



UNIVERSIDAD DE QUINTANA ROO
DIVISIÓN DE CIENCIAS E INGENIERÍA

SEGUIMIENTO DE PERSONAS CON ROBOT MÓVIL

TESIS
Para obtener el grado de
INGENIERO EN REDES



PRESENTA
VÍCTOR ANTONIO RESENDIZ WONG

DIRECTOR
DR. JAIME SILVERIO ORTEGÓN AGUILAR



ASESORES
DR. VÍCTOR MANUEL SÁNCHEZ HUERTA
DR. HOMERO TORAL CRUZ
DR. FREDDY IGNACIO CHAN PUC
MSI. LAURA YÉSICA DÁVALOS CASTILLA



UNIVERSIDAD DE QUINTANA ROO
DIVISIÓN DE CIENCIAS E INGENIERÍA

Trabajo de tesis bajo la supervisión del comité del programa de licenciatura y aprobada como requisito para obtener el grado de:

INGENIERO EN REDES

Comité de Tesis

DIRECTOR:

DR. JAIME SILVERIO ORTEGÓN AGUILAR

SUPERVISOR:

DR. HOMERO TORAL CRUZ

SUPERVISOR:

DR. VÍCTOR MANUEL SÁNCHEZ HUERTA

UNIVERSIDAD DE
QUINTANA ROO
SERVICIOS ESCOLARES
TITULACIONES

CHETUMAL, QUINTANA ROO, MÉXICO, SEPTIEMBRE DE 2014.



UNIVERSIDAD DE QUINTANA ROO

División de Ciencias e Ingeniería

Agradecimientos

Agradezco a dios por la vida y la motivación que me ha llevado a crecer, a querer responder todas aquellas preguntas que me ha planteado la vida, así como las pruebas que me ha puesto en el camino y la fuerza para poder superarlas y forjarme como una persona más realizada.

Por la compañía y la formación de mis padres y hermanos que siempre me han apoyado en todo momento, que entre bromas buenas o malas siempre estuvieron para decirme las cosas de frente, para recordar mi posición, así como mis metas y que siempre esté al tanto de ellas para alcanzarlas.

Agradezco a mis compañeros que me demostraron que los verdaderos compañeros no son aquellos que te acompañan o solo están en las buenas sino aquellos que te motivan y te levantan para poder seguir adelante.

A los profesores que siempre estuvieron ahí que más que hablar solo de temas de la academia, incluso para interesarse en el bienestar propio de uno como estudiante y persona, aquellos profesores que establecían un vínculo de hermandad brindando su confianza y fortaleza así como el apoyo para no desistir ante nada.

Agradezco a mi tutor por su paciencia y fe en mí, por su carácter optimista, al igual que sus bromas y su interés de siempre tratar de motivarme y alentarme en esos días difíciles, por ser más que un tutor, un gran amigo que compartiera su confianza y alegría.

A los administrativos quienes me tuvieron paciencia para los trámites, que también fueron optimistas y amigables, así como a la Universidad que me dio la mejor experiencia como estudiante, una experiencia sana y divertida que siempre recordaré a los largo de los años, así como a los que están en ella, como la mejor experiencia de mi vida.

Para finalizar gracias a todas esas personas que estuvieron ahí, para hacer de mi vida lo que soy una persona con formación, que me llenaron de orgullo y me alentaron a ser como ellos o incluso mejores, porque son y siempre serán mi admiración, muchas gracias.



Dedicatoria

UNIVERSIDAD DE QUINTANA ROO

División de Ciencias e Ingeniería

Le dedico este trabajo a mi familia así como a mis profesores que me alentaron a querer hacer las cosas en grande, pero en especial le dedico este trabajo a mi abuelo Antonio Wong, que aunque no tuvo la oportunidad de verme crecer y poder compartir esto, agradeciendo los pocos pero grandes momentos que estuvo conmigo, las ideas asombrosas así como las charlas donde planeábamos un futuro juntos. Doy gracias a mi abuelo por escucharme incondicionalmente, por darle importancia a mi preparación y mis ideas, por creer en mí hasta sus últimos días. No hay palabras suficientes para agradecer todo el apoyo brindado y de corazón le dedico este trabajo como parte de la promesa hecha. Este trabajo es una prueba de cómo podemos lograr grandes cosas si tan solo tenemos el deseo y la determinación de lograrlos. *“Porque nuestros sueños pueden ser realizados mañana si nos lo proponemos”.*



UNIVERSIDAD DE QUINTANA ROO

División de Ciencias e Ingeniería

Resumen

En este documento se presenta un sistema de seguimiento de video. El cual consiste en una investigación del método aplicado, así como algunas aplicaciones de estos métodos en diferentes entornos y la aplicación de seguimiento.

Adicionalmente, se trabaja con un robot de tipo UGV o vehículo terrestre no tripulado, el cual proporciona los requerimientos necesarios para hacer funcionar este sistema de seguimiento de video. El método a utilizar es la segmentación de color debido a su fácil uso y sencillez en el manejo de un área controlada, así como un método de análisis radial el cual nos permite buscar curvatura entre las piernas de una persona.

Y como finalización se presenta una propuesta para complementar el proyecto de "sistema de seguridad de reconocimiento de múltiples objetos y asistencia inteligente", que requiere un subsistema de seguimiento de video para su funcionamiento en conjunto con otros sub proyectos.



UNIVERSIDAD DE QUINTANA ROO

División de Ciencias e Ingeniería

Contenido

CAPÍTULO 1.....	2
INTRODUCCIÓN.....	2
JUSTIFICACIÓN.....	4
OBJETIVO GENERAL.....	5
OBJETIVOS PARTICULARES.....	5
ALCANCE.....	5
CAPÍTULO 2.....	7
2.1 Video Tracking.....	7
2.1.1 Factores influyentes en el seguimiento de video.....	8
2.1.2 Componentes Principales.....	13
2.1.3 Interactividad contra la automatización del seguimiento.....	14
2.2 Aplicaciones.....	16
2.2.1 Producción de medios y la realidad aumentada.....	16
2.2.2 APLICACIONES MÉDICAS E INVESTIGACIÓN BIOLÓGICA.....	19
2.2.3 Vigilancia e inteligencia empresarial.....	21
2.2.4 Robótica y vehículos no tripulados.....	23
2.2.5 Tele-colaboración y juegos interactivos.....	24
2.2.6 Instalaciones de arte y actuaciones.....	26
2.3 APLICANDO EL SEGUIMIENTO DE VIDEO.....	27
2.3.1 Extracción de características.....	27
2.3.2 CARACTERÍSTICAS DE BAJO NIVEL.....	34
2.3.3 <i>CARACTERÍSTICAS nivel medio</i>	40
CAPÍTULO 3.....	46
3.1 Instalacion de Prerequisitos.....	50
3.1.2 <i>CONFIGURACION DEL OPENCV</i>	71
3.1 IMPLEMENTACION DEL CODIGO.....	51



UNIVERSIDAD DE QUINTANA ROO

División de Ciencias e Ingeniería

CAPÍTULO 4.....	61
RESULTADOS EXPERIMENTALES	61
CAPÍTULO 5.....	66
CONCLUSIONES	66
REFERENCIAS BIBLIOGRÁFICAS.....	68
ANEXO B	71
ANEXO B	78



UNIVERSIDAD DE QUINTANA ROO

División de Ciencias e Ingeniería

Tabla de Imágenes

FIGURA 2. 1 EJEMPLOS DE DESORDEN DE APARIENCIA O COLOR.....	8
FIGURA 2. 2 EJEMPLOS DE OBJETIVOS CON CAMBIOS DE APARIENCIA.	10
FIGURA 2. 3 MAPA CONCEPTUAL DE DESAFÍOS.	12
FIGURA 2. 4 EJEMPLO DE UN RASTREADOR DE CÁMARA.	17
FIGURA 2. 5 EJEMPLOS DE FRAGMENTOS DE MOVIMIENTO Y LA COLOCACIÓN DE OBJETOS EN UNA ESCENA DINÁMICA.	17
FIGURA 2. 6 CAPTURA DE MOVIMIENTO CON SISTEMA BASADO EN EL MARCADOR.	18
FIGURA 2. 7 SEGUIMIENTO DEL VIDEO APLICADO A LA PRODUCCIÓN DE LOS MEDIOS DE COMUNICACIÓN. ...	18
FIGURA 2. 8 VIDEO DE SEGUIMIENTO PARA APLICACIONES MÉDICAS Y DEPORTIVAS.	20
FIGURA 2. 9 VIDEO DE SEGUIMIENTO PARA LA INVESTIGACIÓN MÉDICA.....	20
FIGURA 2. 10 EJEMPLOS DE SEGUIMIENTO DE OBJETOS EN APLICACIONES DE VIGILANCIA.	22
FIGURA 2. 11 SEGUIMIENTO DE VIDEO PARA APLICACIONES COMERCIALES INTELIGENTES INTELLIVID.	22
FIGURA 2. 12 EJEMPLO DE SEGUIMIENTO DE OBJETOS A PARTIR DE UN VEHÍCULO AÉREO NO TRIPULADO. .	23
FIGURA 2. 13 EJEMPLOS DE LAS INTERFACES GESTUALES PARA JUEGOS INTERACTIVOS.	25
FIGURA 2. 14 ALGUNOS EJEMPLOS DE SEGUIMIENTO DE VIDEO APLICADOS A INSTALACIONES DE ARTE INTERACTIVO.....	26
FIGURA 2. 15 IMAGEN DE DETECCIÓN PARA EL SEGUIMIENTO DE VIDEO.	28
FIGURA 2. 16 IMAGEN ORIGINAL UTILIZADA PARA LAS COMPARACIONES DE ESPACIO DE COLOR.	35
FIGURA 2. 17 LOS TRES COMPONENTES DEL ESPACIO DE COLOR XYZ EXTRAÍDO DE LA IMAGEN 2.16.....	36
FIGURA 2. 18 LOS TRES COMPONENTES DEL ESPACIO DE COLOR RGB EXTRAÍDO DE LA IMAGEN 2. 16.....	36
FIGURA 2. 19 MODELO HSV EN FORMA DE PIRÁMIDE REGULAR HEXAGONAL INVERTIDA.....	38
FIGURA 2. 20 LOS TRES COMPONENTES DEL ESPACIO DE COLOR HSV EXTRAÍDOS DE LA FIGURA 2. 16.....	38
FIGURA 2. 21 EJEMPLO DEL MODELO DE DILATACIÓN	40
FIGURA 2. 22 EJEMPLO DEL MODELO DE DILATACIÓN.	42
FIGURA 2. 23 EXTRACCIÓN DE BORDES.	44
FIGURA 3. 1 EQUIPO DESKTOP DC7700.	47
FIGURA 3. 2 DR. ROBOT X80SVP.....	49
FIGURA 3. 3 PARTES DE DR. ROBOT X80SVP.....	49
FIGURA 3. 4 ESQUEMA GENERAL DE CONECTIVIDAD.....	50
FIGURA 3. 5 APLICACIÓN INTERMEDIA PARA CONECTIVIDAD.....	51
FIGURA 3. 6 DIAGRAMA DE PROCESOS	53
FIGURA 3. 7 FILTRADO CON COLOR AZUL MEZCLILLA.	54
FIGURA 3. 8 FILTRADO CON COLOR NARANJA.	54



UNIVERSIDAD DE QUINTANA ROO

División de Ciencias e Ingeniería

FIGURA 3. 9 FILTRADO CON COLOR AZUL MARINO.....	55
FIGURA 3. 10 FILTRADO CON COLOR ROJO.....	55
FIGURA 4. 1 HISTORIAL DE SEGUIMIENTO EN LA NOCHE.....	61
FIGURA 4. 2 COMPLEJIDAD DE RADIO.....	62
FIGURA 4. 3 FILTRADO AZUL 1PM.....	63
FIGURA 4. 4 FILTRADO EN COLOR ROJO.....	64
ANEXO A. 1 NUEVO PROYECTO.....	71
ANEXO A. 2 AGREGANDO REFERENCIAS.....	72
ANEXO A. 3 ELEMENTOS AGREGADOS.....	73
ANEXO B. 1 APLICACIÓN FINAL.....	74
ANEXO B. 2 CÁMARA CONECTADA.....	75
ANEXO B. 3 IMAGEN BINARIA.....	75
ANEXO B. 4 PANTALLA HSV.....	76
ANEXO B. 5 INTERFAZ GATEWAY.....	77

CAPÍTULO 1

CAPÍTULO 1

INTRODUCCIÓN

El manejo del video es cada día más fácil ya que las máquinas se ven más involucradas en entornos que son capaces de entenderse, dando apertura en diversas áreas como la vigilancia, creación de contenidos, comunicaciones personales, robótica incluso la interacción hombre-máquina.

El proceso de estimación en el tiempo y la ubicación de uno o más objetos utilizando una cámara se conoce como el seguimiento de video.

Una de las características fundamentales de las máquinas para ver, entender y reaccionar con el medio ambiente es su capacidad para detectar y rastrear objetos. Gracias a la rápida mejoría tanto en la calidad y la resolución de los sensores de imagen y el aumento en la potencia de cálculo de la última década, han favorecido la creación de nuevos algoritmos y aplicaciones.(Maggio & Cavallaro, 2011)

En la actualidad el objetivo de interés depende de la aplicación específica a requerir. Por ejemplo, en una aplicación de vigilancia de los edificios, los objetivos pueden ser personas, mientras que en otros, los juegos interactivos pueden formar parte de los objetivos utilizando las manos o la cara de una persona para un centro recreativo o de arte.

Otro ejemplo puede ser situado en el área de video vigilancia inteligente, donde por medio de cámaras situadas en establecimientos o áreas se puede predecir la salida o entrada de algún vehículo así como posibles estadísticas para aplicar métodos de seguridad y monitoreo dando mejores resultados en el ámbito empresarial donde la seguridad puede ser un punto clave para la empresa. Por si fuera si fuera poco esto ha llevado incluso a la aplicación del Marketing donde por medio de analistas en tiendas y sucursales mediante la ubicación de la persona y los productos, pueden rastrear su ruta y así obteniendo una estadística de sus movimientos, productos analizados y de posible interés. (IBM, 2014)

El seguimiento de video ya no solo se ha vuelto una realidad aplicada para el ámbito empresarial o áreas de videovigilancia, estadísticas de comercio y el área médica, así abarcando la aplicación de los vehículos autónomos para realizar tareas tediosas y arriesgadas logrando así ser sustitutos rentables para sus contrapartes humanas. Basándose en cierto entorno de trabajo, una clasificación aproximada de los vehículos autónomos incluye vehículos aéreos no tripulados (UAVs¹), vehículos terrestres no tripulados (UGVs²), vehículos submarinos autónomos (AUVs³), y vehículos de superficie autónomos (ASVs⁴). UAVs¹, UGVs², AUVs³, y ASVs⁴ conocidos como vehículos no tripulados (UVs⁵) en la actualidad. Esto ha provocado que en las últimas décadas, el desarrollo de vehículos autónomos no tripulados haya aumentado, provocando que diferentes tipos de vehículos autónomos se hayan estudiado y desarrollado por todo el mundo.

En particular esto ha resultado muy favorable para la sociedad en razón que los UAVs¹ tienen muchas aplicaciones en situaciones de emergencia; por ejemplo en algunos casos los seres humanos a menudo no podrían acercarse a un desastre natural peligroso, como un terremoto, una inundación, un volcán activo, o un desastre nuclear; en su lugar se emplea ésta novedosa tecnología. Desde el desarrollo de los primeros vehículos aéreos no tripulados, los esfuerzos de investigación se han centrado incluso en aplicaciones militares. (Kenzo, Farid, Satoshi, Wei, & Daisuke, 2010)

¹Unmanned Aerial Vehicles

² Unmanned Ground Vehicles

³ Autonomous Underwater Vehicles

⁴ Autonomous Surface Vehicles

⁵Unmanned Vehicles

JUSTIFICACIÓN

El desarrollo de un sistema de extracción de características en imágenes es un método ideal para la identificación de personas. Gracias al avance en tecnología es posible identificar a individuos en grandes multitudes; también puede ser considerado como un método natural dentro de los sistemas de reconocimiento biométricos, aplicada en dispositivos de área de seguridad, guía asistida o soporte empresarial, manejo de vehículos no tripulados, así como en el área médica, artística, etc.

En cuanto al presente proyecto el sistema de reconocimiento de objetos que utilizaré se enfocará en el área de vigilancia y seguimiento de personas, lo cual resulta útil ya que no solo dará las bases de un reconocimiento inteligente, sino que se realizará la manipulación de un robot capaz de poder interactuar en un entorno controlado, iniciando tareas de navegación robótica.

OBJETIVO GENERAL

Realizar un sistema de segmentación y seguimiento de objetos, empleando las bases robóticas para extender el área de seguimiento.

OBJETIVOS PARTICULARES

- ⌚ Conocer el robot y analizar sus dispositivos, así como sus librerías.
- ⌚ Iniciar el estudio del área de reconocimiento de Objetos y analizar el método a utilizar.
- ⌚ Programar la aplicación para la segmentación de color.
- ⌚ Determinar puntos líneas y texturas para el reconocimiento del objetos
- ⌚ Realizar el reconocimiento con la cámara del robot y delimitar los movimientos del mismo

ALCANCE

Debido al tamaño de este proyecto se dividirá en los siguientes subproyectos:

- “Reconocimiento de Objetos en un área controlada”,
- “Guía asistida por parte del robot”
- “Sistema de seguridad inteligente para áreas programadas”.

Este trabajo de tesis se enfoca únicamente al área de “Reconocimiento de Objetos en un área controlada”, dejando los demás para futuros proyectos; en particular la unificación en un “sistema de seguridad y asistencia inteligente”.

CAPÍTULO 2

CAPÍTULO 2

En esta investigación se presenta brevemente términos y elementos acerca del seguimiento del video, así como las diferentes partes que lo conforman, algunas de sus aplicaciones y características de los dispositivos del robot a manejar.

2.1 VIDEO TRACKING

Para diseñar un seguidor de video, las cámaras de video necesitan capturar información, en la forma de un conjunto de píxeles conformando una imagen. Al modelar la relación entre la aparición del objetivo y sus valores del píxel es posible calcular el cambio de los píxeles del objeto. Un rastreador de video estima la ubicación del objeto en el tiempo, esta relación entre un objeto y su proyección, puede depender de factores como la posición, sombras o incluso luminosidad haciendo del rastreo una tarea compleja, a esto se le conoce como seguimiento de video.

2.1.1 FACTORES INFLUYENTES EN EL SEGUIMIENTO DE VIDEO

Un rastreador para el seguimiento está relacionado principalmente con la apariencia del objetivo, influyendo así con la apariencia o variaciones del mismo con la similitud de otros objetos como se muestra en la figura 2.1.



Figura 2. 1 Ejemplos de desorden de apariencia o color.

Por ejemplo, los objetos en el fondo de la figura 2.1 con cajas de color verde, comparten un color del lado izquierdo o la forma del lado derecho similar en cuanto a sus propiedades, tratando de distraer al perseguidor del objeto del objetivo deseado, los cuales se señalan con cajas de color verde. A la izquierda: la imagen representa el conjunto de datos de seguimiento de la cabeza *Birchfield*. Derecha: Vigilancia del escenario de conjunto de datos PETS-2001.

Al igual que la aparición de otros objetos o del fondo puede ser similar al del objetivo, las características extraídas del área de la imagen no deseadas, pueden ser difíciles de diferenciar de las características que esperamos que el objetivo.

Este fenómeno se conoce como el desorden. En la figura 2.1 se puede apreciar un ejemplo de la ambigüedad que puede distraer aun rastreador del objetivo, este desafío puede ser tratado mediante el uso de múltiples métodos propuestos por su fiabilidad.

Además del desafío de seguimiento debido al desorden, se hizo seguimiento de video difícil por los cambios de la apariencia de destino, en el cual el plano de la imagen se debe a uno o más de los siguientes factores:

- **Los cambios en la pose.** En objeto móvil varia su aspecto al realizar cambios desde el punto de vista observado como cuando se gira o incluso en casos de una persona se sienta o cambian de posición como en las figuras 2.2 incisos a y b.
- **Iluminación ambiente.** La dirección, la intensidad y el color de la ambiente son factores en que influye la luz. Por otra parte, los cambios en iluminación global son a menudo un desafío en los escenarios al aire libre. Por ejemplo, la luz ambiente cambia cuando las nubes tapan la luz del sol, así como los ángulos de iluminación y el tipo de superficie que puede ser más reflejante que otras, lo cual afecta el estado de distinción del objetivo influyendo en la lente del rastreador.
- **Ruido.** El proceso de captura introduce en la señal de imagen un cierto grado de ruido, que depende de la calidad del sensor. Este ruido a veces puede ser identificado como pequeñas variantes o pixeles que varían minuciosamente afectando así a diferenciar le fondo o los objetos de primer plano.



Figura 2. 2 Ejemplos de objetivos con cambios de apariencia.

❖ **Las oclusiones.** Estas pueden ser interpretadas cuando un objeto que se está siguiendo queda oculto detrás del fondo u otro objeto, el cual queda obstruido. Las oclusiones son generalmente debido a:

- Un objetivo en movimiento detrás de un objeto estático, tal como una columna, una pared, o un escritorio como se muestra en la figura 2.2(c).
- Otros objetos en movimiento oscureciendo la vista del objetivo como se muestra en la figura 2.2(d).

Para hacer frente a este reto, los diferentes enfoques se pueden aplicar dependiendo del nivel esperado de oclusión:

- Oclusiones parciales que afectan sólo a una pequeña porción del área

objetivo que puede ser tratada por el modelo de apariencia de destino o por el propio algoritmo de detección de objetivos. Donde la invariancia es una de las propiedades de algunos métodos de representación para hacer frente a las oclusiones. Además, la sustitución de una representación global con múltiples características codificadas para una pequeña región del objetivo puede aumentar la robustez de un seguidor de video.

-La información sobre la aparición de destino no es suficiente para hacer frente a las oclusiones totales. En este desafiante escenario, la continuidad del seguimiento solo se puede lograr a través de un razonamiento de orden superior, es decir, a través de métodos múltiples hipótesis, que predicen el seguimiento, a través de cierto periodo en el tiempo. Esta Información acerca de los comportamientos del movimiento y los patrones típicos de oclusión pre-existentes, también se puede utilizar para propagar la trayectoria del objeto al desaparecer. Cuando el objetivo vuelve a aparecer, la propagación de múltiples hipótesis de seguimiento y modelización de apariencia puede proporcionar las señales necesarias para reinicializar nuevamente la predicción del objetivo.

Un resumen de los principales retos en el seguimiento de video se presenta en la figura 2.3.

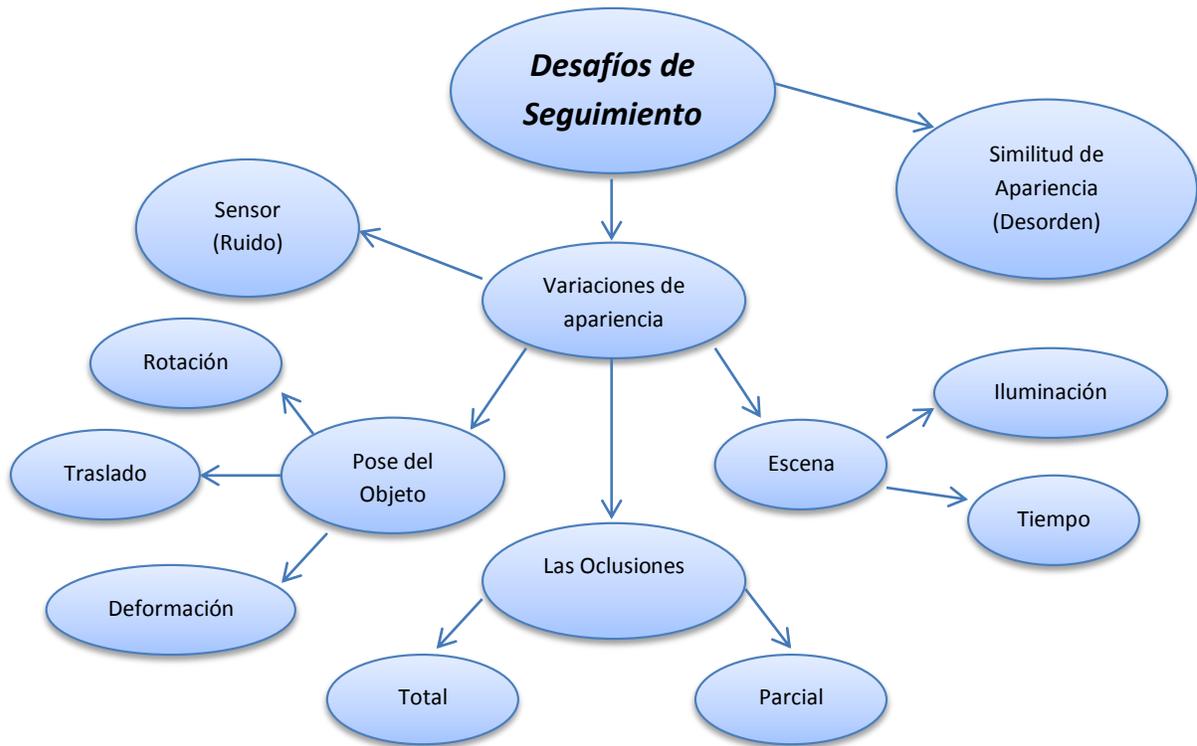


Figura 2. 3 Mapa conceptual de desafíos.

2.1.2 COMPONENTES PRINCIPALES

Para poder identificar y preparar un seguidor, se definen algunos componentes principales en el seguimiento de video o Video Traking que nos ayuden a poder aplicarnos en un área más específica y tener un mejor desempeño en la identificación de nuestros objetivos; en donde según Maggio y Cavallaro(2011) se señalan los siguientes componentes como parte fundamental del VideoTraking:

1. La definición de un método para extraer información relevante de una imagen en un área ocupada, por un objetivo puede basarse en clasificación de movimiento, detección de cambios, clasificación de objetos o la simplemente extracción de características de bajo nivel, como el color, el degradado o las características de nivel medio como bordes o puntos de interés.
2. Representación para la codificación de la apariencia y el estado: se define como las características del objetivo para ser usado por el rastreador. En general, la representaciones un compromiso entre la exactitud de la descripción y la invariancia, en donde debe ser lo suficientemente descriptivo para hacer frente con el desorden y descartar los falsos objetivos, al tiempo que permite un cierto grado de flexibilidad para hacer frente a los cambios de escala del objetivo, pose, iluminación y las oclusiones parciales.
3. La definición de un método para propagar el estado de la objetivo con el tiempo, se utiliza de forma recursiva para la información procedente de la etapa de extracción de características, o de las estimaciones estatales que ya están disponibles para formar una trayectoria. Esta tarea enlaza diferentes instancias del mismo objeto sobre tiempo, y tiene que compensar las oclusiones, el desorden local y global de cambios de iluminación.
4. La definición de una estrategia, para administrar los destinos que aparecen y desaparecen de una imagen, también se conoce como la administración de la trayectoria, inicializa el trayecto de un objeto entrante de interés y pone fin a la trayectoria asociada a un objetivo desaparecido. Cuando un nuevo objetivo aparece en la escena (el nacimiento del objetivo), el seguidor debe inicializar una nueva trayectoria. Un nacimiento de la trayectoria suele ocurrir:

- En los límites de la imagen (En el borde del campo de vista de la cámara).
- En las zonas de entrada específicos (Por ejemplo puertas).
- En el campo lejano de la cámara (Cuando el tamaño de la proyección sobre el plano de imagen aumenta y el objetivo se hace visible).
- Cuando un objeto se pone desde otro objetivo (Por ejemplo, un controlador de aparcamiento de un auto y luego cuando sale).

Del mismo modo, una trayectoria debe terminarse (fin de la trayectoria) cuando el objetivo:

- Deja el campo de visión de la cámara.
 - Desaparece a distancia o dentro de otro objeto (por ejemplo, un edificio).
5. La extracción de meta-datos, en una forma compacta y sin ambigüedades para ser utilizado por la aplicación específica, como la anotación de video, comprensión de escena y el reconocimiento del comportamiento.

2. 1.3 INTERACTIVIDAD CONTRA LA AUTOMATIZACIÓN DEL SEGUIMIENTO.

De acuerdo con Maggio y Cavallaron(2011) hay un límite entre la interactividad y la automatización en el seguimiento; el cual depende de la cantidad de interacción con los usuarios para identificar el objeto o los objetos de interés donde algunos algoritmos de video de seguimiento se pueden agrupar en tres categorías: *manual*, *interactiva (supervisada)* y *automático*.

- El seguimiento o tracking de videos se puede realizar directamente por el usuario, como un seguimiento manual para una alta precisión, por ejemplo, en la definición de los límites de los objetivos. Un ejemplo de aplicación que requiere seguimiento manual se encuentra en la producción de películas, cuando el contorno de un caracter está seleccionado y su evolución a

través del tiempo se define por un operador, encada imagen. Este procedimiento permite una buena definición de la posición (límites) orientar, pero es muy lento y no puede ser ampliamente utilizado para los grandes volúmenes de datos visuales.

- El seguimiento automatizado utiliza información a priori acerca de los objetivos que se codifica en un algoritmo. Por ejemplo, un detector de objetos se puede utilizar para inicializar el rastreador y / o para apoyar a la estimación de estado con el tiempo. Algunos ejemplos de métodos son basados en la detección de rostros y movimiento a objetos. Las técnicas de video-seguimiento para aplicaciones del mundo real son todavía muy recientes, puesto que las propiedades de la definición de un objetivo en criterios algorítmico es una tarea difícil en los escenarios no restringidos.
- Estrategias interactivas (semiautomáticos) se utilizan como un intercambio entre un perseguidor, un rastreador manual y uno automatizado. El principio en la base de técnicas semiautomatizadas es la interacción del usuario durante algunas etapas del proceso de seguimiento, donde la información sobre la zona de destino se proporciona directamente por el usuario, cuando el interactivo de seguimiento se utiliza en aplicaciones de etiquetas y pistas, (por ejemplo, edición de video y vigilancia) cuando un operador inicializa manualmente un objetivo de interés que es seguido por el algoritmo de seguimiento. Después que el usuario proporciona la definición inicial del objetivo, el rastreador sigue su evolución temporal en los siguientes fotogramas propagando así la información de inicialización. Esta segunda fase puede ser supervisada o bien sin supervisión, o el operador todavía puede verificar la calidad de la seguimiento resultados y corregirlos si es necesario. Esto permite, por ejemplo, editores de video o anotadores para seleccionar la apariencia de un objetivo y después propagar estos cambios a través del tiempo, ahorrando así la tarea que consume tiempo de editar manualmente cada fotograma. Otro ejemplo de una aplicación es el control de la cámara para la vigilancia, cuando una persona está seleccionado por un operador y luego seguido automáticamente con una inclinación y zoom (PTZ).(Maggio & Cavallaro, 2011)

2.2 APLICACIONES

Dentro del seguimiento de objetos se pueden realizar muchas aplicaciones, que van desde el área de la vigilancia a distancia, robótica, predicción para juegos interactivos, rastreadores de video para mejorar nuestra comprensión de grandes conjuntos de datos de video desde aplicaciones médicas y de seguridad nacional, marketing, entre otros. Con el seguimiento de objetos se aumenta la productividad mediante la reducción de la cantidad de trabajo manual para completar una tarea y permitir la interacción natural con la máquina, como por ejemplo en:

- Producción de medios y la realidad aumentada.
- Las aplicaciones médicas y la investigación biológica.
- Vigilancia e inteligencia empresarial.
- Robótica y vehículos no tripulados.
- Tele-colaboración y juegos interactivos.
- Instalaciones de arte y actuaciones.

2.2.1 PRODUCCIÓN DE MEDIOS Y LA REALIDAD AUMENTADA.

El seguimiento de video es un elemento importante en la post-producción y captura de movimiento para las industrias del cine y la televisión. Este es el argumento de las capturas con elementos gráficos de computadora y efectos especiales, que se representan en la película, con el fin de añadir constantemente estos nuevos elementos para las capturas. El procedimiento de representación requiere el conocimiento de la información 3D en la escena. Esta información puede ser estimada por una cámara rastreadora, que calcula el tiempo la posición de la cámara, la orientación y longitud focal. La estimación 3D se deriva a partir del análisis de un gran conjunto de trayectorias de la imagen en 2D, que a su vez contiene características salientes que identifica el algoritmo de seguimiento en las capturas (Triggs, McLauchlan, Hartley, & Fitzgibbon, 1999). Un ejemplo de parches de seguimiento y puntos se muestra en la figura 2.6, donde se utilizan trayectorias 2D de bajo nivel para estimar el nivel más alto Información 3D. La figura 2.7 muestra dos ejemplos de los fragmentos de movimiento donde el humo, efectos especiales y objetos adicionales como un barco y un edificio se añaden a

las escenas originales. Una aplicación relacionada es la colocación del producto virtual que incluye uno específico, que se anuncian en un video o se ajusta un logotipo o textura en torno a un objeto real capturado en la escena.



Figura 2. 4 Ejemplo de un rastreador de cámara.



Figura 2. 5 Ejemplos de fragmentos de movimiento y la colocación de objetos en una escena dinámica.

Otra tecnología basada en el seguimiento de video y utilizada por los medios de producción casera es la captura de movimiento, que se utilizan para animar personajes virtuales de seguimiento de los actores reales. Aunque el movimiento

sin captura de marcadores está recibiendo cada vez más atención, la mayoría de los sistemas de captura de movimiento rastrear un conjunto de marcadores unidos al cuerpo y las extremidades para estimar sus poses como en la figura 2.8, los cuales sistemas de captura de movimiento están especializados en recuperar los movimientos de actores reales en 3D. Donde a continuación, el movimiento de los fabricantes, es proyecto sobre personajes generados por los gráficos de computadora, el cual, el seguimiento de video se utiliza para el análisis y la valoración de los eventos. Como se muestra en el ejemplo de la figura 2.9, un algoritmo de seguimiento puede estimar la posición de los jugadores en el campo, a fin de recopilar estadísticas sobre un juego (por ejemplo, un partido de fútbol). Estadísticas y visualizaciones mejoradas ayudan a los comentaristas, entrenadores y simpatizantes en destacar las tácticas del equipo y rendimiento de los jugadores.



Figura 2. 6 Captura de movimiento con sistema basado en el marcador.



Figura 2. 7 Seguimiento del video aplicado a la producción de los medios de comunicación.

2.2.2 APLICACIONES MÉDICAS E INVESTIGACIÓN BIOLÓGICA

Describiendo la sección anterior, también se aplican estas herramientas para el análisis del movimiento humano, así como para mejorar el rendimiento en los atletas como en la figura 2.10 en los incisos a y b, donde también se evalúa el análisis de la marcha de un paciente, para el estado de las articulaciones y los huesos como en la figura 2.10 inciso c. En el seguimiento general video se ha utilizado cada vez más por los sistemas médicos para ayudar al diagnóstico y agilizar la tarea del operador. De acuerdo con Bardinet, Cohen y Ayache(1996) algunos ejemplos, son algoritmos automatizados de seguimiento ventricular en imágenes de ultrasonido (Toshiba) y por otra parte, el seguimiento de video puede estimar la posición de determinados tejidos blandos (Mountney & Yang) o de instrumentos como agujas y broncoscopios durante la cirugía.(Bardinet, Cohen, & Ayache, 1996)

En la investigación biológica el seguimiento del movimiento de los organismos no humanos permite analizar y comprender los efectos de los fármacos específicos o los efectos de envejecimiento (Geng, Cosman, Member, Berry, Feng, & Schafer, 2004). La figura 2.11 muestra dos ejemplos de aplicación que el seguimiento de video estimando la ubicación de las bacterias *Escherichiacoli* del lado izquierdo y *Caenorhabditiselegans gusanos* del lado derecho de la imagen.

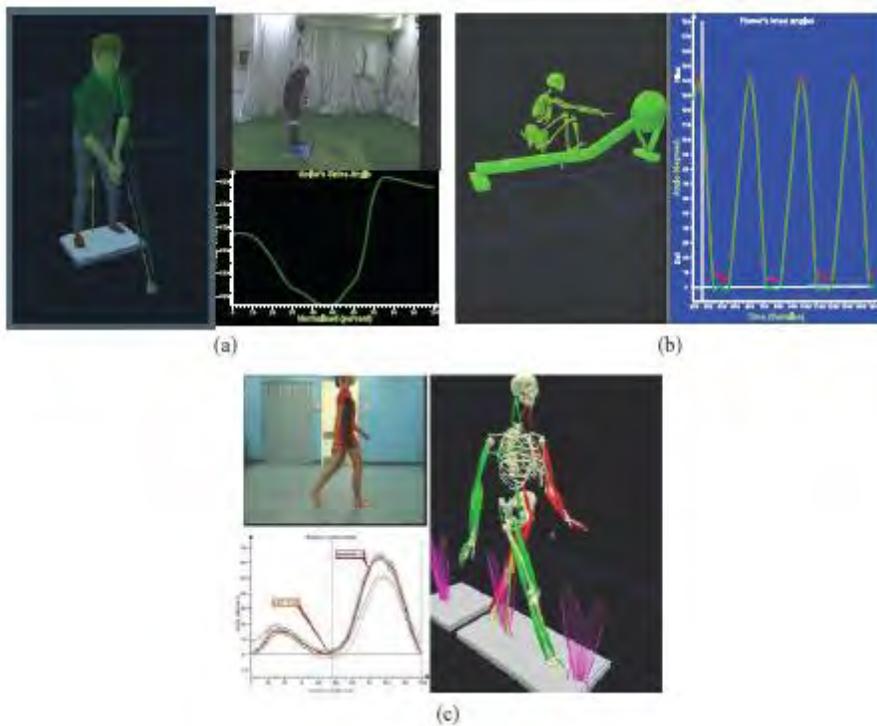


Figura 2. 8 Video de seguimiento para aplicaciones médicas y deportivas.

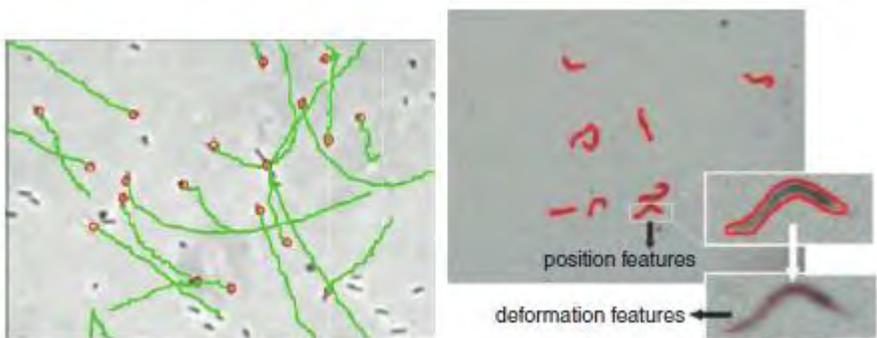


Figura 2. 9 Video de seguimiento para la investigación médica.

2.2.3 VIGILANCIA E INTELIGENCIA EMPRESARIAL

En el seguimiento de video según Anjum y Cavallaro(2008), también es considerada como una herramienta en el área de la vigilancia automatizada para aplicaciones de seguridad, vida asistida así como en la inteligencia de negocios. En la vigilancia sistemas, el seguimiento puede ser utilizado ya sea como una herramienta forense o como una etapa de procesamiento que clasifican los comportamientos. Por otra parte, el video de seguimiento como software combinado con otras herramientas de análisis se puede utilizar para redirigir la atención de los operadores humanos a eventos de interés. Sistemas de vigilancia inteligentes pueden ser desplegados en una variedad de diferentes interiores y ambientes al aire libre, tales como carreteras, aeropuertos, puertos, estaciones de ferrocarril, edificios públicos y privados por ejemplo, escuelas, bancos y casinos (Ajum & Cavallaro, 2008). Ejemplos de sistemas de video vigilancia se pueden apreciar en la figura 2.12 los cuales son el Sistema IBM Smart Surveillance (Tian, Brown, Hampapur, Lu, Senior, & Shu), la plataforma inteligente de Video General Electrics Vision Wave(General Electric) y objetos de video VEW(Object Video).

El seguimiento de video también puede servir como una observación y una herramienta de medición en entornos comerciales (por ejemplo, en la inteligencia minorista), tales como supermercados, donde la posición de los clientes lleva a cabo un seguimiento en el tiempo(INTELLIVID) como en la figura 2.13. Combinado datos de trayectorias con la información del punto de ventas con la caja registradora y así construyendo modelos de conducta que describen donde los clientes pasan más tiempo en la tienda, la interacción con los productos dependiendo de su ubicación, y los elementos que compran. Mediante el análisis de esta información, el equipo de marketing puede mejorar la colocación de productos en el espacio comercial. Por otra parte, con la mirada de seguimiento frente a las carteleras se puede utilizar para seleccionar automáticamente el tipo de anuncio para mostrar u para cambiar dinámicamente dependiendo del contenido en función de la atención o la comercialización estimada del perfil de la persona, basado por ejemplo en el género y la edad estimada.

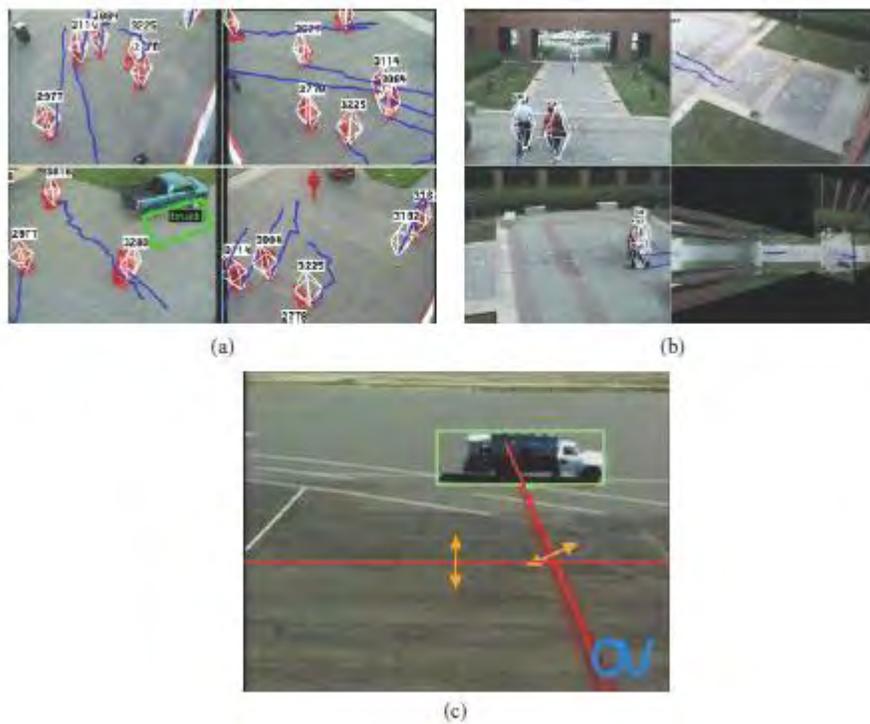


Figura 2. 10 Ejemplos de seguimiento de objetos en aplicaciones de vigilancia.



Figura 2. 11 Seguimiento de video para aplicaciones comerciales inteligentes IntelliVid.

2.2.4 ROBÓTICA Y VEHÍCULOS NO TRIPULADOS.

Otra área de aplicación que se utiliza extensivamente son los algoritmos de video de seguimiento en la robótica. La robótica incluye el desarrollo de robots humanoides, con cámaras PTZ automatizadas conocidas como pan-tilt-zoom y vehículos aéreos no tripulados o también llamados UAVs. La visión inteligente a través de una o más cámaras montadas en los robots, proporciona información o necesaria para interactuar con o navegar en el ambiente. También el medio ambiente exploración y cartografía(Davison, Reid, Molton, & Stasse), así como la interacción humano-robot a través de gesto reconocimiento dependen de video de seguimiento(Wu & Huang, 1999).

El problema de estimar el movimiento mundial de robots y vehículos no tripulados se relaciona con el problema de la cámara de seguimiento. Mientras que los algoritmos de rastreo para la producción de los medios de comunicación pueden ser aplicados fuera de línea, video rastreadores para la robótica necesitan localizar de forma simultánea en tiempo real la posición del robot (es decir, de la cámara) y para generar un mapa del entorno. Dicha información de localización en 3D se genera mediante el seguimiento de la posición destacado características de la imagen tales como esquinas y bordes(Fitzgibbon & Zisserman, 1998), como se muestra en la figura 2.14.

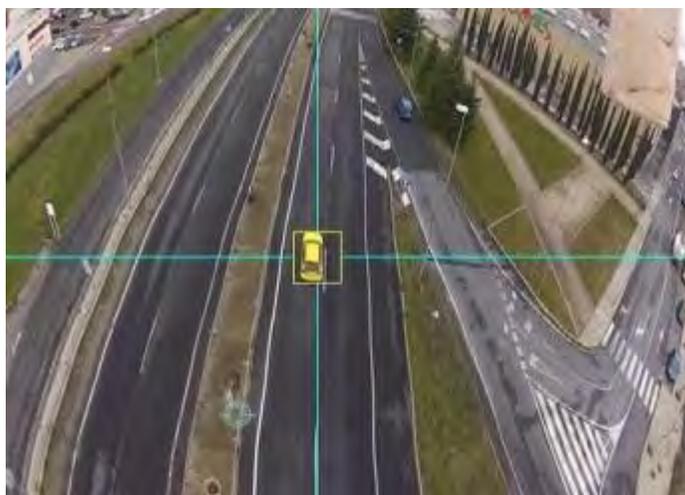


Figura 2. 12 Ejemplo de seguimiento de objetos a partir de un vehículo aéreo no tripulado.

La información sobre la posición en 3D también se utiliza para generar una malla en 3D que se aproxima a la estructura de los objetos que lo rodean y el medio ambiente. En particular, los UAVs hacen un amplio uso del seguimiento de video para encontrar la posición de concreto objetos en el suelo como se muestra en la figura 2.14, así como para permitir el aterrizaje automático.

2.2.5 TELE-COLABORACIÓN Y JUEGOS INTERACTIVOS.

De acuerdo a Freeman(1998) las webcams estándar ya administran el software de seguimiento la cual rastrea la cara de un usuario para videoconferencias. Por otra parte, el seguimiento basado en video se utiliza para simular el contacto visual entre los asistentes de una reunión para mejorar la eficacia de la interacción en la videoconferencia (Gemmell, Toyama, Zitnick, Kang, & Seitz, 2000). Dicha tecnología se aplica en aulas, que dispone de un conjunto de cámaras PTZ (pan-tilt-zoom) para seguir la posición del profesor(Huanf & Trivedi, 2003). Las cámaras PTZ (pan-tilt-zoom) explotan la información sobre la trayectoria en tiempo real para guiar a los parámetros de panorámica, inclinación y zoom de la cámara. Para mejorar la precisión de seguimiento, la información de una formación de micrófonos también puede estar resumida con la información de la cámara(Freeman, y otros, 1998).

Dicho seguimiento de video también está cambiando la manera en que enviamos el control de las máquinas. Esta modalidad de interacción natural está siendo utilizada en juegos interactivos, por ejemplo, la acción de pulsar un botón en un dispositivo se sustituye por un conjunto gestos más intuitivos realizados por el usuario en frente de la cámara como se muestra en la figura 2.15. Del mismo modo, en los juegos generalizados, donde la experiencia se extiende con el mundo físico, el seguimiento basado en visión refina los datos de posición de la Sistema de Posicionamiento Global o GPS.



Figura 2. 13 Ejemplos de las interfaces gestuales para juegos interactivos.

2.2.6 INSTALACIONES DE ARTE Y ACTUACIONES.

Seguimiento de video es cada vez más utilizado en las instalaciones de arte y actuaciones donde la interacción está habilitada por el uso de cámaras de video y, a menudo por sistemas de proyección. La interactividad se puede utilizar para mejorar la narrativa de una pieza o para crear acciones inesperadas o reacciones del medio ambiente como se muestra en la figura 2.16.

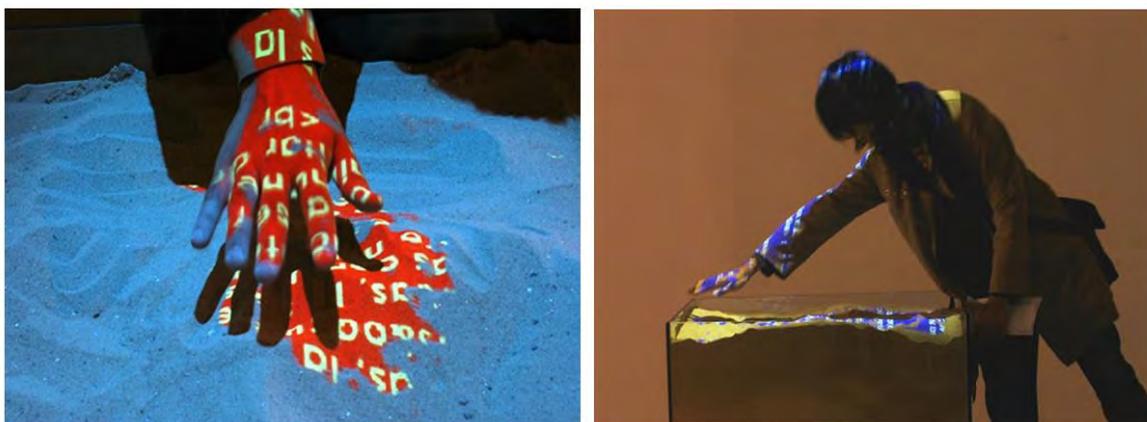


Figura 2. 14 Algunos ejemplos de seguimiento de video aplicados a instalaciones de arte interactivo.

Por ejemplo, la tecnología de seguimiento permite la interacción entre los visitantes del museo e instalaciones visuales como se muestra del lado derecho de la figura 2.16. También, alguien en un grupo puede ser detectado selectivamente y luego seguido a través del tiempo, mientras que una luz 'animada', se proyecta junto a la persona seleccionada como se muestra en la figura 2.16 del lado izquierdo. El arte interactivo basado en el seguimiento de video también puede permitir nuevas formas de comunicación entre personas distantes. Por ejemplo, la posición relativa de un objeto rastreado y un cuerpo humano pueden conducir un conjunto de efectos de iluminación (Stenslie, 2010).

2.3 APLICANDO EL SEGUIMIENTO DE VIDEO

Dentro de lo que lleva acabo el seguimiento se tiene que tomar en cuenta ciertas características o factores influyentes del entorno o del dispositivo para decidir el método a utilizar, brevemente se explicaran algunos de los métodos a retomar y se presentan los métodos aplicados en este trabajo.

2.3.1 EXTRACCIÓN DE CARACTERÍSTICAS

El rendimiento de un seguidor de video depende de la calidad de la información que podemos extraer de las imágenes. Para entender cómo aprovechar mejor la información de la imagen, resumiremos brevemente la formación de la imagen a procesar y discutiremos los métodos para extraer características que son importantes para la desambiguación de los objetos de interés:

- De la escena de fondo.
- De otros objetos.

La extracción de características es el primer paso en el seguimiento y nos permite resaltar la información de interés de las imágenes para representar un objetivo. Dichas características para ser extraídos se pueden agrupar en tres categorías principales:

- De bajo nivel (por ejemplo, color, gradiente de movimiento).
- De nivel medio (por ejemplo, los bordes, las esquinas, las regiones).
- De alto nivel (objetos).

Para ello hay que extraer las características principales en el seguimiento de video, donde la información capturada es explotada por un sensor de imagen, y generar una descripción compacta de la localización de los objetos de interés para ser utilizado en la aplicación deseada.

2.3.1.1 MEDICIÓN DE LUZ

Según Bengtsoon(2008) un sensor de imagen se compone de un conjunto de elementos fotosensibles, pixeles. Cuando la luz incide en un elemento fotosensible, debido al efecto fotoeléctrico, el elemento cambia de estado. Este cambio es proporcional al número de fotones (es decir, la intensidad de la luz) que golpea el elemento.

El elemento fotosensible funciona como un condensador. Podemos obtener una estimación cuantitativa de la intensidad de la luz mediante la medición de la tensión en la salida (pins) del elemento fotográfico. Por ejemplo, en sensores de nivel de gris, cuando la tensión es cero (es decir, no hay fotones) se representa con el color negro, mientras que cuando la tensión máxima se produce como un consecuencia de muchos fotones que golpean el sensor, se torna blanco.

Los avances en la tecnología de silicio permiten la integración de millones de pixeles en un único microchip donde los pixeles forman una matriz bidimensional como se muestra en la figura 2.17, donde la luz reflejada por los objetos dentro del campo de visión de la cámara se convierte en información digital, que se utiliza como entrada para el rastreador. El número de pixeles de un sensor también se conoce como su resolución.

Debido al pequeño tamaño del sensor (por lo general de 1 a 10 milímetros), se necesita una lente para concentrar la luz en la matriz de manera que cada pixel codifica la información de diferentes direcciones dentro del ángulo de visión de la lente.

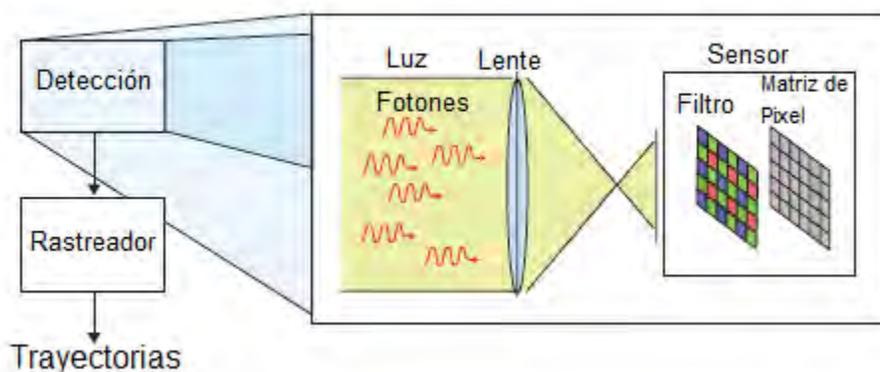


Figura 2. 15 Imagen de detección para el seguimiento de video.

1.- **El sensor de color:** La adquisición de datos en Color sigue un procedimiento similar al descrito anteriormente. Sin embargo, existen diferencias entre los dispositivos que utilizan uno o varios sensores:

- En dispositivos de color que utilizan un solo sensor, el filtrado se basa en los diferentes filtros se colocan en frente de cada pixel, como se muestra en la figura 2.17 para formar una matriz de filtros que dejan pasar la luz dentro de un rango de frecuencias. Generalmente, el arreglo se compone de una matriz de tres filtros diferentes que son sensibles a bandas de frecuencia. Después del filtrado, la salida del pixel se ordena para generar una representación virtual o conversión analógico-digital, donde cada pixel de color está compuesta de una combinación de tres colores correspondiente al rojo, al verde y al azul. Aunque los dispositivos sensores son más baratos que los dispositivos de sensores múltiples, más de 60% de la información se pierde durante el proceso de filtrado, resultando así en un rendimiento insuficiente, en términos de calidad de imagen.
- Los dispositivos de color que utilizan múltiples sensores aprovechan mejor los fotones que pasan a través de la lente. El conjunto de filtros es utilizado en los dispositivos de un solo sensor que está constituido por un prisma que divide los fotones de acuerdo con su frecuencia (es decir, su color asociado), formando de este modo múltiples haces de luz. Cada haz está dirigido hacia un sensor separado. La cámara del dispositivo de acoplado de tres cargos o multi-sensor, utiliza un prisma *Trichroic* o de tres vigas las cuales se combinan con los tres sensores. A diferencia de los dispositivos de un solo sensor, en una de sensores múltiples no es necesario la interpolación de información de color y, por tanto, la resolución es efectiva e igual a la resolución física de cada sensor.

Una gran variedad de sensores de color muestran diferencias en términos de bandas de filtro de color, la densidad de pixeles por banda y la disposición de pixeles. Además, los sensores pueden funcionar en bandas de frecuencia más allá del espectro visible (por ejemplo, cámaras de infrarrojos) o a través de múltiples bandas de frecuencia del espectro electromagnético es decir MultiSpec central y cámaras híper espectrales(Bengtsson, 2008).

2.- **A partir de imágenes de videos:** Las cámaras de video digitales activan el obturador electrónico múltiples veces por segundo, generando así una secuencia de imágenes que representan una escena en diferentes instantes de tiempo. El número de activaciones por segundo define la velocidad de fotogramas de la cámara. La velocidad de fotogramas varía desde unos pocos fotogramas por

segundo a cientos o miles de fotogramas para las cámaras de propósito especial. (vicon, 2013)

Desde una perspectiva de video de seguimiento, la tasa de cuadro de la cámara es un parámetro muy importante: puesto que a mayor velocidad de fotogramas se facilita el seguimiento, el objeto a igual velocidad, la magnitud del movimiento entre dos marcos es más pequeña. Por lo tanto, los algoritmos más simples pero más rápidos pueden correr a altas velocidades de fotogramas y pueden funcionar mejor que las elaboradas. Además, una velocidad de obturación rápida se utiliza con cámaras de alta velocidad de fotograma ya que reducen el desenfoque de movimiento, lo que mejora la nitidez de las imágenes con objetos en movimiento rápido.

3.- Representando el video: Si hacemos caso omiso de los efectos cuánticos debido a la naturaleza discreta de la luz, la tensión sigue siendo una señal analógica que abarca un rango entre cero y su valor máximo. Para obtener una señal digital, un convertidor analógico-digital codifica el voltaje en un conjunto finito de números binarios. Por ejemplo, en muchas cámaras comerciales, la tensión se convierte en un número entero sin signo de 8 bits en el intervalo de 0-255. La colección de bits que representa una imagen se refiere a menudo como una imagen en bruto o sin comprimir. Del mismo modo, el video en bruto es la unión de las imágenes en bruto que así mismo representan las tramas.

El número de bits necesarios para representar una imagen cruda depende de dos factores: la resolución de la imagen y la profundidad de color, medida en bits por pixel. La profundidad de color depende del número de niveles de cuantificación. Además de estos dos factores, en un video sin procesar el número de bits por segundo también depende de la velocidad de fotogramas.

El ancho de banda y espacio de almacenamiento requerido por el video en bruto crecen rápidamente con el aumento de la resolución del sensor y los niveles de cuantificación, en consecuencia, siendo menos práctico en muchas aplicaciones y las técnicas de compresión de video. Ciertos codificadores de video tienen como objetivo reducir el número de bits necesarios para representar un video en bruto por explotar las redundancias espaciales y temporales entre los pixeles vecinos de la misma imagen y a través de otras imágenes. Además, los codificadores de video tienen en cuenta factores humanos para descartar datos perceptivamente irrelevantes o menos importantes con el fin de aumentar la relación de compresión global COM (Modelo de Componentes de Objetos). Como la corriente principal los codificadores (Gall, 1991) basados en bloques, que operan pixeles en una imagen separada, así como las discontinuidades entre los límites de bloques pueden ser relevantes en altas relaciones de compresión. Estos

dispositivos de bloqueo cambian artificialmente las funciones dentro de una imagen, lo que complica la tarea de seguimiento de video. Por otra parte, los codificadores con pérdidas descartan información que, aunque menos relevante, puede ser importante para el seguimiento de video. Por estas razones, se debe tomar un cuidado especial cuando los dispositivos de compresión se introducen en un video antes del seguimiento.

Existe un vínculo interesante entre los codificadores de video y algoritmos que realizan el seguimiento en el dominio comprimido. El seguimiento en el dominio comprimido se prefiere cuando la potencia de cálculo es limitada, en cambio en Rastreadores-dominio comprimido se pueden utilizar algunos codificadores para localizar objetos en la escena. Por ejemplo, la fase en el dominio de la frecuencia se asocia con cambios de señal, y, posiblemente de movimiento, en el dominio espacial. Por otra parte, los vectores de movimiento producidos por técnicas de correlación temporal se pueden usar para seguimiento de objetivos (Fonseca & Nesvadba, 2006).

2.3.1.2 LA APARIENCIA DEL OBJETIVO

La apariencia del color de los objetos, depende de tres componentes:

- Una fuente de energía electromagnética visible.
- El objeto en sí mismo, cuya superficie modula las propiedades de la energía electromagnética.
- Un dispositivo de captura, tal como una cámara.

La combinación de estos tres elementos hace que el color sea difícil de manejar. La comprensión de la física de la formación de la luz, la reflexión y la imagen es un requisito para la extracción de características eficaz del destino.

1.- **Modelos de Reflexión:** Los modelos de reflexión describen las propiedades espaciales y cromáticas de la luz reflejada por una superficie. Ejemplos de modelos de reflexión son el modelo de sombreado Phong y el modelo de reflexión dichromatic.

El modelo de Phong(Phong, 1975) es un modelo de iluminación local empírica, que trata de fuentes de luz y tres tipos de modelos de luz reflejada:

- *Reflexión de Ambiente*, modela la reflexión de la luz que llega al objeto de todas las direcciones, después de haber sido rebotada de la entorno en múltiples reflexiones de todas las superficies de la entorno. Como veremos más adelante, la reflexión ambiental puede tener una fuerte influencia en el color percibido de un objetivo.
- *Reflexión difusa*, modela el reflejo de las superficies no brillantes que dispersan la luz en todas las direcciones por igual.
- *La reflexión especular*, modela la reflexión de la luz en superficies como espejos. El especular refleja la intensidad representada en los aspectos más destacados, el reflejo brillante de los objetos, que es el resultado de la iluminación incidente que rebota en la superficie. Destacando en la superficie de un blanco y el gradiente de las propiedades locales.

La intensidad de la luz reflejada desde la superficie domina la apariencia del objetivo. La intensidad del color en la superficie de un objetivo es la suma de las intensidades de cada uno de los tres tipos de reflexiones.

El modelo de reflexión dicromático (Shafer, 1985) describe a la luz que se refleja desde un punto en un material dieléctrico no uniforme como una combinación lineal (mezcla) de los siguientes puntos:

- La luz reflejada en la superficie del material (Reflexión de la superficie).
- La luz reflejada por el cuerpo material (Cuerpo de reflexión).

La reflexión de superficie representa los aspectos más destacados, como el reflejo brillante de los objetos. Por lo general, tiene aproximadamente la misma distribución de potencia espectral como la iluminación.

La luz que no se refleja en la superficie penetra en el cuerpo material en el que se dispersa y absorbe selectivamente. Algunas fracciones de luz llegan de nuevo a la superficie y sale del material. La luz que viaja a través del cuerpo se absorbe cada vez más en las longitudes de onda que son característicos para el material. La reflexión del cuerpo proporciona el color de referencia característico y a su vez la luz reflejada que por lo general se difunde por igual en todas las direcciones.

2.- Variaciones de iluminación locales: Son modelos de color que distinguen a los cambios de color en los límites materiales de cambio de color, debido a las variaciones de iluminación locales, como sombras y luces, las cuales son deseables para identificar blancos en una imagen, como delimitado las fronteras materiales.

En particular, las sombras pueden modificar la forma de destino percibido. Una sombra se produce cuando un objeto se oculta parcial o totalmente de la luz directa de una fuente de iluminación. Una sombra es visible cuando la posición de la cámara no coincide con la posición de la fuente de luz. Las sombras se pueden dividir en dos clases:

- Auto sombras, que se producen en la parte de un objetivo que no está iluminado directamente.
- Sombras proyectadas, que son el área proyectada por el objetivo en la dirección directa de la luz.

Se debe tener en cuenta que las características espectrales de la luz ambiente pueden ser muy diferentes a las de la luz directa. Por ejemplo, si hay otro objeto (u objetos) inesperado influyen en el color del objetivo esperado, este último esté expuesto a la iluminación ambiente que no tiene las mismas características espectrales como la iluminación incidente. Este caso se conoce como inter-reflexión. Estas influencias adicionales de iluminación ambiental, se conocen como encendidas o las regiones de sombra, donde, el color de las dos regiones puede

ser muy diferente. Esto haría que la tarea de identificarla región de sombra y la región iluminada como parte del mismo objetivo se convierta en una tarea complicada.

2.3.2 CARACTERÍSTICAS DE BAJO NIVEL

Dentro de las características de bajo nivel, se tiene propuesto aplicar la extracción de color RGB y luego la conversión a HSV en donde se aplicará un filtrado de color. Este documento solo explicará las partes definidas de RGB y HSV como parte del propósito de la investigación.

2.3.2.1 COLOR

De acuerdo a Fairchild existe una amplia cantidad de representaciones matemáticas para la especificación de color. Un espacio de color es una representación matemática de nuestras percepciones visuales, y nos permite analizar y gestionar el color. La rama de la ciencia del color permite que se trate numéricamente el color de un estímulo visual, definida físicamente como la colorimetría.

Definimos los espacios del color en función de la *Commission International de l'Eclairage* (CIE) o Comisión internacional de *l'Eclairage*, el cual representa el sistema de colorimetría. El sistema CIE se basa en los principios de la teoría tricrómica, una generalización empírica que resume las leyes experimentales de coincidencia de colores. La teoría de tri-cromática establece que cualquier estímulo de color puede ser igual por cantidades adecuadas de tres estímulos primarios (por ejemplo, colores aditivos de color rojo, verde y azul). Esta teoría se basa en la hipótesis de que la retina humana tiene tres tipos de sensores de color (conos), y que la diferencia en sus respuestas espectrales contribuye a la sensación de color.

Las cantidades de los tres colores primarios conocidos como triestímulo del espectro, se necesita para que coincida con una cantidad de unidad de potencia en cada longitud de onda, donde el espectro visible está dado por las funciones de igualación de color. Estas funciones de igualación de color están, relacionadas con las sensibilidades espectrales de los tres conos por transformaciones lineales. Un

color de clasificación sistema de acuerdo con el sistema visual humano, se define por la CIE. Este sistema de distribución de energía espectral es emitido por un objeto en término de tres funciones de igualación de color. Estas funciones son las sensibilidades de un observador estándar a la luz de diferentes longitudes de onda. La ponderación se realiza sobre el espectro visual, de alrededor de 360 nm a 830 nm a intervalos de 1nm(Fairchild).

El proceso anterior produce tres valores triestímulo CIE, XYZ, que describen un color como en la figura 2.16 antes de ser separada como se muestra en la otra figura 2.17.



Figura 2. 16 Imagen original utilizada para las comparaciones de espacio de color.

Las coordenadas de cromaticidad y espacios de color se pueden derivar de estos valores triestímulo. Si nos imaginamos que cada uno de los tres atributos utilizados para describir un color son ejes en un espacio 3D, esto se define como un espacio de color. Los colores que podemos percibir pueden ser representados por el sistema CIE. Otros espacios de color son subconjuntos de este espacio perceptual.



Figura 2. 17 Los tres componentes del espacio de color XYZ extraído de la imagen 2.16.

2.3.2.2 RGB

El modelo de color RGB (Red Green Blue) es un espacio no lineal de color que emplea un sistema de coordenadas cartesianas, con los tres ejes que corresponden a los colores primarios como roja, verde y azul como se muestra en la figura 2.18. Todos los colores se encuentran en la unidad cúbica del subconjunto del sistema de coordenadas 3D. La diagonal principal del cubo, con cantidades iguales de cada primaria, representa gris: negro (0, 0, 0) y el blanco (1, 1, 1). La transformación para convertir desde el espacio RGB a la norma CIE XYZ independiente del dispositivo está dada por:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} X_r & X_g & X_b \\ Y_r & Y_g & Y_b \\ Z_r & Z_g & Z_b \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (2.1)$$



Figura 2. 18 Los tres componentes del espacio de color RGB extraído de la imagen 2. 16.

2.3.2.3 HSV

El acrónimo HSV (tonalidad, saturación, valor) representa una variedad de espacios de color de una transformación no lineal del RGB. El modelo HSV es un sistema de coordenadas cilíndrica, y el subconjunto del espacio dentro del cual se define el modelo es una pirámide de seis lados, o HexaCone como se muestra en la figura 2.19. La parte superior de la pirámide corresponde a $V = 1$, que contiene los colores brillantes. La tonalidad H , se mide por el ángulo alrededor del eje vertical, con rojo en 0° , verde en 120° , y así sucesivamente. Donde los colores complementarios son 180° uno del otro. El valor de la saturación S , es una relación que varía de 0 en la línea central (eje V) a 1 en los lados triangulares de la pirámide. La pirámide es una unidad en V , con el vértice en el origen. El punto en el vértice es negro y tiene una coordenada v de 0. En este punto los valores de H y S son irrelevantes. El punto $S = 0, V = 1$ es de color blanco. Los valores intermedios de V de $S = 0$ (en la línea central) son los grises. Cuando $S = 0$, el valor de H es irrelevante. Por ejemplo, el rojo puro es en $H = 0, S = 1, V = 1$. De hecho, cualquier color con $V = 1, S = 1$ es similar al pigmento puro de un artista utilizado como el punto de partida en la mezcla de colores. La adición de pigmento blanco corresponde a la disminución de S (sin cambiar V). Las sombras se crean manteniendo $S = 1$ y con la disminución de V . Los tonos se crean mediante la disminución tanto S y V . El cambio de H corresponde a la selección del pigmento puro con el que se empieza, un ejemplo claro se puede observar en la figura 2.20, donde los valores de tono y saturación se mapean en el rango $[0, 255]$ para la visualización y la matiz está representado por un ángulo, los niveles de imagen en blanco y negro corresponden al mismo valor de matiz.

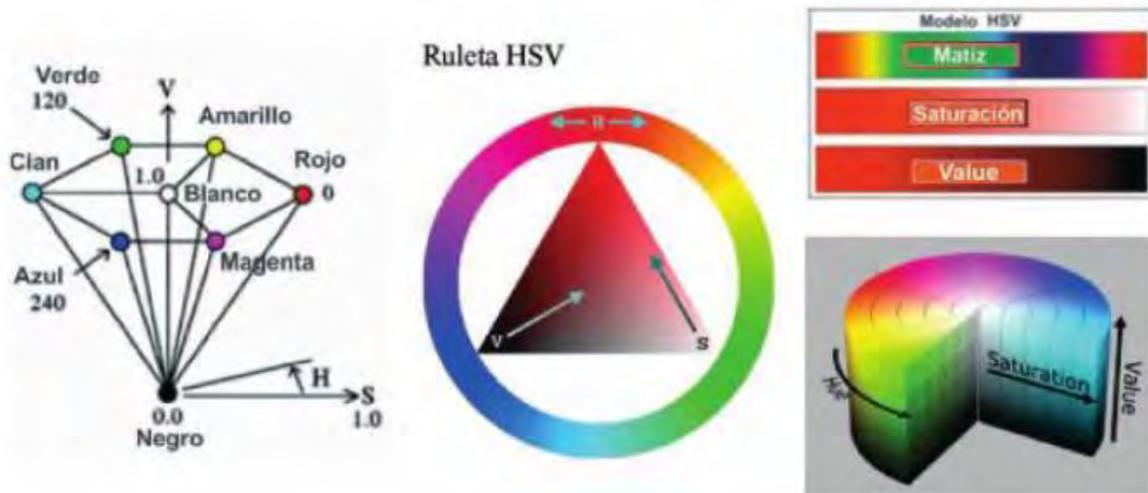


Figura 2. 19 Modelo HSV en forma de pirámide regular hexagonal Invertida.



Figura 2. 20 Los tres componentes del espacio de color HSV extraídos de la figura 2. 16.

El procedimiento que realiza la conversión entre el espacio RGB y el espacio HSV está dada por las siguientes ecuaciones (Hughes, y otros, 2014):

Donde h define como:

$$H = 60 \begin{cases} \frac{G-B}{\delta} & \text{if } R = \max(R, G, B) \neq 0 \\ 2 + \frac{B-R}{\delta} & \text{if } G = \max(R, G, B) \\ 4 + \frac{R-G}{\delta} & \text{if } B = \max(R, G, B) \end{cases} \quad (2.2)$$

Donde V va relacionado al valor máximo de cualquier valor R , G o B :

$$V = \text{imax}(R, G, B) \quad (2.3)$$

$$S = \begin{cases} \frac{\max(R, G, B) - \min(R, G, B)}{\max(R, G, B)} & \text{if } \max(R, G, B) \neq 0 \\ 0 & \text{if } \max(R, G, B) = 0 \end{cases} \quad (2.4)$$

2.3.3 CARACTERÍSTICAS NIVEL MEDIO

El mapeo de una imagen en cuanto a características de bajo nivel, puede no ser el más adecuado para lograr resumir el contenido de una imagen, reduciendo así la eficacia de un seguidor de video. Una solución muy utilizada es el de analizar el video aplicando subconjuntos de píxeles que representan las estructuras pertinentes por ejemplo, los bordes, cuyos píxeles comparten algunas propiedades comunes como el color. En las siguientes secciones vamos a discutir la forma de cómo generar estos métodos para llegar a la aplicación de bordes aplicando la dilatación y erosión para optimizar dichos resultados.

2.3.3.1 DILATACIÓN

La dilatación consiste en adicionar los píxeles del contorno de objetos deseados en una imagen binaria, por ejemplo, supongamos que tenemos un cuadro blanco con puntos negros, al dilatar la imagen esta irá reduciendo los puntos negros y en algunos casos los más pequeños los pintará por completo de color blanco, al igual que su contorno ira expandiendo el cuadro haciéndolo ver más grande como se muestra en la figura 2.21 (Gonzales & Woods).

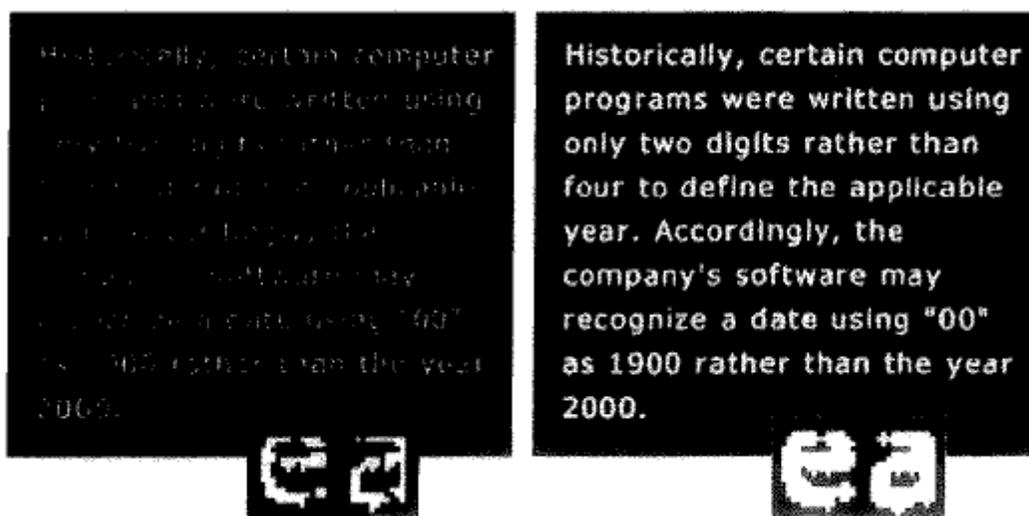


Figura 2. 21 Ejemplo del modelo de dilatación

Para poder aplicar este método se puede resumir en cinco pasos:

1. Seleccionar el pixel de la imagen a tratar (RGB.255, 255, 255).
2. Seleccionar si tiene una coincidencia con algún pixel a dilatar (RGB. 0, 0, 0).
3. Separar el pixel tratado en uno de sus otros canales RGB por ejemplo 0, 0, 255.
4. Realizar este paso hasta cubrir todos los pixeles de la imagen
5. Buscar todos los pixeles dilatados (0, 0, 255) y pasarlos al pixel tratado (255, 255, 255).

2.3.3.2 EROSIÓN

La erosión consiste en eliminar píxeles del contorno de objetos deseados presente en la imagen. Por ejemplo, supongamos que tenemos el mismo cuadro blanco con puntos negros; al erosionar la imagen esta resultará contraria a la dilatación, aumentando los puntos negros dentro de la imagen y al igual que su contorno irá disminuyendo el cuadro haciéndolo ver más pequeño o desgastado como se muestra en la figura 2.22(Gonzales & Woods).

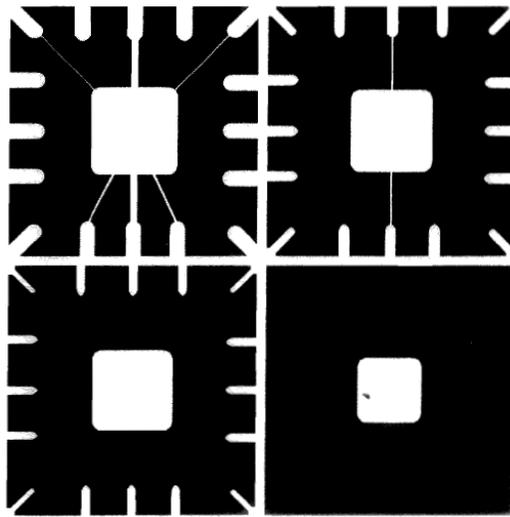


Figura 2. 22 Ejemplo del modelo de dilatación.

Este método se resume en cinco pasos:

1. Seleccionar el pixel de la imagen a tratar (RGB.255, 255, 255).
2. Seleccionar si se tiene una coincidencia con algún pixel a erosionar (RGB. 0, 0, 0), en caso de que el pixel tratado (255, 255, 255) se encuentre entre puros pixeles separados para la erosión (0, 0, 255) este no se verá afectado por la erosión.
3. Separar el pixel tratado en uno de sus otros canales RGB por ejemplo 0, 0, 255.

4. Repitiendo este procedimiento del paso 3 hasta cubrir todos los pixeles de la imagen
5. Busca todos los pixeles dilatados (0, 0, 255) y pásalos al pixel tratado (255, 255, 255).

2.3.3.2 BORDES

Los detectores de borde generalmente producen un mapa binario que pone de relieve la presencia de bordes o elementos de borde en una imagen. Este mapa binario se denomina mapa de bordes, y normalmente están asociados a menudo con las respuestas de los operadores de derivación de primer orden. Una forma muy simple de detección de bordes consiste en un umbral a la magnitud imagen o gradiente. El borde del mapa se puede representar de la siguiente manera:

$$E(\mu, v) = \begin{cases} 1 & |\nabla I(\mu, v)| > T \\ 0 & \text{si no} \end{cases}$$

Donde T es un umbral que depende de la técnica de cálculo del gradiente así como la intensidad de los bordes que se quieren resaltar, como se muestra en la figura 2.23(Gonzales & Woods).

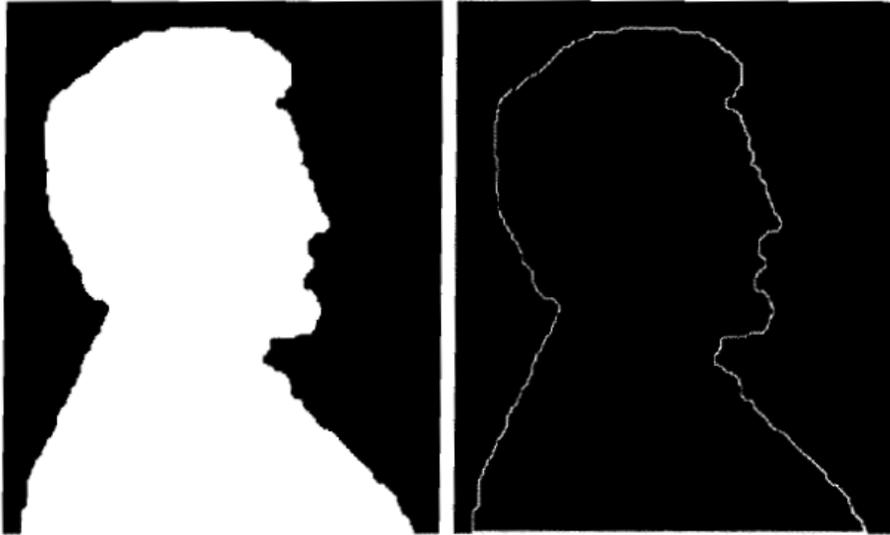


Figura 2. 23 Extracción de Bordes.

CAPÍTULO 3

CAPÍTULO 3

Para realizar este proyecto se utilizó un equipo HP con aditamentos de expansión en la memoria RAM, con las siguientes características:

- ❖ El Equipo HP Compaq Business Desktop dc7700 - Core 2 Duo E6600 2.4 GHz contiene las siguientes características:
 - Tiene una dimensión de 16.8 cm x 45.2 cm x 44.8 cm con un peso aproximado de 14.1kg;
 - Procesador: Core 2 Duo E6600 2.4 GHz;
 - Memoria RAM: originalmente de 1GB expandida a 4GB DDR II SDRAM - no ECC -800 MHz -PC2 – 6400;
 - Controlador Serial SATA con un disco duro de 80 GB SATA3.0 Gb/s de 7200 rpm.
 - Tipo de chasis Mini torre.
- ❖ Características del sistema:
 - Subsistema gráfico: Intel® Graphics Media Accelerator 3000.
 - Características del adaptador de video: 3D/2D controlador (Microsoft® DirectX® 9), controlador integrado VGA, RANDAC integrado (400 Mhz).
 - Tarjeta de video: Memoria gráfica de 8 a 128 MB.
 - 2 full-height PCI, 1 full-height PCI Express x1, 1 full-height PCI Express x16, 2 opcional full-height PCI, HP 16-in-1 Media Reader.
 - Sonido: Integración de gran definición de audio Realteck ALC262 con códecs de Alta definición de Audio.
 - Modem: Hi-Speed 56K PCI SoftModem
 - Interface de red: Intel 82566DM Gigabit
 - Tarjeta de red: Intel Pro 1000 PCIe Gigabit NIC
 - Puertos Externos I/O: Traseros: 6 USB 2.0, 1 puerto serial estándar, 1 puerto serial opcional, 1 puerto paralelo, 2 PS/2, 1 RJ-45, 1 VGA, audio entrada/salida, Delanteros: 2 USB 2.0, auricular and micrófono.

- Teclado: 2004 teclado estándar PS/2.
- Mouse /dispositivo de indicación: HP PS/2 2- Botón Ratón óptico.
- Gestión de Seguridad: Embedded TPM1.2 con HP ProtectTools opcional, Embedded Security software.
- Consumo de Poder: 365 Watts máximo.
- Requerimientos de poder: Entrada de Voltaje 90 – 264 / 100 – 240 VAC, 50/60 Hz, 47 – 63 Hz, activa PFC.
- Rango de temperatura: 10° a 25°
- Rango de Humedad: 10% a 90% RH.
- Rango de Humedad Inoperativa: 5% a 95% RH.



Figura 3. 1 Equipo Desktop dc7700.

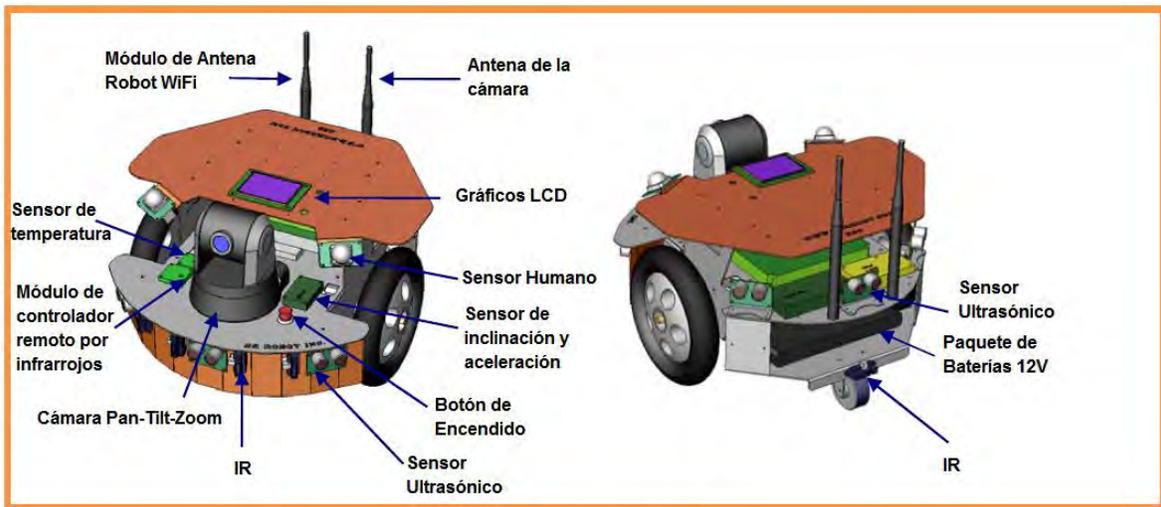
Características del Robot *Dr. robot X80SVP* con cámara Pan-Tilt-Zoom de alta resolución:

- ❖ Características generales:
 - 2.6X Camara Pan-Tilt-Zoom. CCD de alta resolución con capacidad de audio de doble sentido.
 - 2 Motores de 12V con 500 oz.-pulgada (40 kg.cm) de fuerza de torsión.
 - Red Inalámbrica 802.11g.
 - herramientas de desarrollo de aplicaciones independientes.
 - Máxima velocidad de 75 cm/s.
 - LCD grafico de 128x64; el cual muestra imágenes, mensajes o datos del sensor.
 - Sensores de detección de colisión incluyendo 6 ultrasónicos y 7 sensores infrarrojos.
 - 2 sensores piroeléctricos de movimiento humano.
 - Circuito de protección.
 - Máxima carga útil 15 kg , con el peso del robot de 3.5 kg.
 - Dimensiones: 38 cm x 35 cm x 28 cm.

El dispositivo se puede apreciar en la figura 3,2, así como una imagen de la ubicación específica de los dispositivos que con este cuenta en la figura 3.3.



Figura 3. 2 Dr. robot X80SVP.



Visión General X80SVP

Figura 3. 3 Partes de Dr. robot X80SVP.

3.1 INSTALACION Y CONFIGURACIÓN

Dentro de los propósitos necesarios se tuvo que llevar a cabo ciertas tareas para la configuración del equipo así como la instalación de paquetería o librerías necesarias para el análisis biométrico:

- Configuración del Robot Dr. Robot.
- Instalación de OPENCV (revítese el ANEXO B).

3.1.1 CONFIGURACION DEL ROBOT

En la primera fase, Se debe llevar a cabo la instalación del software , que servirá para propósitos de conectividad con el robot.

La siguiente imagen puede explicar su funcionalidad figura 3.4.

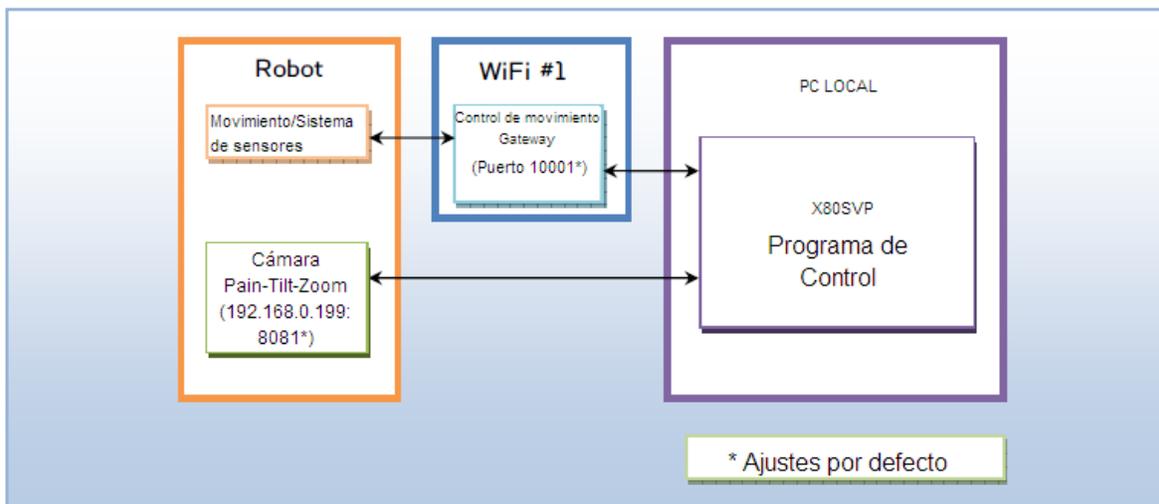


Figura 3. 4 Esquema General de Conectividad.

El robot para poder interactuar completamente con el Sistema de sensores y movimiento, cuenta con una aplicación intermedia llamada “*Gateway de control de movimiento*”, la cual direcciona los datos en un puerto especificado así como en una dirección asignada, sirviendo de intermediario para que el programa pueda comunicarse con el robot mientras el equipo procesa los datos de la cámara (*Pan-Tilt-Zoom*).

Una vez instalado los elementos del Dr. Robot, se utilizará la aplicación mostrada en la Figura 3.5 la cual solo se modifica el nombre (Robot ID) y la dirección mostrada por la que se proporciona en la parte inferior del robot en el caso con terminación .203.

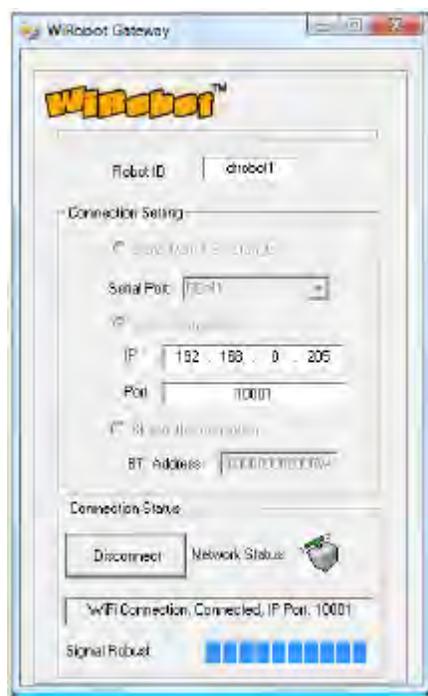


Figura 3. 5 Aplicación Intermedia para Conectividad.

3.2 IMPLEMENTACION DEL CODIGO

Para poder extraer los datos de nuestra cámara IP tuvimos que utilizar unas librerías que tienen elementos o codificadores internos para ese dispositivo en común, conocido como “VITAMINDECODERLib”, la cual contiene la información de la cámara IP permitiendo el acceso y la exportación de información. Dicha librería se encuentra contactando al proveedor en su sitio <http://www.drrobot.com/>, la cual dispone de la información necesaria para la manipulación de los dispositivos contenidos en el Robot.

VITAMINDECODERLib

Una vez teniendo en cuenta dicha librería se aplicó el siguiente código para poder capturar el video en un vector de bits.

```
axVitaminCtrl1.GetSnapshot(VITAMINDECODERLib.EPictureFormat.ePicFmtBmp, outvData, outvInfo);
```

El cual nos permite extraer la información de la cámara en diversos formatos (bmp, YUV, jpeg, RAW24, YUY2, YV12) de los cuales por motivo de configuración y compatibilidad se decidió aplicar el bmp (ePicFmtBmp). Otra es la extracción del video por medio del objeto vData el cual nos permite por medio de una asignación a un vector tener la información numérica distribuida en 3 regiones en un modelado RGB, donde cada 3 valores en el vector van formando un solo pixel y así sucesivamente.

Una vez determinado el formato a extraer este será asignado en una variable tipo “byte” para luego ser convertida en un “*Bitmap*”, debido a que c# no puede realizar la conversión directa de un tipo objetos a un imagen, se tuvo que pasar a un tipo byte, posterior mente a un tipo Bitmap y luego a un tipo image.

```
//Extrae los datos de vData en tipo Byte
bData = (byte[])vData;
//hace referencia a un método
TypeConverter tc = TypeDescriptor.GetConverter(typeof(Bitmap));
//Convierte los valores de bData y los asigna en una nueva variable
Bitmap img = (Bitmap)tc.ConvertFrom(bData);
```

Esto nos permitirá poder asignar nuestra nueva variable “*img*” a una de tipo “*image*” la cual nos permitirá manejar la información mucho más rápido y efectivo con los métodos de OPENCV

```
Image<Hsv, Byte> impr = newImage<Hsv, byte>(img);
```

```
//original es un tipo Image<Hsv, Byte> original;
original = impr;
```

Nótese que “*image*” está declarado en HSV debido a que es mucho más fácil manipular los colores en grados por su forma HexaCone, esto se puede verificar en los aportes del marco teórico de este documento en la sección HSV.

Una vez que se haya obtenido original podemos trabajar con él para realizar un filtrado como se muestra a continuación:

```
//realizando el filtrado de HSV
//procesado es un tipo Image<Gray, Byte> original;
procesado = original.InRange(
//valores mínimos para el filtrado
newHsv((trackBar1.Value - trackBar4.Value), trackBar6.Value, trackBar5.Value),
//valores máximos para el filtrado
newHsv((trackBar1.Value + trackBar4.Value), trackBar2.Value, trackBar3.Value));
```

El siguiente figura 3.6 puede mostrar mejor el funcionamiento del código anterior.

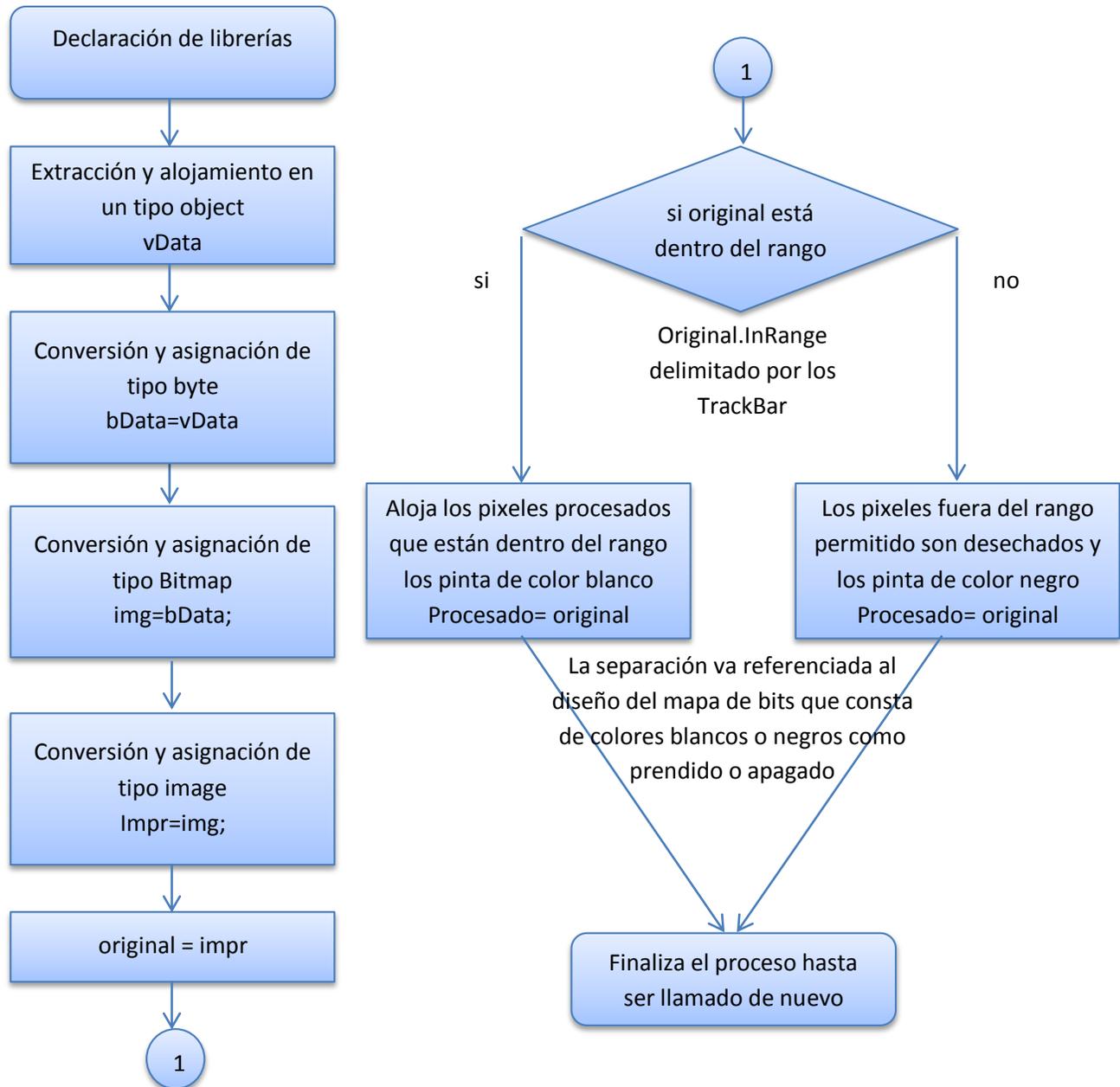


Figura 3. 6 Diagrama de procesos

Este método permite poner parámetros de máximos y mínimos para así almacenarlos en una variable de cualquier otro tipo. En nuestro caso se almacena en la variable “*procesado*”, la cual es de tipo “*Gray*”. La imagen que se almacenó en esta variable, no es otra que una imagen blanco o negro o mejor conocido como una imagen binaria. Las figuras 3.7 – 3.10 muestran los resultados de aplicar este método con diversas tonalidades en diferentes horarios:



Figura 3. 7 Filtrado con Color Azul mezclilla.

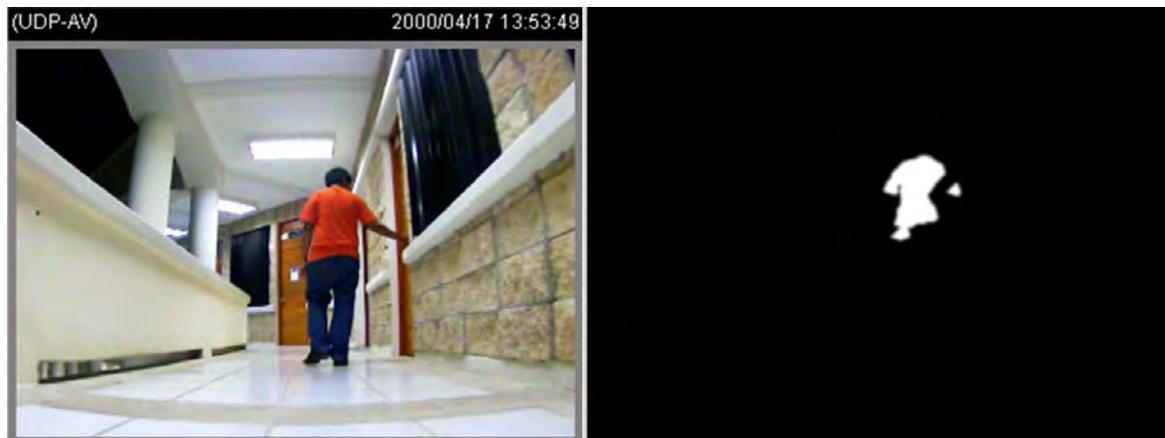


Figura 3. 8 Filtrado con Color Naranja.



Figura 3. 9 Filtrado con Color Azul Marino.



Figura 3. 10 Filtrado con Color Rojo.

En la tabla 3.1 podemos ver los parámetros aplicados para el segmentado de color y los horarios de aplicación, estos datos van relacionados con los resultados que se muestran en las figuras 3.7 - 3.10.

Tabla 3.1

Tabla de Valores de Comparación de Filtrado colores en modelo HSV							
Figura	Hue Max	Hue Min	Sat Max	Sat Min	Val Max	Val Min	Horario
3.7	140	80	260	100	140	150	8:30PM a 8:45PM
3.8	15	0	260	95	260	138	7:30 PM a 7:45pm
3.9	135	85	260	110	260	0	1:00 PM a 1:30PM
3:10	200	160	260	80	200	50	3:30PM a 4:00PM

En la Figura 3.7 se puede apreciar un leve ruido causado por la luz artificial debido al Ángulo en que se encuentra y la tonalidad produce leves colores azulosos en los bordes negros de las ventanas. En la Figura 3.10 se puede apreciar la pérdida de color sobre el objetivo y el aumento del ruido por la combinación del rojo en el marrón de las puertas y parte del rostro.

El siguiente paso es la aplicación de un método de cerrado el cual consiste en dilatar la imagen para llenar los espacios vacíos (esta sección puede verificarse en la parte de Dilatación y Erosión del marco teórico), y posteriormente aplicar una erosión para eliminar puntos salientes en los bordes así como disminuir el ruido amplificado o incluso eliminarlo:

```
//método de cerrado
//Obtiene una imagen en bitmap
Image<Gray, Byte> src = newImage<Gray, Byte>(procesado.Bitmap);
//Obtiene sus dimensiones
Image<Gray, Byte> dst = newImage<Gray, Byte>(src.Width, src.Height);
//proceso de cerrado empezando por 5 dilataciones y luego 5 erosiones
StructuringElementEx element = newStructuringElementEx(5, 5, 1, 1,
Emgu.CV.CvEnum.CV_ELEMENT_SHAPE.CV_SHAPE_CROSS);
//proceso de cerrado que se repite 3 veces
CvInvoke.cvMorphologyEx(src, dst, IntPtr.Zero, element, CV_MORPH_OP.CV_MOP_CLOSE, 3);
//Almacena el nuevo valor en procesado de nuevo
procesado = dst;
```

Para darle mayor contorno al objetivo le aplicaremos una gaussiana, el cual suaviza la imagen por medio de una gaussiana.

```
//Se aplica un suavizado
procesado = procesado.SmoothGaussian(9); //aplicacion de una gaussiana
```

Ahora analizaremos un área radial, por medio del método CifcleF, este nos permite calcular el radio de una zona percibida, donde por medio de este método se puede percibir un radio constante para detectar las mayores presencias en la imagen binaria, descartando parte del ruido.

```
CircleF[] circulos = procesado.HoughCircles(
newGray(100), //Umbral de canny
newGray(50), //acumulación del Umbral
2, //acumulación de resolución
procesado.Height / 4, //Mínimo de distancia entre pixeles para detectar el circulo
10, //mínimo de radio de círculos
70)[0]; //máximo de radio de círculos
```

Para extraer la posición así como el radio de los elementos, se aplica un foreach dentro de otro método CircleF, permitiendo tener una posición dentro de los límites de la imagen en base a las coordenadas XY.

//Iniciamos un foreach para dentro del círculos podamos extraer la posición y el radio de circulo

```
foreach (CircleF circulo in circulos)
{
//En caso que las coordenadas del punto cambie reinicia el máximo
if ((x != int.Parse(circulo.Center.X.ToString().PadLeft(4))) && (y !=
int.Parse(circulo.Center.Y.ToString().PadLeft(4))))
{
radiomax = 0;
}
//Si el radio es mayor o igual lo almaceno como máximo
if (radiomax >= double.Parse(circulo.Radius.ToString("###.000").PadLeft(7)))
{
radiomax = double.Parse(circulo.Radius.ToString("###.000").PadLeft(7));
}

//Asignación de variables de x y del radio
x = int.Parse(circulo.Center.X.ToString().PadLeft(4));
y = int.Parse(circulo.Center.Y.ToString().PadLeft(4));
//extracción del radio
radio = double.Parse(circulo.Radius.ToString("###.000").PadLeft(7));
```

Finalmente, resta asignar las condiciones para poder posicionar en el centro y en un radio adecuado, donde la condición 1 y 2 que se definen mas adelante del documento centran o desvían al robot a la izquierda o derecha dependiendo de si el objetivo está dentro del rango menor a 200 o mayor a 440 del tamaño total de la pantalla (640).

Así como las condiciones 3 y 4 realizan su cálculo por el radio a dependiendo de que si el rango es muy pequeño obligando al robot a acercarse para aumentarlo y así inversamente y aplicando limitantes en las que lo centran.

En la condición 5 neutraliza todo movimiento ubicándolo en una zona central permitiendo así poder meter en esa condición los métodos de pausa de motores.

Condición 1

```
//giro a la izquierda
if ((x < 200) && (radiomax <= radio))
```

Condición 2

```
//giro a la derecha
if ((x > 440) && (radiomax <= radio))
```

Condición 3

```
//se acerca al objetivo si la condición detecta que el radio mínimo es menor que el rango permitido
//descartando así posibles ruidos
if ((radio < 25) && (radio > 15) && (radiomax <= radio) && ((y + radio) > 50) || ((y + radio) < 330))
```

Condición 4

```
//retrocede del objetivo si la condición del radio máximo lo amerita para descartar ruidos así como la
posición
if ((radio > 60) && (radio < 70) || ((y + radio) > 330) && (radiomax >= radio))
```

Condición 5

```
//centrado el objetivo para apagar el movimiento
/////150/////
/////-----/////
//250/|x|/390//
/////-----/////
/////330/////
```

```
if ((x > 250) &&
(x < 390) &&
(radio > 25)&&
(radio <60))
```

Para el método de pausa y de giros de motores se aplicó el motor con un valor intermedio de 0 a de16363 y otro de 16363 a 32767, en donde el 50% o 16363 representan ningún movimiento en los motores sin apagarlos, mientras más alejado este el valor asignado de 16363 este aumentara el voltaje de los motores y por lo tanto el movimiento hasta llegar al 100% en las cantidades de 0 y 32767.

Izquierda:

```
//velocidad calculada del motor para giro a la izquierda
short speed = (short)(7183 - (x * giro));
//encendiendo motores, recordemos que los motores se ven en lados opuestos
//por lo tanto cuando tengan el mismo valor giraran de lado contrario
myRobot.DcMotorPwmTimeCtrAll(speed, speed, 16363, 16363, 16363, 16363, 1);
```

Derecha:

```
//velocidad calculada del motor para giro a la izquierda
short speed = (short)(25545+((x - 440) * giro));
//encendiendo motores, recordemos que los motores se ven en lados opuestos
//por lo tanto cuando tengan el mismo valor giraran de lado contrario
myRobot.DcMotorPwmTimeCtrAll(speed, speed, 16363, 16363, 16363, 16363, 1);
```

Adelante:

```
//velocidad calculada del motor para avanzar
short speed = (short)(25545+(y * vel + (radio*10)));
short speed2 = (short)(7183-(y * vel + (radio * 10)));
//encendiendo motores, de diferentes valores giran al mismo lado
myRobot.DcMotorPwmTimeCtrAll(speed, speed2, 16363, 16363, 16363, 16363, 1);
```

Atrás:

```
//velocidad calculada del motor para retroceder
short speed = (short)(25545 + (y * vel + (radio * 10)));
short speed2 = (short)(7183 - (y * vel + (radio * 10)));
//encendiendo motores, de diferentes valores giran al mismo lado
myRobot.DcMotorPwmTimeCtrAll(speed2, speed, 16363, 16363, 16363, 16363, 1);
```

Pausado:

```
//detención del movimiento de los motores sin apagado
myRobot.DcMotorPwmTimeCtrAll(16363, 16363, 16363, 16363, 16363, 16363, 1);
```

CAPÍTULO 4

CAPÍTULO 4

RESULTADOS EXPERIMENTALES

Para la fase de pruebas, se aplicaron 4 tipos diferentes de filtros de color, con la finalidad de disminuir el ruido del entorno producido por el seguimiento en diversos horarios, y así compensar el problema de luminosidad.

Una vez ya bien configurado el código mostrado en el capítulo 3 pudimos realizar las primeras pruebas de orientación, mostrando circunferencias al momento de identificar un círculo dentro del filtrado de color este lo registra como un punto a seguir, tal como se muestra en la figura 4.1.

Dentro del historial mostrado en la figura 4.1 del lado derecho se puede aplicar constantes cambios que en algunos casos eran efecto de las sombras producidas por la contra luz o la luz expuesta del ambiente.



Figura 4. 1 Historial de seguimiento en la noche

Al seguir probando con otras tonalidades de colores, podríamos notar ciertos patrones, los cuales provocaban confusión por su cambio tan repentino en la tonalidad. En algunos de estos casos, el radio a pesar de encontrarse enfrente del objeto aparecía muy pequeño por lo que mostraba que perdía de vista al objeto por un cambio mínimo de luz, como se muestra en la Figura 4.2.

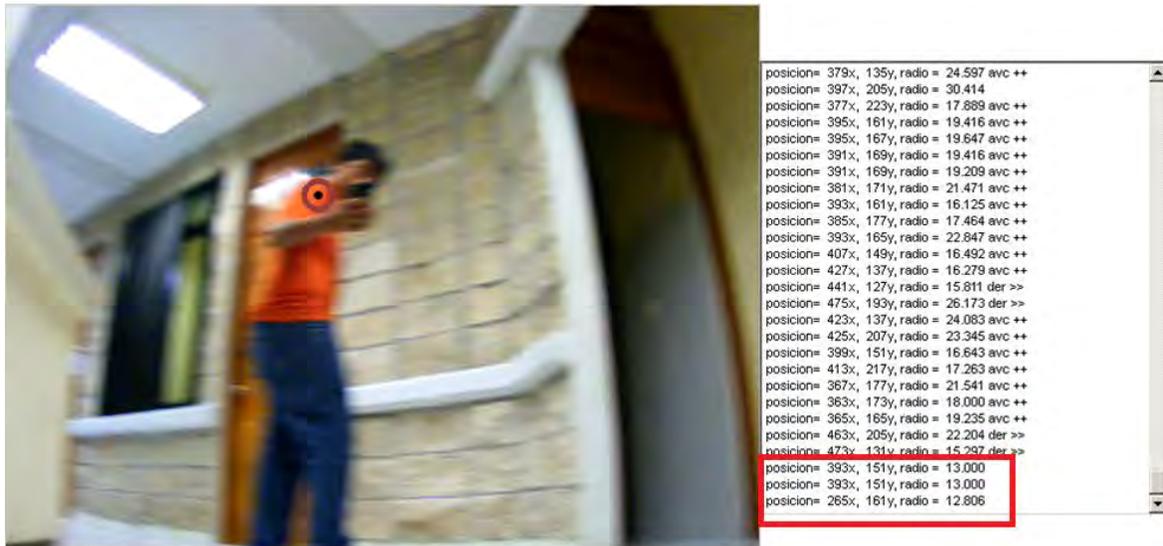


Figura 4. 2 Complejidad de radio.

Para corregir este detalle, se tuvo que aplicar otro objetivo de color, por lo que se llegó a la conclusión que era mejor aplicar colores azules en el día y rojizos en la noche, puesto que como el entorno en el que se trabajó no era totalmente controlado, surgían los problemas de pérdida de datos o ruido, en donde algunos casos era tan fuerte, que perdía de vista al objetivo, para seguir al falso positivo.

En la Figura 4.3 se aplicó un filtro, para una persona con pantalón azul marino, la cual está en el horario de la 1 pm a 2 pm. La imagen obtenida parecía mostrarse perfectamente perfilada, facilitando la tarea de rastreo o seguimiento, incluso, se puede apreciar el lado derecho de la persona. En la Figura 4.3 se muestra el listado de las instrucciones que recibía en ese momento el robot, de girar al detectar la presencia del objetivo cerca de su límite derecho.

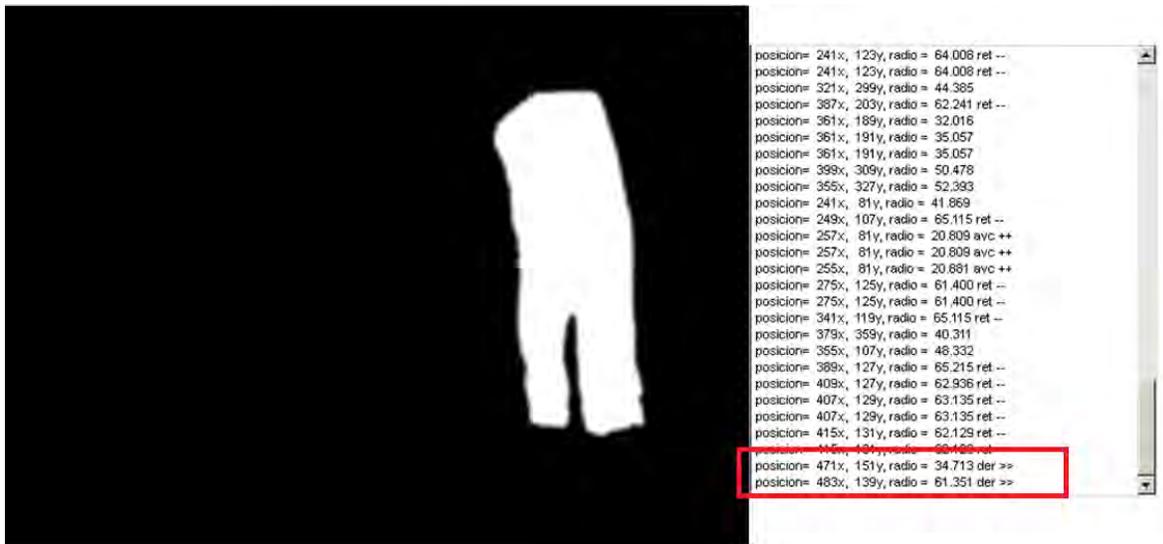


Figura 4. 3 Filtrado azul 1pm.

Conforme pasaba el día, se tenía que cambiar los límites del filtrado de color para poder percibir mejor el objetivo. Para ello, se realizaron pruebas hasta después de las 10 de la noche. En este horario no había luz externa de los pasillos que interfiriera, además de que permitió tener control casi total de la luz en un cubículo. De esta forma se logró tener un poco de precisión, aplicando un filtrado rojo como se muestra en la Figura 4.4. Se aprecia claramente como el color rojo como objetivo es resalta en el filtrado, sin que falsos objetivos se introduzcan dentro del filtro,

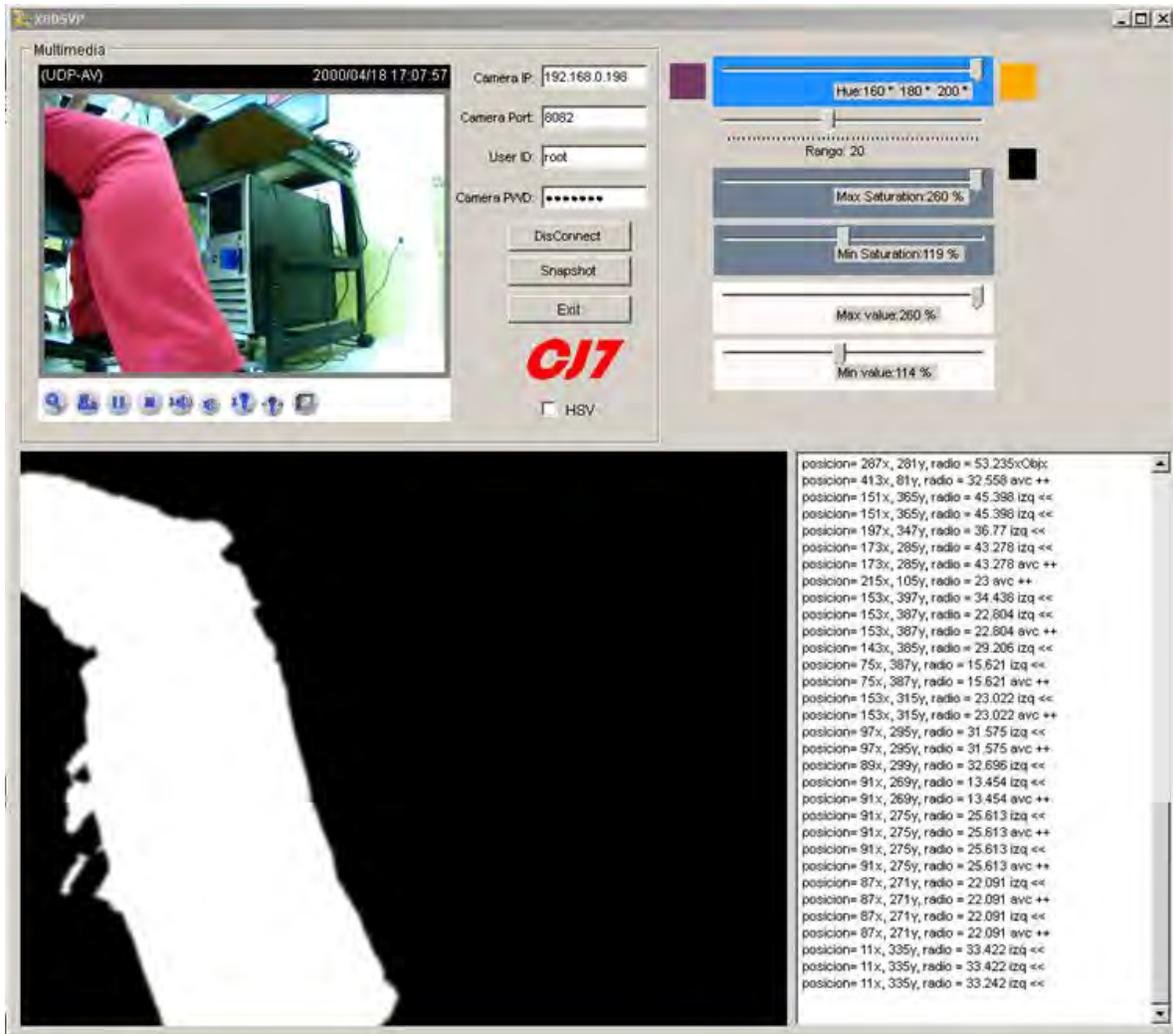


Figura 4. 4 Filtrado en Color rojo.

CAPÍTULO 5

CAPÍTULO 5

CONCLUSIONES

El seguimiento de objetos es un tema de estudio amplio. Hoy en día, el seguimiento de objetos se aplica cada vez más y más en los medios comerciales y de seguridad. Con ello, cada día se mejoran más los dispositivos que nos permiten capturar los datos producidos por la tarea de seguimiento, para luego convertirlas en instrucciones. Con los avances de la tecnología, cada día se puede esperar más métodos ágiles que procesen más información y de mejor calidad, sin mencionar el área de la electrónica, donde podemos tener dispositivos más pequeños y funcionales, con mejor procesamiento así como mayor velocidad de transmisión de datos y rendimiento en cuanto al tiempo de encendido permitiendo de esta área ser un campo mucho más amplio.

En la actualidad hay métodos de seguimiento aplicados para dispositivos como cámaras web en entornos empresariales o de aficionados. Para la gente que aún está entrando en esta área de la visión computacional, es muy útil, gracias al aprendizaje que se va adquiriendo poco a poco, con métodos no tan robustos, óptimos y eficaces que como principio del seguimiento, dando la apertura a que no solo se puede aplicar para un entorno virtual, sino que ya se puede experimentar en un área más amplia, como la robótica, dando una nueva perspectiva de lo que se puede lograr el día de mañana, dejando siempre la duda de cómo ir mejorando, y aprendiendo más, de cómo ir dándole otros usos a este a la visión computacional.

En el caso particular de este trabajo, se vio algunas particularidades de este dispositivo Drrobot X80SVP, como era la falla de resolución de la cámara por la variación de la corriente de la batería, así como la problemática de la iluminación,

puesto que el proyecto, se planteó para un entorno controlado, y muchas veces la influencia de la luz ambiental afectaba creando sombras o incluso la luz del techo al tener la cámara desde una posición mucho más baja del Drrobot X80SVP creaba contra luz o sombras en algunas partes de la ropa. Aun así, se obtuvieron buenos resultados, puesto que a pesar de los problemas presentados, el robot logro hacer el seguimiento y distinguir entre el entorno que resultaba perjudicial y realizar su seguimiento de manera constante.

De igual manera al trabajar con este Drrobot X80SVP en particular se visualiza que en un futuro pueda aprovecharse más, este mismo dispositivo, gracias a la cantidad de sensores con los que cuenta; se podría no solo ponerse en un patrón de seguimiento de una persona sino, aprovechar la detección de obstáculos como paredes para realizar mapeos o seguimientos programados para asignación de rutinas. Una de las conclusiones que tome para mejorar el aspecto de iluminación, es diseñar y conectar una tableta de Leds en ciertas áreas del mismo robot, para que la cámara pueda tener una mejor visión y aplicar algoritmos más robustos que le permitan no solo el seguimiento, sino también el reconocimiento de diversos tipos de objetos, más que para un seguimiento, un análisis del área de su entorno.

También gracias a lo que aprendido en este trabajo de tesis, he tenido una nueva perspectiva que se puede tener con simples dispositivos y programas que se tengan al alcance; ya que simplemente hay que dedicarles tiempo para darse cuenta de lo que uno puede llegar a ser capaz de desarrollar. Además, me ha inspirado a tener nuevas metas de querer desarrollar más el día de mañana, de profundizar más, en métodos que permitan ampliar, no solo en el área de la robótica, sino en áreas de uso cotidiano donde se pueda tener un método que fusione el reconocimiento facial, con el seguimiento, así como el reconocimiento de objetos, o incluso para el manejo de vehículos no tripulados.

REFERENCIAS BIBLIOGRÁFICAS

- Ajum, N., & Cavallaro, A. (2008). *Multi-feature object trajectory clustering for video analysis*. IEEE Transactions on Circuits and Systems for Video Technology.
- Bardinet, E., Cohen, L. D., & Ayache, N. (1996). Tracking and motion analysis of left ventricle with deformable superquadrics. En *Medical Image Analysis* (págs. 129-149). Oxford University Press.
- Bengtsson, E. (noviembre de 2008). *CB*. Recuperado el 2014 de mayo de 2, de Digital Cameras, CMOS and CCD sensors: http://www.cb.uu.se/~ewert/DIS_2008_CCD_CMOS.pdf
- Davison, A. J., Reid, I. D., Molton, N. D., & Stasse, O. (s.f.). MonoSLAM: Real-Time Single Camera SLAM., (págs. 1-24).
- EmguCV. (s.f.). Sesión Práctica 1: Instalación y Configuración del entorno;. En *Visión Artificial con OpenCV: Un Enfoque Práctico* (págs. 1-15).
- Fairchild, M. D. (s.f.). En *Color Appearance Models* (págs. 1-34, 53-82, 183-205). John Wiley & Sons.
- Fitzgibbon, A. W., & Zisserman, A. (1998). Automatic Camera Recovery for Closed or Open Image Sequences., (págs. 1-16). London,UK.
- Fonseca, P., & Nesvadba, J. (2006). Face Detection in compressed Domain. En *EURASIP Journal on Applied Signal Processing* (pág. 187). New York: Hindawi.
- Freeman, W. T., Anderson, D. B., Beardsey, P. A., Dodge, C. N., Roth, M., Weissman, C. D., y otros. (1998). Computer Vision for Interactive Computers Graphics. En C. Society (Ed.), *IEEE Computer Graphics and Applications* (Vol. 18, págs. 42-53).
- Gall, D. L. (1991). MPEG: A Video Compression Standard for Multimedia Applications. En *Digital Multimedia Systems* (Vol. 34, págs. 47-58).
- Gemmell, J., Toyama, K., Zitnick, L., Kang, T., & Seitz, S. (2000). Gaze Awareness for Video-conferencing: A software Approach. IEEE.
- General Electric. (s.f.). Recuperado el 22 de abril de 2014, de VicioWave: Intelligent Video Platform: http://utdfs.com.au/documents/VisioWave%20IVP%20Brochure_2006.pdf

- Geng, W., Cosman, P., Member, S., Berry, C. C., Feng, Z., & Schafer, W. R. (2004). Automatic Tracking, Feature Extration and Classification of *C. elegans* Phenotypesqq. En *IEEE Transactions on Biomedical Engineering* (Vol. 51, págs. 1811-1820).
- Gonzales, R. C., & Woods, R. E. (s.f.). Morphological Image Processing. En *Digital Image Processing* (págs. 652-710). Pearson Prentice Hall.
- Huanf, K. S., & Trivedi, M. M. (2003). Video arrays for treat-time tracking of person, head, and face in an intelligent room. En *Machine Vision and Applications* (págs. 103-111). Springer-Verlag.
- Hughes, J. F., Dam, A. V., Mcguire, M., Sklar, D. F., Foley, J. D., Feiner, S. K., y otros. (2014). *Principles and Practice*. Addison-Wesley.
- IBM. (11 de Febrero de 2014). *IBM Smart SurveillanceSystem (S3)*. Recuperado el 2014 de Febrero de 11, de http://www-ee.cuny.cuny.edu/wwwn/yltian/Publications/IBMS3_MVAPaper.PDF
- (s.f.). Software Installation. En D. R. Inc, *WiFi Mobile Tobot Development Platform with High Resolution Pan-Tilt-Zoom CCD camera Quick Start Guide X80SVP*. 1-14.
- INTELLIVID*. (s.f.). Recuperado el 27 de abril de 2014, de IntelliVid's Video INvestigator: <http://www.objectvideo.com/>
- Kenzo, N., Farid, K., Satoshi, S., Wei, W., & Daisuke, N. (2010). *Autonomous Flying Robots: Unmanned Aerial Vehicles and Micro Aerial Vehicles* (primera ed.). Tokyo: Springer.
- Maggio, E., & Cavallaro, A. (2011). *Video Tracking: Theory and Practice*. Chichester, United Kingdom: John Wiley & Sons, Ltd.
- Mountney, P., & Yang, G.-Z. (s.f.). *mountney*. Recuperado el 22 de abril de 2014, de Soft Tissue Tracking for Minimally Invasive Surgery: Learning Local Deformation Online: <http://www.mountney.org.uk/presentations/Miccai%202008.pdf>
- Object Video*. (s.f.). Recuperado el 2014 de abril de 24, de OV: <http://www.objectvideo.com/>
- Phong, T. B. (1975). Illumination for Computer Generated Pictures. En W. Newman, *Graphics and Image Processing* (págs. 311-317).
- Shafer, S. A. (1985). Using color to separate reflection components. En *Color Research & Application* (Vol. 10, págs. 210-218).
- Stenslie, S. (2010). Artistic Projects on Touch Technologies. En *Virtual Touch - A study of the use and experiencie of touch in artistic, multimodal and computer-based environmets* (págs. 148-152). Unipub Forlag AS.

- Tian, Y.-I., Brown, L., Hampapur, A., Lu, M., Senior, A., & Shu, C.-f. (s.f.). *IBM Smart Surveillance System (S3): Event Based Video Surveillance System with an Open and Extensible Framework*. Recuperado el 22 de abril de 2014, de IBM:
http://www-ee.ccny.cuny.edu/wwwn/yltian/Publications/IBMS3_MVAPaper.PDF
- Toshiba*. (s.f.). Recuperado el 22 de abril de 2014, de Artida 2D ultrasound scanner:
<http://www.toshiba-medical.eu/en/Our-Product-Range/Ultrasound/Systems/Artida1/>
- Triggs, B., McLauchlan, P. F., Hartley, R. I., & Fitzgibbon, A. W. (1999). *Bundle Adjustent - A Modern Synthesis*. London, UK.
- vicon. (s.f.). Recuperado el 2 de mayo de 2014, de Vicon:
<http://vicon.com/Press/Details/21>
- Wu, Y., & Huang, T. S. (1999). *Vision-Based Gesture Recognition: A Review.*, (págs. 1-12).

ANEXO A

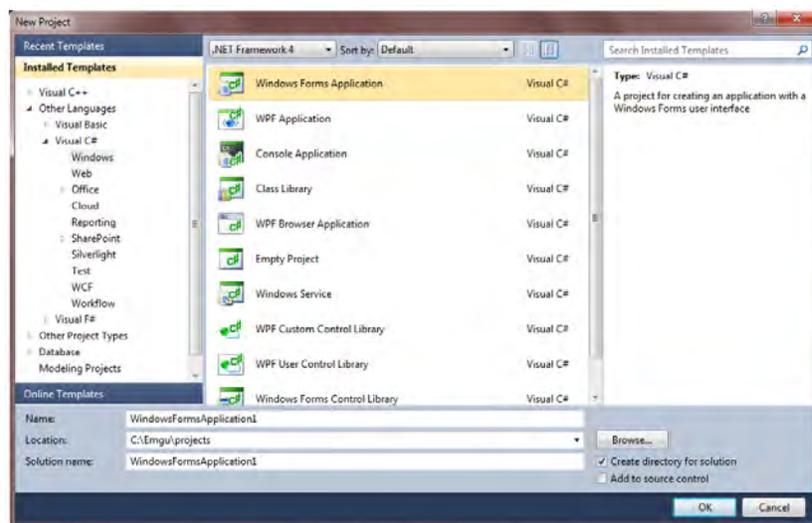
Configuración del OPENCV

Para la configuración de OPENCV se utilizará un programa puente conocido como EMGUCV, el cual está basado en OPENCV pero este trae las librerías directas para el lenguaje a utilizar, en este caso c#, para ello es indispensable descargar las librerías o los .dll de su página oficial:

http://www.emgu.com/wiki/index.php/Main_Page

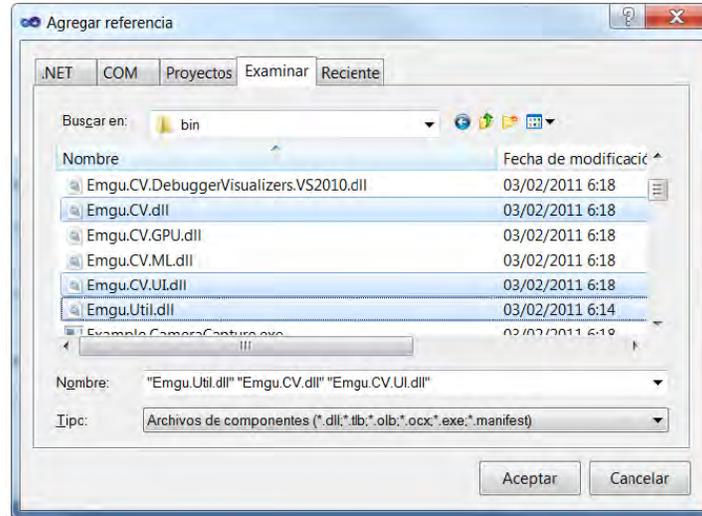
Después de haber descargado los elementos de EMGUCV deben instalarse en el proyecto de c#. Para ello deberán seguirse los siguientes pasos (EmguCV), suponiendo que ya se tiene instalado el Microsoft Visual Studio:

1. Primero se crea el proyecto a requerir con el nombre deseado, como se muestra en la imagen ANEXO A. 1.



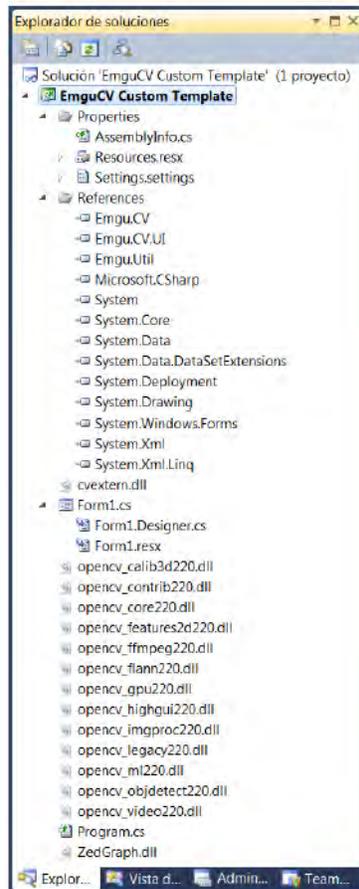
ANEXO A. 2 Nuevo proyecto.

2. Luego en el explorador de soluciones se dará clic derecho y deberá agregarse las referencias: "Emgu.CV.dll", "Emgu.CV.UI.dll" y Emgu.Util.dll, como muestra la imagen ANEXO A. 2.



ANEXO A. 3 Agregando Referencias.

3. Después se procederá a agregar unos elementos existentes, siendo opcional agregar todos o solo los que se vayan a utilizar, para ello debe abrirse el explorador de soluciones y dar clic derecho agregar ahora elemento existente donde se agregan todos los elementos que contengan "opencv_" y terminación ".dll" así como el elemento "cvextern.dll" y "ZedGraph.dll", como se muestra en la imagen ANEXO A. 3.



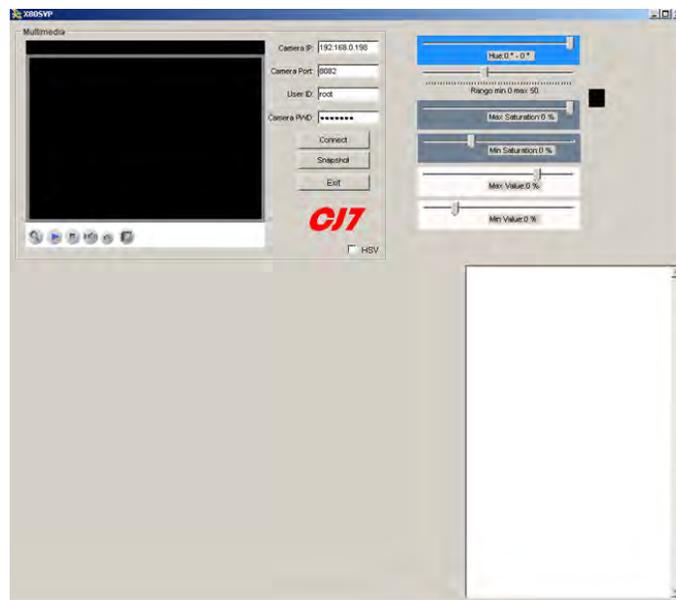
ANEXO A. 4 Elementos agregados.

4. Debido a que tanto EMGU como OPENCV están desarrollados para entornos con equipos que cuentan con tarjeta “NVIDIA”, en caso de estar trabajando con una tarjeta Intel, será necesario realizar algunos cambios, para ello se descarga el siguiente programa llamado “Dependency Walker”.
5. Una vez instalado el programa se abrirá con él la librería “open_core” y dependiendo de la versión del EMGU, esta puede variar, por ejemplo “opencv_core220.dll”
6. Una vez abierta la librería este programa mostrará las dependencias faltantes; las cuales dependiendo del equipo pueden variar, por ejemplo la librería cuda.dll, en caso que la marque solo se tiene que descargar y copiar en su carpeta system32.
7. Una vez copiada las .dll faltantes del “open_core” ya se podrá ejecutar cualquier elemento de los EMGUCV en su c#.

ANEXO B

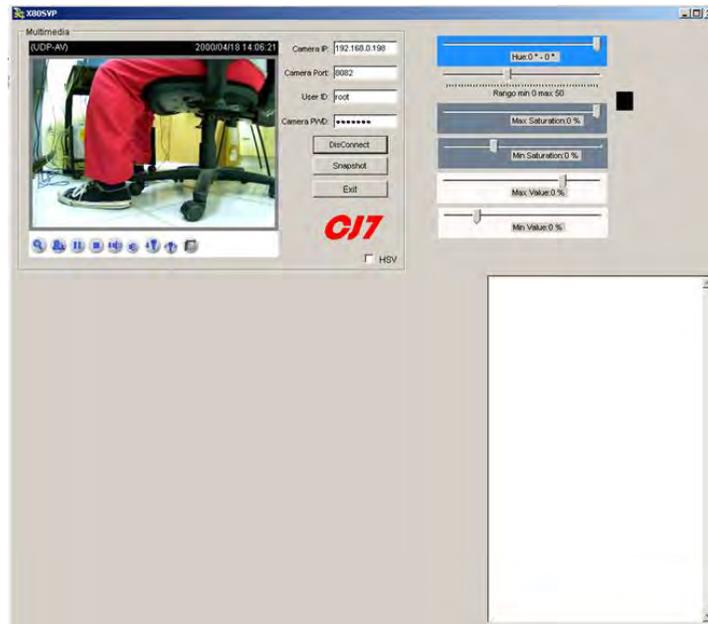
Los siguientes pasos describen el proceso para el manejo de la aplicación que viene con este documento.

1. Inicialice el proyecto X80SVP
2. Ejecútelo y se mostrará una imagen como en la imagen ANEXO B. 5.



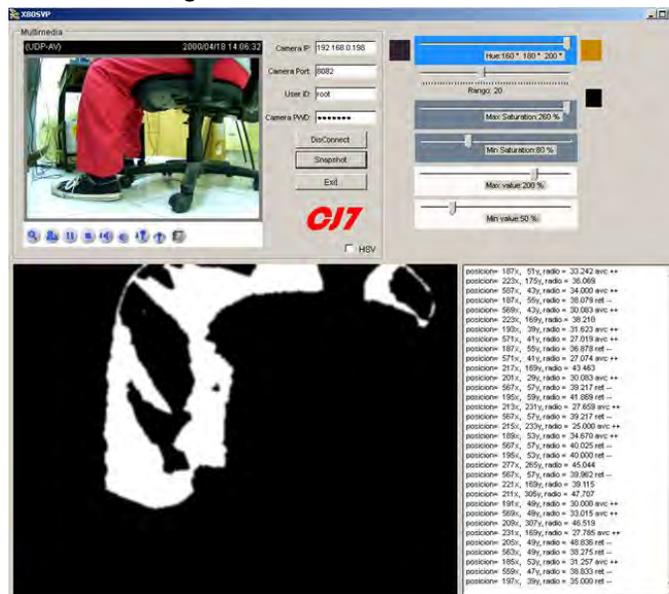
ANEXO B. 1 Aplicación Final.

3. Oprima el botón Conectar a no ser que cambien los datos de la cámara a utilizar, se deberá mostrar una imagen como la imagen ANEXO B. 2.



ANEXO B. 2 Cámara Conectada.

- Al presionar el botón de Snapshot se mostrara en pantalla la imagen binaria como se muestra en la imagen ANEXO B. 3.



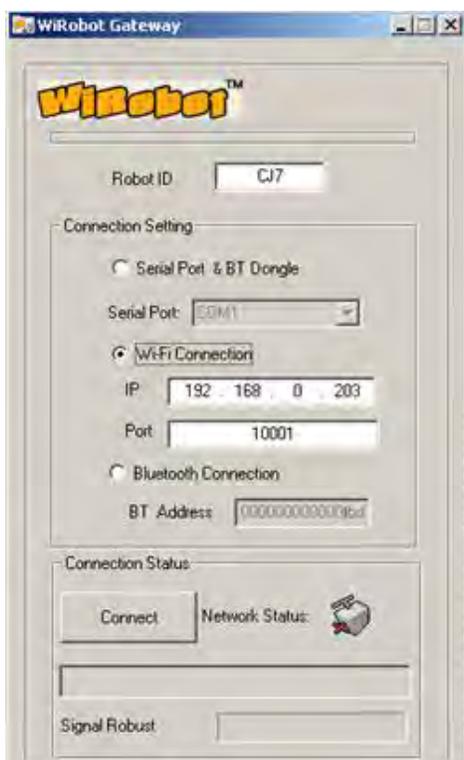
ANEXO B. 3 Imagen Binaria.

- Si habilita la Opción HSV, podrá cambiar la pantalla por la del seguimiento con radios como la imagen ANEXO B. 4.



ANEXO B. 4 Pantalla HSV.

6. Luego solo tiene que abrir la aplicación Gateway y poner el nombre de letras rojas de la aplicación final “CJ7”; así como la dirección de su dispositivo y su puerto o el que trae por defecto, como se muestra en la imagen ANEXO B. 5.



ANEXO B. 5 Interfaz Gateway.

ANEXO C

```
// Importamos las librerías a requerir en el programa
```

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
using Microsoft.DirectX.DirectInput;
using Microsoft.DirectX;
using System.Drawing.Imaging;
using System.Threading;
using System.Drawing.Drawing2D;
using System.IO;
using System.Timers;
using System.Linq;
using System.Globalization;

using Emgu.CV;
using Emgu.CV.CvEnum;
using Emgu.CV.Structure;
using Emgu.CV.UI;

namespace DrRobot.X80SVP
{
    public partial class seguimiento : Form
    {
        public seguimiento()
        {
            InitializeComponent();
        }

        private void Form1_Load(object sender, EventArgs e)
        {
            myRobot.connectRobot(name.Text);
        }

        /*
        try
        {
            webc = new Capture();
        }
        */
    }
}
```

```
}
catch (NullReferenceException exepcion)
{
textBox1.Text = exepcion.Message;
return;
}
Application.Idle += btnSnapshot_Click;
band = true;

*/

}
privatevoid btnExit_Click(object sender, EventArgs e)
{
Close();
}

privatevoid btnCameraConnect_Click(object sender, EventArgs e)
{
if (btnCameraConnect.Text == "Connect")
{

if (axVitaminCtrl1.ControlStatus == VITAMINDECODERLib.EControlStatus.ctrlConnecting
||
axVitaminCtrl1.ControlStatus == VITAMINDECODERLib.EControlStatus.ctrlRunning)
{
axVitaminCtrl1.Disconnect();
return;
}

axVitaminCtrl1.UserName = textBoxCameraID.Text;
axVitaminCtrl1.Password = textBoxCameraPWD.Text;
axVitaminCtrl1.Url = ""; // use the default value of the control for each type

axVitaminCtrl1.RemoteIPAddr = textBoxCameraIP.Text;
long lPort = System.Convert.ToInt32(textBoxCameraPort.Text);
if (lPort <= 0 || lPort > 65535)
lPort = 80;
axVitaminCtrl1.HttpPort = (int)lPort;
axVitaminCtrl1.AutoAVISettings = true;
axVitaminCtrl1.ConnectionProtocol = VITAMINDECODERLib.EConnProtocol.eProtUDP;
axVitaminCtrl1.ServerModelType =
VITAMINDECODERLib.EServerModelType.esrv7KDualStream;
axVitaminCtrl1.Connect();

}
else
{
axVitaminCtrl1.Disconnect();
btnCameraConnect.Text = "Connect";
}

}
}
```

```

privatevoid axVitaminCtrl1_OnConnectionOK(object sender,
AxVITAMINDECODERLib._IVitaminCtrlEvents_OnConnectionOKEvent e)
{
    btnCameraConnect.Text = "DisConnect";
}

//declaracion de variables de tiempo asi como numero de capturas por segund
int tiempo = 0, val = 0, v = 0, fps = 200;

//declaracion de los vectores de captura
byte[] bData =newbyte [921654];

/*
////////////////////////////////////
Capture webc = null;
bool band = false;
Image<Bgr, Byte> frame;
////////////////////////////////////
*/

private void axVitaminCtrl1_OnConnectionBroken(object sender,
AxVITAMINDECODERLib._IVitaminCtrlEvents_OnConnectionBrokenEvent e)
{
    btnCameraConnect.Text = "Connect";
}

/*
private void Form_cerrado(object sender, FormClosedEventArgs e)
{
    if (webc != null)
    {
        webc.Dispose();
    }
}
*/

//convercion a blanco y negro por medio de distincion de color a enfocar
//declaracion de variable para contador interno en el boton
//declaracion del tiempo limtie de busqueda sin objetivo al girar asi como la
velocidad de giro y de avance
int cont = 1, temporizador = 0, vel = 4, giro = 5, time = 5, x=0, y=0;
Image<Hsv, Byte> original;
Image<Gray, Byte> procesado;
double radio = 0, radiomax = 0;
privatevoid btnSnapShot_Click(object sender, EventArgs e)
{
    //declaracion del timer

```

```

timer1.Enabled = true;
timer1.Interval = 1;
//declaracion de vectores o variables

object vData, vInfo;

//metodo de fotogra mas

axVitaminCtrl1.GetSnapshot(VITAMINDECODERLib.EPictureFormat.ePicFmtBmp, out vData,
out vInfo);

/*
////////////////////
frame = webc.QueryFrame();
vData = frame;
original = new Image<Hsv, byte>(frame.Bitmap);

if (frame == null) return;
*/
////////////////////
//Extrae los datos de vData en tipo Byte
bData = (byte[])vData;
TypeConverter tc = TypeDescriptor.GetConverter(typeof(Bitmap));
//hace referencia a un método
Bitmap img = (Bitmap)tc.ConvertFrom(bData);
//Convierte los valores de bData y los asigna en una nueva variable
Image<Hsv, Byte> impr = newImage<Hsv, byte>(img);
//original es un tipo Image<Hsv, Byte> original;
original = impr;
//realizando el filtrado de HSV
//procesado es un tipo Image<Gray, Byte> original;
procesado = original.InRange(
//valores mínimos para el filtrado
newHsv((trackBar1.Value - trackBar4.Value), trackBar6.Value, trackBar5.Value),
//valores máximos para el filtrado
newHsv((trackBar1.Value + trackBar4.Value), trackBar2.Value, trackBar3.Value));
//colorea los picturbox de los laterales de los trackbar para mostrar los colores
máximos o mínimos
Image<Hsv, Byte> imin = newImage<Hsv, Byte>(30, 30, newHsv((trackBar1.Value -
trackBar4.Value), trackBar6.Value, trackBar5.Value));
//colorea los picturbox de los laterales de los trackbar para mostrar los colores
máximos o mínimos
Image<Hsv, Byte> imax = newImage<Hsv, Byte>(30, 30, newHsv((trackBar1.Value +
trackBar4.Value), trackBar2.Value, trackBar3.Value));
//imprime los picturbox
pictureBox2.Image = imin.ToBitmap();
//imprime los picturbox
pictureBox3.Image = imax.ToBitmap();
//muestra los label con los valores de los trackbar
label1.Text = "Hue:" + (trackBar1.Value - trackBar4.Value) + " ° " +
trackBar1.Value + " ° " + (trackBar1.Value + trackBar4.Value) + " ° ";
label2.Text = "Max Saturation:" + trackBar2.Value + " %";
label6.Text = "Min Saturation:" + trackBar6.Value + " %";
label3.Text = "Max value:" + trackBar3.Value + " %";
label5.Text = "Min value:" + trackBar5.Value + " %";
label4.Text = "Rango: " + trackBar4.Value;

```

```

//método de cerrado
//Obtiene una imagen en bitmap
Image<Gray, Byte> src = newImage<Gray, Byte>(procesado.Bitmap);
//Obtiene sus dimensiones
Image<Gray, Byte> dst = newImage<Gray, Byte>(src.Width, src.Height);
//proceso de cerrado empezando por 5 dilataciones y luego 5 erosiones
StructuringElementEx element = newStructuringElementEx(5, 5, 1, 1,
Emgu.CV.CvEnum.CV_ELEMENT_SHAPE.CV_SHAPE_CROSS);
//proceso de cerrado que se repite 3 veces
CvInvoke.cvMorphologyEx(src, dst, IntPtr.Zero, element, CV_MORPH_OP.CV_MOP_CLOSE,
3);
//Almacena el nuevo valor en procesado de nuevo
procesado = dst;

//Se aplica un suavizado
procesado = procesado.SmoothGaussian(9); //aplicacion de una gaussiana

CircleF[] circulos = procesado.HoughCircles(
newGray(100), //Umbral de canny
newGray(50), //acumulación del Umbral
2, //acumulación de resolución
procesado.Height / 4, //Mínimo de distancia entre pixeles para detectar el circulo
10, //mínimo de radio de círculos
70)[0]; //máximo de radio de círculos
//Iniciamos un foreach para dentro del círculos podamos extraer la posición y el
radio de circulo
foreach (CircleF circulo in circulos)
{
//En caso que las coordenadas del punto cambie reinicia el máximo
if ((x != int.Parse(circulo.Center.X.ToString().PadLeft(4))) && (y !=
int.Parse(circulo.Center.Y.ToString().PadLeft(4))))
{
radiomax = 0;
}
//Si el radio es mayor o igual lo almaceno como máximo
if (radiomax >= double.Parse(circulo.Radius.ToString("###.000").PadLeft(7)))
{
radiomax = double.Parse(circulo.Radius.ToString("###.000").PadLeft(7));
}
//Asignación de variables de x y del radio
x = int.Parse(circulo.Center.X.ToString().PadLeft(4));
y = int.Parse(circulo.Center.Y.ToString().PadLeft(4));
//extracción del radio
radio = double.Parse(circulo.Radius.ToString("###.000").PadLeft(7));

//comparocon el RadioButton maximo

String texto = "";

//SOLO DIBUJA EL CIRCULO MAS GRANDE Y MUEVE EL ROBOT CON RESPECTO AL CIRCULO MAS
GRANDE
//posicionamiento
MCvFont f = newMCvFont(FONT.CV_FONT_HERSHEY_TRIPLEX, 3, 3);

```

```

//giro a la izquierda
if ((x < 200) && (radiomax <= radio))
{
    original.Draw("<", ref f, new System.Drawing.Point(50, 300), newHsv(180, 150,
150));
    //velocidad calculada del motor para giro a la izquierda
    short speed = (short)(7183 - (x * giro));
    //encendiendo motores, recordemos que los motores se ven en lados opuestos
    //por lo tanto cuando tengan el mismo valor giraran de lado contrario
    myRobot.DcMotorPwmTimeCtrAll(speed, speed, 16363, 16363, 16363, 16363, 1);
    CvInvoke.cvCircle(original, newPoint(x, y), 3, newMCvScalar(110, 260, 260), -1,
    LINE_TYPE.CV_AA, 0);
    original.Draw(circulo, newHsv(30, 260, 260), 3);
    texto = " izq <<";
    //anexando al textbox
    textBox1.AppendText("posicion= " + x + "x, " + y + "y, radio = " + radio + texto);
    textBox1.AppendText(Environment.NewLine);
}else

//giro a la derecha
if ((x > 440) && (radiomax <= radio))
{
    original.Draw(">", ref f, new System.Drawing.Point(490, 300), newHsv(180, 150,
150));
    //velocidad calculada del motor para giro a la izquierda
    short speed = (short)(25545+((x - 440) * giro));
    //encendiendo motores, recordemos que los motores se ven en lados opuestos
    //por lo tanto cuando tengan el mismo valor giraran de lado contrario
    myRobot.DcMotorPwmTimeCtrAll(speed, speed, 16363, 16363, 16363, 16363, 1);
    //imprimiendo punto y circulo si amerita la condicion
    CvInvoke.cvCircle(original, newPoint(x, y), 3, newMCvScalar(110, 260, 260), -1,
    LINE_TYPE.CV_AA, 0);
    original.Draw(circulo, newHsv(30, 260, 260), 3);
    texto = " der >>";
    //anexando al textbox
    textBox1.AppendText("posicion= " + x + "x, " + y + "y, radio = " + radio + texto);
    textBox1.AppendText(Environment.NewLine);
}

//se acerca al objetivo si la condición detecta que el radio mínimo es menor que el
rango permitido
//descartando así posibles ruidos
if ((radio < 25) && (radio > 15) && (radiomax <= radio) && ((y + radio) > 50) || ((y
+ radio) < 330))
{
    original.Draw("^", ref f, new System.Drawing.Point(300, 50), newHsv(180, 150,
150));
    //velocidad calculada del motor para avanzar
    short speed = (short)(25545+(y * vel + (radio*10)));
    short speed2 = (short)(7183-(y * vel + (radio * 10)));
    //encendiendo motores, de diferentes valores giran al mismo lado
    myRobot.DcMotorPwmTimeCtrAll(speed, speed2, 16363, 16363, 16363, 16363, 1);
    //imprimiendo punto y circulo si amerita la condicion
    CvInvoke.cvCircle(original, newPoint(x, y), 3, newMCvScalar(10, 260, 260), -1,
    LINE_TYPE.CV_AA, 0);
}

```

```

original.Draw(circulo, newHsv(110, 260, 260), 3);
texto = " avc ++";
//anexando al textbox
textBox1.AppendText("posicion= " + x + "x, " + y + "y, radio = " + radio + texto);
textBox1.AppendText(Environment.NewLine);
}else

//retrocede del objetivo si la condición del radio máximo lo amerita para descartar
ruidos así como la posición
if ((radio > 60) && (radio < 70) || ((y + radio) > 330) && (radiomax >= radio))
{
original.Draw("v", ref f, new System.Drawing.Point(300, 400), newHsv(180, 150,
150));
//velocidad calculada del motor para retroceder
short speed = (short)(25545 + (y * vel + (radio * 10)));
short speed2 = (short)(7183 - (y * vel + (radio * 10)));
//encendiendo motores, de diferentes valores giran al mismo lado
myRobot.DcMotorPwmTimeCtrAll(speed2, speed, 16363, 16363, 16363, 16363, 1);
//imprimiendo punto y circulo si amerita la condicion
CvInvoke.cvCircle(original, newPoint(x, y), 3, newMCvScalar(110, 260, 260), -1,
LINE_TYPE.CV_AA, 0);
original.Draw(circulo, newHsv(0, 260, 260), 3);
texto = " ret --";
//anexando al textbox
textBox1.AppendText("posicion= " + x + "x, " + y + "y, radio = " + radio + texto);
textBox1.AppendText(Environment.NewLine);
}
//centrado el objetivo para apagar el movimiento
////////150////////
////////-----////////
//250/|x|/390//
////////-----////////
////////330////////

if ((x > 250) &&
(x < 390) &&
(radio > 25)&&
(radio < 60))
{
//detención del movimiento de los motores sin apagado
myRobot.DcMotorPwmTimeCtrAll(16363, 16363, 16363, 16363, 16363, 16363, 1);
//reinicio de bandera de tiempo
temporizador = 0;
//condicion que pregunta si el radio es igual que el maximo para diferenciar entre
//posibles filtraciones de ruido
// if (radiomax <= radio){
//imprimiendo punto y circulo si amerita la condicion
CvInvoke.cvCircle(original, newPoint(x, y), 3, newMCvScalar(30, 260, 260), -1,
LINE_TYPE.CV_AA, 0);
original.Draw(circulo, newHsv(60, 260, 260), 3);
texto = "xObjx";
//anexando al textbox
textBox1.AppendText("posicion= " + x + "x, " + y + "y, radio = " + radio + texto);
textBox1.AppendText(Environment.NewLine);
//textBox1.ScrollToCaret();
// }
}
}

```

```
//end foreach
}

if (temporizador == time) {
//detencion del movimeinto de los motores sin apagado
myRobot.DcMotorPwmTimeCtrAll(16363, 16363, 16363, 16363, 16363, 16363, 1);
//reinicio de bandera de tiempo
temporizador = 0;
}
else
temporizador++;

//cambio de tipo de color
if (checkBox1.CheckState == CheckState.Checked)
pantalla.Image = original;
else
pantalla.Image = procesado;
}
}
}
```