividad = \frac{344 - 177}{344}$$

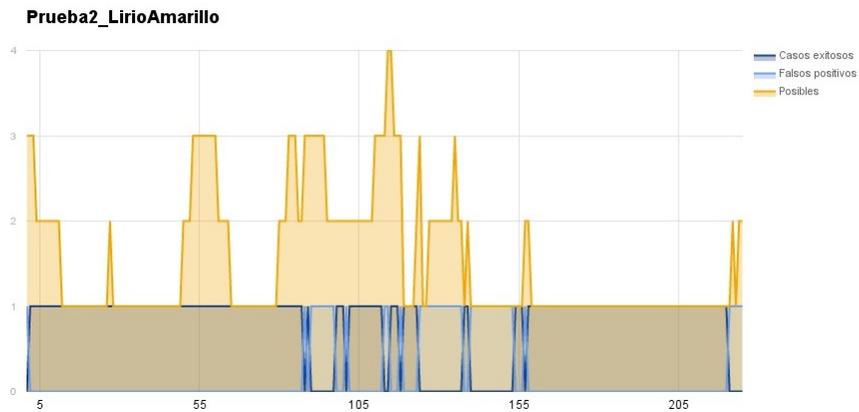


Figura 6.44: Gráfico de áreas entre éxitos, falsos positivos, y opciones posibles.

## 6.12. Estimación de distancias usando método de objeto conocido

Inicialmente se consideró un enfoque estereoscópico para la estimación de distancias a un objeto, si bien este enfoque parecía promisorio, la gran cantidad de información que debe ser procesada (2 frames para la estimación) imposibilita su implementación en un SoC para una aplicación en tiempo real, esto último siendo el objetivo deseado del proyecto. Por esta razón se migró a un enfoque monoscópico.

El fuerte de este algoritmo monoscópico consiste en estimar la distancia a un objeto previamente caracterizado, de este debemos conocer sus dimensiones físicas y su tamaño en pixeles a una distancia fija para poder extrapolar la distancia a la cual se encuentra.

La prueba consistió en estimar la distancia de una pelota de tenis a una distancia previamente establecida, básicamente deseábamos medir el error de la estimación que nos retorna el algoritmo y la distancia real a la cual se encuentra la pelota. La prueba se realizó en dos ambientes diferentes, uno con iluminación controlada y otro en el exterior.

En el ambiente con luz exterior los resultados hallados no son concluyentes, ya que la gran cantidad de variaciones en el brillo e iluminación afecta el reconocimiento del objeto conocido, por esta razón la distancia que el algoritmo arroja varía demasiado entre frame y frame. Este comportamiento errático nos permite concluir que el

## 6.12. ESTIMACIÓN DE DISTANCIAS USANDO MÉTODO DE OBJETO CONOCIDO63

método de identificación que usa el algoritmo es mejorable, siendo esto importante para una nueva iteración sobre el proyecto.

En la prueba en interior, si bien los resultados varían entre frame y frame, el intervalo de variación es mucho menor al encontrado en el exterior, estando acotado a unos cuantos centímetros por encima y por debajo de la distancia real. Dando este método una buena aproximación de la distancia real a la cual se encuentra el objeto en el mundo físico. A continuación se muestra una tabla que resumen los resultados hallados:

Distancia Real(cm)	Valor Obtenido (cm)	error
15	17,04	13,6
30	30.7	2.333333333
40	41.8	4.5
50	49.2	1.6
60	58.5	2.5
70	69.37	0.9
80	83.62	4.525
90	Mucha Variación	No concluyente

Se puede decir que el éxito de esta técnica se basa en el algoritmo de detección del objeto conocido, ya que como se pudo evidenciar, si no se detecta claramente el objetivo, no es posible estimar de forma correcta la distancia a la cual se encuentra. Por otra parte, en la prueba en interior vimos que a partir de distancias mayores a un metro, la técnica no daba las mejores respuestas debido a que la resolución de la cámara no permitía capturar el objeto de forma nítida.

Adicionalmente, a pequeñas distancias el estimado que retorna el algoritmo contiene un porcentaje de error.

### 6.13. Rendimiento del módulo en la tarjeta Intel Galileo

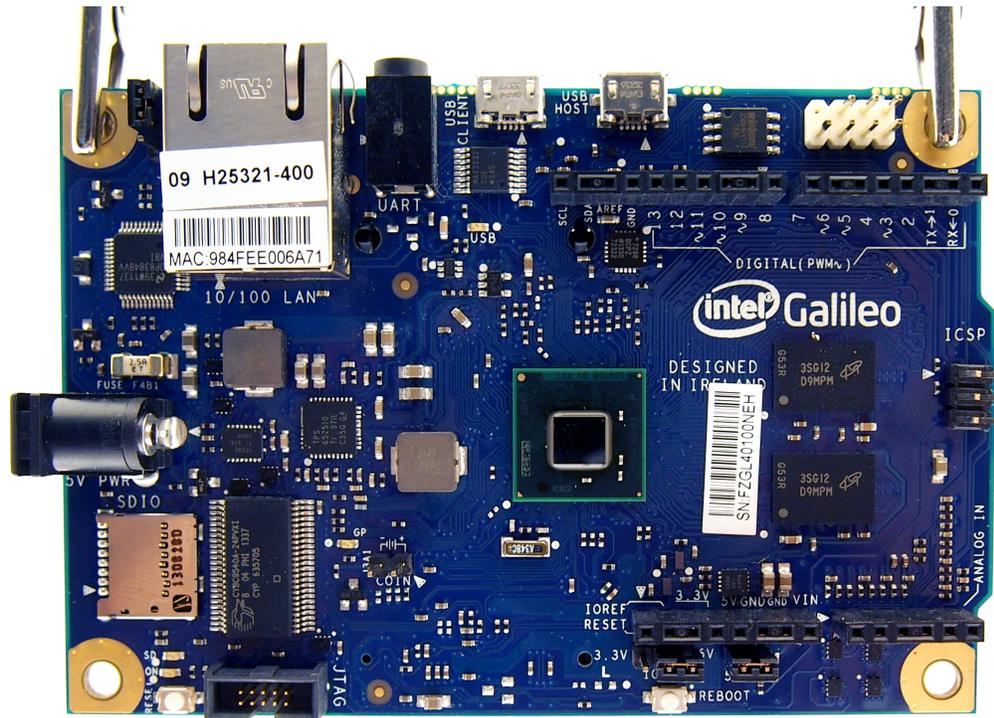


Figura 6.45: Intel Galileo Gen. 1

Durante las pruebas sobre la tarjeta (Figura 6.45) se evidenció su insuficiencia de recursos para ejecutar el módulo en forma adecuada, ya que uno de los requisitos de aceptación es que el módulo se pudiera desplegar en un video en tiempo real, lo cual se logró, pero con un rendimiento muy bajo, el cual se evidenció una demora mayor a un segundo en la evaluación de cada frame del video.

# Bibliografía

- [1] R. E. W. Rafael C. Gonzalez, *Digital Image Processing*. New Jersey: Prentice Hall, third edition ed., 2008.
- [2] C. Valero Ubierna, L. M. Navas Gracia, F. González Herrero, J. Gómez Gil, G. Ruiz Ruiz, P. Barreiro Elorza, D. Andújar Sánchez, B. Diezma Iglesias, E. Baguena Isiegas, and M. Garrido Izard, “Ahorro y eficiencia energética en la Agricultura de Precisión,” 2010.
- [3] J. F. Sotomayor, A. P. Gómez, and A. F. Cela, “Sistema de visión artificial para el análisis de imágenes de cultivo basado en texturas orientadas,” *Revista Politécnica*, vol. 33, no. 1, 2014.
- [4] G. Bradski and A. Kaehler, *Learning OpenCV: Computer vision with the OpenCV library*. .<sup>o</sup>Reilly Media, Inc.”, 2008.
- [5] S. Nedevschi, R. Danescu, D. Frentiu, T. Marita, F. Oniga, C. Pocol, R. Schmidt, and T. Graf, “High accuracy stereo vision system for far distance obstacle detection,” in *IEEE Intelligent Vehicles Symposium*, pp. 292–297, 2004.
- [6] R. I. Hartley and P. Sturm, “Triangulation,” *Computer vision and image understanding*, vol. 68, no. 2, pp. 146–157, 1997.
- [7] C. Szegedy, A. Toshev, and D. Erhan, “Deep Neural Networks for Object Detection,” in *Advances in Neural Information Processing Systems 26* (C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, eds.), pp. 2553–2561, Curran Associates, Inc., 2013.
- [8] A. Yilmaz, O. Javed, and M. Shah, “Object tracking: A survey,” *Acm computing surveys (CSUR)*, vol. 38, no. 4, p. 13, 2006.
- [9] H. Zhou, Y. Yuan, and C. Shi, “Object tracking using SIFT features and mean shift,” *Computer vision and image understanding*, vol. 113, no. 3, pp. 345–352, 2009.

- [10] M. J. Patel and B. Bhatt, “A Comparative Study of Object Tracking Techniques,” *International Journal of Innovative Research in Science, Engineering and Technology*, vol. 4, no. 3, 2015.
- [11] Itseez, “OpenCV-Python Tutorials,” 2014.
- [12] H. Grabner, J. Matas, L. Van Gool, and P. Cattin, “Tracking the invisible: Learning where the object might be,” in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pp. 1285–1292, IEEE, 2010.
- [13] E. Rosten, R. Porter, and T. Drummond, “Faster and better: A machine learning approach to corner detection,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 32, no. 1, pp. 105–119, 2010.
- [14] E. Mair, G. D. Hager, D. Burschka, M. Suppa, and G. Hirzinger, “Adaptive and generic corner detection based on the accelerated segment test,” in *Computer Vision–ECCV 2010*, pp. 183–196, Springer, 2010.
- [15] Itseez, “FAST Algorithm for Corner Detection,” 2014.
- [16] Itseez, “ORB (Oriented FAST and Rotated BRIEF),” 2014.
- [17] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, “ORB: an efficient alternative to SIFT or SURF,” in *Computer Vision (ICCV), 2011 IEEE International Conference on*, pp. 2564–2571, IEEE, 2011.
- [18] Itseez, “Introduction to SURF (Speeded-Up Robust Features),” 2014.