

**LOCALIZACIÓN DINÁMICA DE MÓVILES Y OBSTÁCULOS EN UNA ESCENA
CONTROLADA PARA APLICACIONES EN ROBOTICA**



Autores:

**YAMIT DANILO AMAYA QUINTERO
JOHN ALEXANDER RUIZ HERNANDEZ**

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FÍSICO-MECÁNICAS
ESCUELA DE INGENIERÍAS ELÉCTRICA, ELECTRÓNICA Y
TELECOMUNICACIONES
BUCARAMANGA
MARZO DE 2005**

**LOCALIZACIÓN DINÁMICA DE MÓVILES Y OBSTÁCULOS EN UNA ESCENA
CONTROLADA PARA APLICACIONES EN ROBOTICA**



Autores:

**YAMIT DANILO AMAYA QUINTERO
JOHN ALEXANDER RUIZ HERNANDEZ**

TRABAJO DE GRADO

Director:

Dr. Techn. ROBERTO MARTINEZ ANGEL

Codirector:

MSC(c) DIEGO ALEXANDER TIBADUIZA BURGOS

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FÍSICO-MECÁNICAS
ESCUELA DE INGENIERÍAS ELÉCTRICA, ELECTRÓNICA Y
TELECOMUNICACIONES
BUCARAMANGA
MARZO DE 2005**

CONTENIDO

	pág.
INTRODUCCIÓN.....	1
1. GENERALIDADES DEL ANÁLISIS DINÁMICO DE IMAGENES	3
1.1 APLICACIONES DEL ANÁLISIS DINÁMICO DE IMÁGENES	3
1.2 ETAPAS FUNDAMENTALES DE UN SISTEMA DE VISIÓN COMPUTACIONAL	4
1.2.1 Adquisición de imágenes	5
1.2.2 Pre –procesamiento	5
1.2.3 Segmentación	6
1.2.4 Representación y descripción.....	6
1.2.5 Reconocimiento e interpretación	6
1.3 ETAPAS FUNDAMENTALES DE LA APLICACIÓN IMPLEMENTADA.....	7
2. ADQUISICIÓN Y PRE-PROCESAMIENTO	8
2.1 DESCRIPCIÓN DE LA ESCENA	8
2.2 ELEMENTOS DE ADQUISICIÓN	9
2.2.1 Cámara.	10
2.2.2 Tarjeta digitalizadora	10
2.3 ALGORITMO DE ADQUISICIÓN Y CONFIGURACIÓN DE LA TARJETA DIGITALIZADORA.....	10
2.4 PRE-PROCESAMIENTO	14

3. DETECCIÓN Y SEGMENTACION DE MOVIMIENTO	15
3.1 MÉTODO BASADO EN COLOR	15
3.1.1 Modelos xyY y HLS	16
3.1.2 Algoritmo de color	18
3.2 MÉTODO BASADO EN GRADIENTES	22
3.2.1 Algoritmo de Jain	22
3.2.2 Algoritmo basado en gradientes	23
3.3 MÉTODO BASADO EN CORRELACIÓN.	26
3.3.1 Principios fundamentales	26
3.3.2 Algoritmo basado en Correlación	27
4. LOCALIZACIÓN E IDENTIFICACION	30
4.1 LOCALIZACIÓN DE OBJETOS EN LA IMAGEN	30
4.2 CALIBRACIÓN DE LA CÁMARA	31
4.3 CORRECCIÓN DE DISTORSIONES	34
4.4 TRANSFORMACIÓN DE PÍXELES A CENTÍMETROS	36
4.5 ALGORITMO DE IDENTIFICACIÓN	36
4.6 CÁLCULO DE ORIENTACIÓN DEL MÓVIL	40
4.6.1 Sistema de referencia fijo.....	41
4.6.2 Sistema de referencia variable	42
5. PRUEBAS Y ANÁLISIS DE RESULTADOS	43
5.1 POSICIÓN	44
5.2 ANÁLISIS COMPARATIVO DE RESULTADOS PARA LA PRUEBA DE POSICIÓN	47
5.3 ORIENTACIÓN	48

5.4 SEGUIMIENTO	51
5.5 TIEMPO DE LATENCIA.....	53
6. CONCLUSIONES Y OBSERVACIONES	56
7. RECOMENDACIONES PARA TRABAJOS FUTUROS	59
8. REFERENCIAS BIBLIOGRAFICAS.....	60

LISTA DE FIGURAS

	pág.
Figura 1. Etapas fundamentales de un sistema de visión computacional	4
Figura 2. Diagrama de bloques de la aplicación implementada	7
Figura 3. Distribución general de la escena.....	9
Figura 4. Elementos de adquisición.....	9
Figura 5. Diagrama de flujo para algoritmo de adquisición.....	11
Figura 6. Adquisición de video y frames	13
Figura 7. diagrama cromático CIE 1931 para un observador de 2°.....	16
Figura 8. observador de 2°.	17
Figura 9. espacio y color HLS.....	17
Figura 10. Diagrama de flujo para Algoritmo de Color.....	20
Figura 11. Segmentación del color amarillo con modelo HLS.	21
Figura 12. Segmentación del color naranja con modelo xyY.....	21
Figura 13. Diagrama para Algoritmo basado en Gradientes.....	23
Figura 14. Operaciones del algoritmo de Jain	24
Figura 15. Segmentación de movimiento con gradientes.....	25
Figura 16. Diagrama del algoritmo de correlación	28
Figura 17. Resultados del algoritmo de correlación.....	29
Figura 18. Distorsiones en las lentes.....	31

Figura 19.	Modelo de distorsión de la cámara.....	33
Figura 20.	Modelo Pin-Hole de la cámara	34
Figura 21.	Diagrama de flujo para algoritmo de identificación.....	37
Figura 22.	Dimensiones de los móviles	38
Figura 23.	Identificación de un móvil	38
Figura 24.	Diagrama par el algoritmo alternativo	40
Figura 25.	Sistema de referencia fijado en el centro de la escena	41
Figura 26.	Sistema de referencia variable	42
Figura 27.	Posiciones usadas para determinar la distribución del error.	45
Figura 28.	Distribución 3D del error para el algoritmo basado en el modelo de color HLS	46
Figura 29.	Distribución comparativa del error de posición utilizando los algoritmos implementados	48
Figura 30.	Sistema polar para la prueba de orientación	49
Figura 31.	Comparación gráfica del error en orientación.....	50
Figura 32.	Seguimiento de una trayectoria recta	51
Figura 33.	Seguimiento de una trayectoria circular	52

LISTA DE TABLAS

pág.

Tabla 1.	Resultados para el algoritmo basado en el modelo de color HLS.....	45
Tabla 2.	Resultados comparativos para el error de posición.	47
Tabla 3.	Error de orientación para cada uno de los algoritmos implementados.	49
Tabla 4.	Tiempo de latencia.....	53
Tabla 5.	Velocidad máxima de un objeto para ser identificado	54

LISTA DE ANEXOS

	pág.
Anexo A. Resultados de las pruebas de posición y orientación.....	57
Anexo B. Principales características de la tarjeta digitalizadora DFG/Compress PIRANHA y de la cámara DFK 50H13.....	66
Anexo C. Instalación e interfase grafica de “FENIX”.....	78

RESUMEN

TÍTULO: LOCALIZACIÓN DINÁMICA DE MÓVILES Y OBSTÁCULOS EN UNA ESCENA CONTROLADA PARA APLICACIONES EN ROBÓTICA*

AUTORES: AMAYA QUINTERO, YAMIT DANILO, y, RUIZ HERNÁNDEZ, JOHN ALEXANDER**

PALABRAS CLAVES: Análisis dinámico de imágenes, color, correlación, distorsiones, gradientes, identificación, localización, segmentación.

DESCRIPCIÓN:

En el documento aquí presentado se describe como por medio de procesamiento digital de imágenes y algoritmos de detección de movimiento, se determina la localización de objetos móviles (máximo seis) en un escenario controlado (condiciones de iluminación y colores de la escena) donde están presentes algunos obstáculos cuya localización también es encontrada.

Inicialmente se exponen las generalidades del análisis dinámico de imágenes como rama de la visión computacional, relacionándolo con las etapas implementadas en este trabajo. El segundo capítulo explica la forma en que se realizó la adquisición de las imágenes, su pre-procesamiento y la distribución espacial del escenario de trabajo. En el capítulo tres se tratan los métodos para la detección y segmentación del movimiento. La corrección de las distorsiones y la localización de los objetos de interés en el espacio 3D se expone en el capítulo cuatro. Por último las pruebas y los resultados obtenidos son descritos en el capítulo cinco y las conclusiones son descritas en el capítulo seis.

Las pruebas realizadas arrojaron resultados con un error de posición y de orientación inferiores a 1 cm y 2° respectivamente. Lo anterior evidencia la confiabilidad en la información de localización obtenida con los tres algoritmos y la utilidad del proceso de calibración para corregir las distorsiones de la cámara utilizada en el proyecto.

* Trabajo de Grado.

** Facultad de Ingenierías Físico-mecánicas. Ingeniería Electrónica. Roberto Martínez Ángel.

ABSTRACT

TITLE: DYNAMIC LOCALIZATION OF MOBILES AND OBSTACLES IN A CONTROLLED SCENARIO FOR ROBOTICS APPLICATIONS.*

AUTHORS: AMAYA QUINTERO, YAMIT DANILO, and, RUIZ HERNÁNDEZ, JOHN ALEXANDER**

KEY WORDS: Dynamic analysis of images, color, correlation, distortions, gradients, identification, localization, segmentation.

DESCRIPTION:

In this document is described how by digital image processing and motion detection algorithms it can be determined the localization of mobile objects (up to six) and obstacles in a controlled scenario (lightning conditions and scene colors).

Initially, the generalities of dynamic analysis of images as a branch of computational vision are explained, with relation to the stages implemented in this project. The second chapter explains the process of image acquisition, pre-processing and the spatial distribution of the staged scenario. Throughout chapter three, the various methods for motion detection and segmentation are explained. The lens distortion corrections and the localization of the objects in the 3D space are explained in chapter four. The tests and obtained results are described in chapter five, and the conclusions are listed in chapter six.

The carried out tests threw results with a position error and orientation error smaller to 1 cm and 2° respectively. The above-mentioned evidences the reliability in the localization information obtained with the three algorithms and the utility of the calibration process to correct the camera distortions used in the project.

* Work of Grade.

** Faculty of Engineering Physical-mechanics. Engineering Electronic. Roberto Martínez Ángel.

INTRODUCCIÓN

Reconocer y localizar objetos es una necesidad que se ha intentado solucionar utilizando diferentes métodos. Uno de los más avanzados se basa en la percepción de objetos descrita en la teoría de la visión humana. Este método que con el tiempo se vuelve más sofisticado al encontrar nuevas ideas, ingresa en un área de amplio estudio en la actualidad llamada visión computacional.

La visión computacional ha intentado reconocer objetos dentro de una escena generalmente estática por medio de un computador y una cámara logrando así relacionar el mundo que lo rodea con un lenguaje de programación aplicado al procesamiento de imágenes. Ahora bien, el entorno en el cual el hombre vive es dinámico, lo cual plantea un nuevo reto en el estudio de la visión computacional, ya que la información contenida en una secuencia de imágenes requiere un nuevo análisis que le permita extraer información en tiempo real de objetos en movimiento dentro de una escena para aplicaciones en diferentes campos como la medicina, biología, seguridad, automatización industrial, etc. Todo esto dio origen a una rama de la visión computacional llamada ANÁLISIS DINÁMICO DE IMÁGENES.

“LOCALIZACIÓN DINÁMICA DE OBJETOS EN MOVIMIENTO PARA APLICACIONES EN ROBOTICA” es un trabajo de grado desarrollado con el fin de avanzar en la investigación del análisis dinámico de imágenes para lograr una interacción mas eficiente entre el movimiento de los objetos en una escena y su localización por medio de una computadora, lo cual adquiere importancia en aplicaciones relacionadas con robótica móvil y planeación de trayectorias; por tal motivo se espera que lo expuesto en este documento pueda ser utilizado para estudios posteriores, especialmente como apoyo para proyectos de investigación en planeamiento de trayectorias de robots móviles, que se desarrollen en la

Escuela de Ingenierías Eléctrica, Electrónica y Telecomunicaciones de la Universidad Industrial de Santander.

De igual forma, el proyecto constituye un aporte significativo al desarrollo y fortalecimiento del Grupo de Investigación en Control, Electrónica, Modelado y Simulación (CEMOS) en el área de Robótica y Procesamiento Digital de Imágenes.

La detección y extracción de los objetos de interés en la escena, es de gran importancia en este trabajo, ya que es la etapa que finalmente brinda la información útil presente en las imágenes. Esta etapa del proyecto se describe en el capítulo 3. Para el efecto se llevó a cabo con anterioridad el proceso de adquisición de las imágenes, su pre-procesamiento y la distribución espacial del escenario de trabajo, lo cual se presenta en el capítulo 2. Una vez finalizada la extracción de los objetos de interés se procede a su localización en el espacio 3D, para lo cual se hace necesario la corrección de las distorsiones en la imagen causadas por la cámara y el lente como se describe en el capítulo 4. Por último las pruebas y los resultados obtenidos son descritos en el capítulo cinco y las conclusiones en el capítulo seis. También se estudia las generalidades del análisis dinámico de imágenes como rama de la visión computacional, relacionándolo con las etapas desarrolladas en este trabajo, en el capítulo 1.

1. GENERALIDADES DEL ANÁLISIS DINÁMICO DE IMÁGENES

El análisis del movimiento o análisis dinámico de imágenes es una aplicación de la visión computacional que permite realizar el seguimiento y localización de los objetos en movimiento presentes en una secuencia de imágenes.

Existen tres grandes grupos de problemas relacionados con el movimiento desde un punto de vista práctico [12]:

- ✓ La detección del movimiento. Es el problema más simple, se trata de registrar cualquier movimiento detectado.
- ✓ La detección y localización de los objetos en movimiento. Es un problema derivado del anterior, la diferencia radica en la localización de tipo espacial que se le realiza al objeto en movimiento, esto permite una relación entre el mundo real y la secuencia de imágenes adquiridas. Este problema es el que se soluciona con los algoritmos implementados en el presente proyecto
- ✓ Obtención de propiedades 3D de los objetos a partir de un conjunto de imágenes adquiridas en distintos instantes de tiempo.

1.1 APLICACIONES DEL ANÁLISIS DINÁMICO DE IMÁGENES

Las aplicaciones del análisis dinámico de imágenes se extienden a muchas áreas en las cuales la visión computacional es utilizada, se destacan entre ellas:

- ✓ Vigilancia: monitoreo de tráfico, detección de intrusos, etc.
- ✓ Robots móviles y vehículos autónomos: detección de obstáculos, planeamiento de trayectorias en un entorno de navegación.
- ✓ Manipuladores robóticos: posicionamiento exacto de brazos robóticos para manipular objetos.

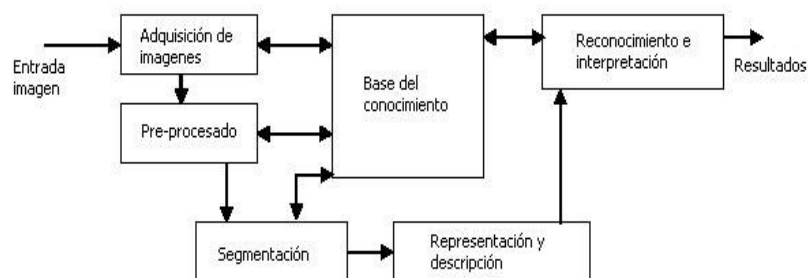
En las aplicaciones enunciadas anteriormente, el presente proyecto se ubica en el planeamiento de trayectorias en un entorno de navegación no siempre conocido, lo cual hace que los algoritmos de detección de movimiento y localización procesen las imágenes obtenidas en un tiempo despreciable, para implementar así una aplicación de planeación de trayectorias en tiempo real para futuros proyectos en el área de robótica.

1.2 ETAPAS FUNDAMENTALES DE UN SISTEMA DE VISIÓN COMPUTACIONAL

Al ser parte de la visión computacional, el análisis dinámico de imágenes se realiza siguiendo las mismas etapas en las que aplicando procesamiento digital de imágenes se extrae la información necesaria para cumplir con la tarea de localización y seguimiento.

En la figura 1 se muestra un esquema de las etapas fundamentales de un sistema de visión computacional.

Figura 1. Etapas fundamentales de un sistema de visión computacional



GONZALEZ, Rafael y WOODS, Richard. Tratamiento digital de imágenes. Estados Unidos de America: Editorial Addison-Wesley iberoamericana.1996

A continuación se describe cada una de estas etapas, resaltando en cada uno los conceptos básicos del procesamiento de imágenes.

1.2.1 Adquisición de imágenes

En esta etapa se obtienen las imágenes visuales. Dos elementos son necesarios para adquirir imágenes digitales de video, el primero es un dispositivo físico que sea sensible a la banda visible del espectro electromagnético y que produzca una señal eléctrica proporcional al nivel de energía percibido, y el segundo es un digitalizador capaz de convertir la señal eléctrica del dispositivo en una forma digital, aunque en la actualidad existen cámaras que entregan ya la señal digital directamente al computador [4].

1.2.2 Pre –procesamiento

La entrada y salida de esta etapa son imágenes con valores de intensidad similares a los datos originales capturados por el digitalizador. El objetivo principal del pre-procesamiento es una mejora de los datos de la imagen suprimiendo los efectos de ruido e incrementando las características deseadas para un exitoso procesamiento.

Aunque los métodos para realizar el pre-procesamiento dependen de la aplicación que se va a desarrollar, se destacan entre otros la utilización de filtros espaciales, que eliminan el ruido presente en la imagen capturada y la ecualización del histograma que mejora la calidad de la imagen variando la distribución de los niveles de gris o color.

1.2.3 Segmentación

En esta etapa los objetos de interés en la imagen son separados del fondo para su posterior estudio. Los métodos de segmentación utilizados varían de una aplicación a otra y en el caso del análisis dinámico de imágenes, los métodos de segmentación utilizados van orientados a separar sólo los objetos que se encuentran en movimiento dentro de una escena.

1.2.4 Representación y descripción

La representación es el proceso mediante el cual se obtienen características convenientes para representar un objeto segmentado en la escena. En esta etapa se debe elegir si las características de representación adecuadas son de contorno como los bordes y las esquinas de un objeto, o de región completa que comprende el objeto segmentado en su totalidad. Aunque en algunas aplicaciones ambas representaciones son necesarias, la descripción extrae rasgos cuantitativos de interés, con los cuales se puede diferenciar entre una y otra clase de objetos. Con la representación y descripción de objetos se consigue disminuir la cantidad de datos que se van a procesar.

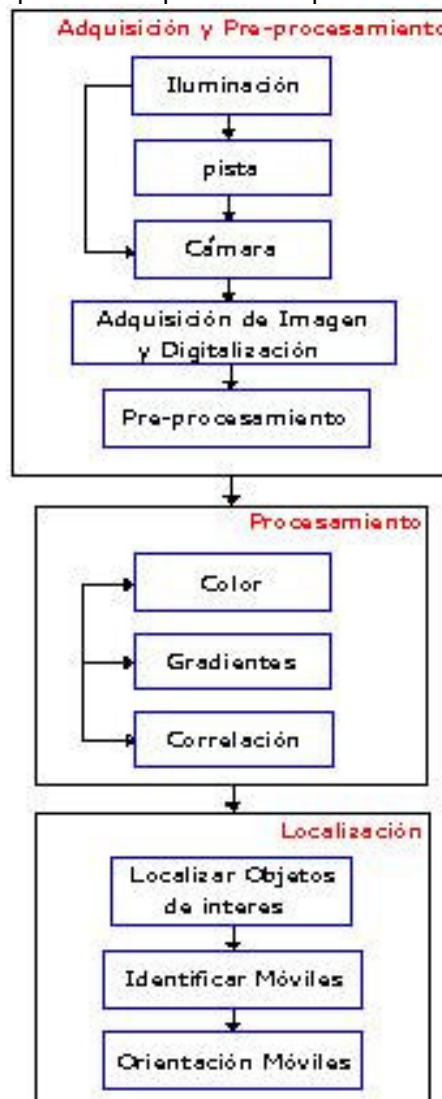
1.2.5 Reconocimiento e interpretación

Con base en la información suministrada por una serie de descriptores utilizados, el reconocimiento asigna una etiqueta a cada elemento. La interpretación se encarga de asignar un significado a un conjunto de objetos reconocidos

1.3 ETAPAS FUNDAMENTALES DE LA APLICACIÓN IMPLEMENTADA

Las etapas implementada en este proyecto, para la localización dinámica de objetos en movimiento se observan en la figura 2. Las etapas de reconocimiento e interpretación y de representación y descripción se encuentran fusionadas en una sola llamada localización. Las demás etapas conservan una estructura semejante a la descrita anteriormente, para un sistema de visión computacional.

Figura 2. Diagrama de bloques de la aplicación implementada



2. ADQUISICIÓN Y PRE-PROCESAMIENTO

La adquisición de imágenes y el pre-procesamiento juegan un papel importante en cualquier aplicación real de visión computacional y en especial de análisis dinámico de imágenes. El tipo de iluminación del escenario, la ubicación de la cámara, las propiedades del lente, las características en la velocidad de captura de la tarjeta digitalizadora, la distribución espacial del escenario, el color de la escena y las características en la calidad de las imágenes obtenidas como el brillo, contraste y ruido presente, son factores que se manipulan desde estas etapas y que definen la correcta detección y localización de los móviles en la escena.

2.1 DESCRIPCIÓN DE LA ESCENA

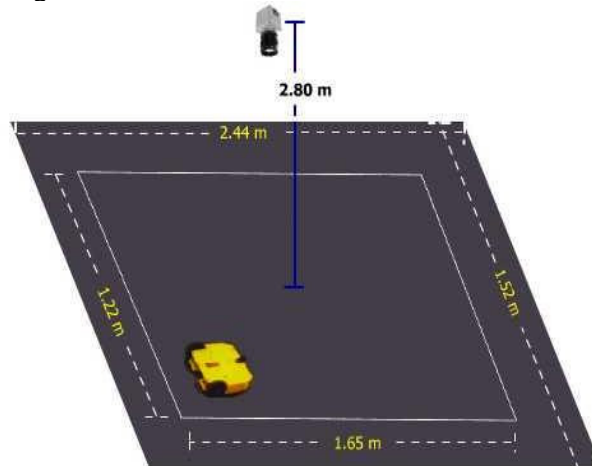
El escenario en general consta de una pista de madera, con un área de 3,7088 m². El color de la pista es negro mate con el objetivo de mejorar los resultados en el proceso de segmentación de los elementos en movimiento y posibles problemas de reflexión de la luz por el uso de pinturas brillantes.

La cámara se ubicó a una altura de 2,80 m desde la pista, procurando que su eje óptico quedara perpendicular al piso, de esta forma se obtiene un área efectiva de captura de 2,013 m².

En cuanto a la iluminación se limitó a las fuentes con las que se cuenta en el laboratorio de robótica, estas fuentes son dos lámparas de techo fluorescentes de 40 W ubicadas en los extremos del laboratorio.

Un esquema de distribución de los principales elementos y algunas dimensiones de importancia en el escenario se observan en la figura 3.

Figura 3. Distribución general de la escena



Diseño de los autores.

2.2 ELEMENTOS DE ADQUISICIÓN

Como se mencionó en el capítulo anterior son dos los elementos fundamentales en un sistema de adquisición: la cámara y la tarjeta digitalizadora. En los siguientes numerales se hará referencia a los dispositivos utilizados en el sistema de adquisición del presente proyecto.

En la figura 4 se observa el esquema de un sistema de adquisición con todos los elementos presentes en este.

Figura 4. Elementos de adquisición



Diseño de los autores

2.2.1 Cámara.

Para la obtención de la señal que se va a digitalizar se hizo uso de la cámara análoga DFK50H13/N existente en el laboratorio de robótica que posee un lente de 8mm y salida de video compuesto. Las especificaciones técnicas más detalladas de la cámara se pueden ver en el anexo B.

2.2.2 Tarjeta digitalizadora

Para realizar la digitalización de las señales provenientes de la Cámara, se utilizó una tarjeta de compresión de video PCI de 32 bits DFG/Compress con una entrada / salida de video compuesto y un conector para obtener la señal de 8 cámaras diferentes. Las especificaciones técnicas mas detalladas de la tarjeta se pueden ver en el anexo B.

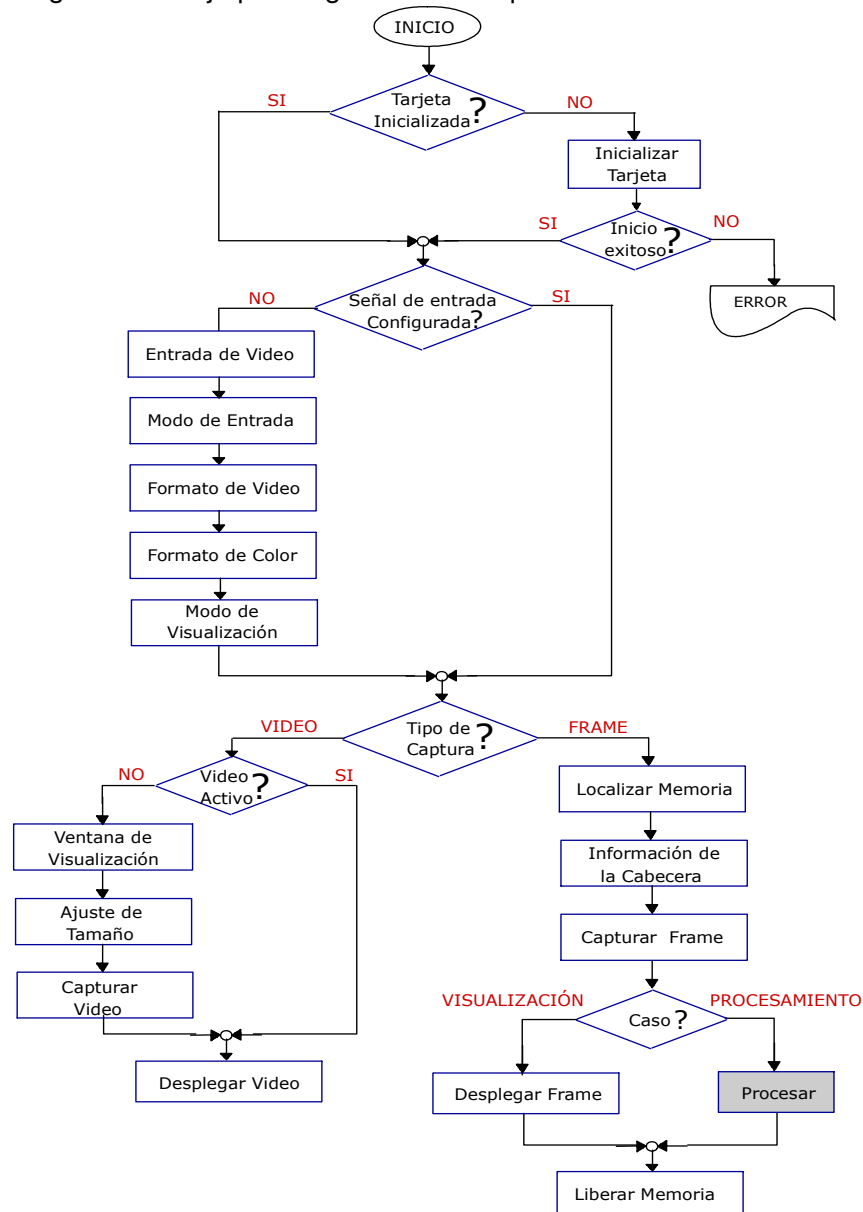
2.3 ALGORITMO DE ADQUISICIÓN Y CONFIGURACIÓN DE LA TARJETA DIGITALIZADORA

Según la aplicación que se desarrolle es necesario implementar estrategias para adquirir las imágenes de forma eficaz y en el caso del análisis dinámico de imágenes es importante que el proceso de adquisición se cumpla en un tiempo razonablemente pequeño en comparación con la velocidad de los objetos en movimiento presentes en la escena. Este requerimiento en la aplicación obliga a implementar de forma más ordenada un algoritmo de adquisición y configuración de la tarjeta que se acomode a las necesidades de los algoritmos de detección de movimiento implementados en el presente proyecto, logrando así, un mayor dinamismo entre la parte de adquisición y procesado.

El algoritmo de adquisición para la tarjeta digitalizadora DFG/Compress se divide en las siguientes partes fundamentales: inicialización del hardware (tarjeta digitalizadora), configuración de parámetros de la señal de video de entrada,

captura y despliegue de video o *frames*¹. Además de los pasos anteriores, siempre que termine el proceso de adquisición hay que desactivar el hardware con el fin de liberar la memoria y la estructura de datos usada por la DFG/Compress. El diagrama de flujo para el algoritmo de adquisición se muestra en la figura 5.

Figura 5. Diagrama de flujo para algoritmo de adquisición



¹ imagen capturada de una secuencia de video.

A continuación se describe detalladamente el algoritmo de adquisición mostrado en la figura 5.

El primer paso para realizar una correcta adquisición consiste en inicializar el hardware, ya que no se puede utilizar alguna otra función sin llevar a cabo este paso. En caso que la tarjeta de adquisición no se haya inicializado correctamente se desplegará un mensaje de error y finalizará el proceso. Si la tarjeta ya ha sido inicializada, estos pasos no se ejecutarán.

Una vez inicializado el hardware, se fijan algunos parámetros de la señal de video de entrada como son:

- a. canal de entrada: la tarjeta soporta cuatro canales de entrada de video.
- b. modo de la señal: video compuesto o S-VHS (Y/C).
- c. formato de video: PAL o NTSC.
- d. formato de color: RGB, YUV. El proceso de adquisición de este proyecto captura imágenes RGB en verdadero color de 24 bits.
- e. modo de visualización: ver anexo B.

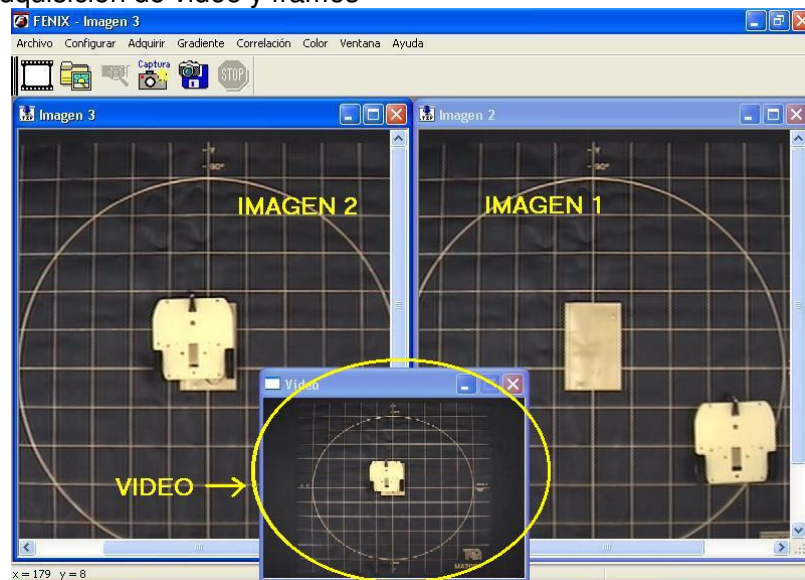
Para la captura del video, se implementó una interfase en la cual presenta la señal de video en una ventana de diálogo de tamaño ajustable [7]. A partir de esta señal, se pueden capturar cuadros que permanecen estáticos en un documento separado.

Para capturar un frame, primero se selecciona un área en memoria dónde se almacenarán los datos de la imagen a adquirir, también se ajusta la cabecera que corresponde a propiedades de la imagen como las dimensiones y el número de bits por píxel. Luego, si lo que se desea son imágenes independientes del proceso de detección y segmentación de objetos en movimiento se efectúa la captura y posterior despliegue, de lo contrario los frames adquiridos no serán desplegados,

lo cual permite optimizar recursos computacionales y disminuir tiempo en el proceso.

Finalmente se libera la memoria ocupada, de tal manera que al capturar los frames siguientes no aumente el tamaño ocupado en memoria. De esta manera, el manejo de memoria tendrá el mismo grado de eficiencia en cuanto al número de bytes usados por la aplicación, sin depender del tiempo que dure el proceso.

Figura 6. Adquisición de video y frames



Diseño de los autores

En la figura 6 se aprecia el resultado de implementar el algoritmo de adquisición descrito anteriormente en visual C++ 6.0, el cual permite capturar video y desplegarlo en un único dialogo y además capturar múltiples imágenes en diferentes vistas. En la figura se puede visualizar el video en la parte inferior central. Este diálogo se puede aumentar o disminuir de tamaño según desee el usuario, también se pueden observar dos imágenes capturadas y desplegadas.

2.4 PRE-PROCESAMIENTO

Una de los objetivos fundamentales del pre-procesamiento en la aplicación desarrollada en el proyecto es eliminar el ruido presente en la imágenes adquiridas, que aunque no es muy notorio si puede afectar en forma drástica los resultados obtenidos en el proceso de detección de móviles. Para solucionar este problema se optó por utilizar filtros de mediana [12], los cuales eliminan en gran parte el ruido presente en la imagen sin alterar de forma notoria la calidad visual de la misma.

3. DETECCIÓN Y SEGMENTACION DE MOVIMIENTO

La etapa de detección y segmentación de movimiento es la más importante en el proyecto aquí presentado. Los algoritmos implementados para el efecto, son de amplio uso en aplicaciones de análisis dinámico de imágenes como es el caso del algoritmo de Jain [6] utilizada en sistemas de vigilancia vial, el cual se basa en la detección de bordes de los vehículos en movimiento en una secuencia de imágenes o algoritmos de reconocimiento de color utilizado en robótica de tipo recreativa como es el caso de la Copa de fútbol robótico.

En este capítulo, se mostrarán los diferentes algoritmos implementados para la detección y segmentación del movimiento, así como una breve fundamentación teórica para cada uno.

3.1 MÉTODO BASADO EN COLOR

Los algoritmos de detección y segmentación de movimiento basados en el reconocimiento del color, distinguen el color de los objetos de interés presentes en la escena para así realizar su segmentación.

En el presente proyecto se implementaron dos modelos de color [12] el xyY y HLS para realizar la detección. La principal razón para elegir estos modelos radica en que la información cromática y la de brillo es separada en componentes distintas, lo cual hace a estos modelos robustos ante los cambios de iluminación que se presenten en el escenario. Además, al estar realizando detección solamente en las componentes que contienen la información cromática, el tiempo en la segmentación disminuye, lo cual disminuye también el tiempo total de procesamiento.

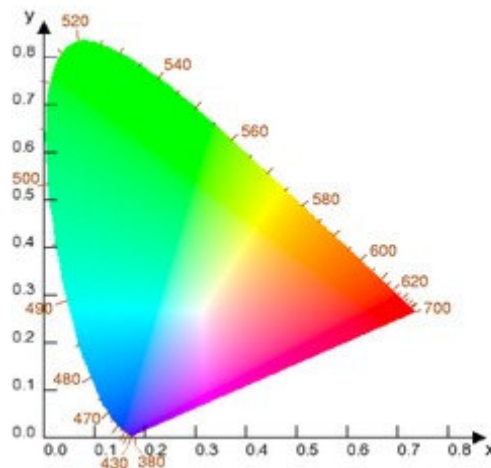
3.1.1 Modelos xyY y HLS

- **Modelo De Color xyY** : en 1931 la *Commission Internationale de l'Éclairage (CIE)*² define tres colores imaginarios base x, y e Y, como una solución al problema que tiene el modelo RGB³ de no poder representar muchos de los colores percibidos por el ojo humano.

El modelo xyY, esta basado en las curvas de respuesta espectral de los tres receptores de color del ojo (observador patrón) las cuales difieren ligeramente de una persona a otra [9]. El CIE ha definido un “observador estándar” basándose en la respuesta espectral media de la población.

Los colores primarios para el observador estándar se encuentran representados en un modelo espacial llamado *diagrama cromático CIE 1931 para un observador de 2°*; este diagrama se observa en la figura 7, además en la figura 8 se observa en qué consiste el concepto de un observador de 2°.

Figura 7. diagrama cromático CIE 1931 para un observador de 2°

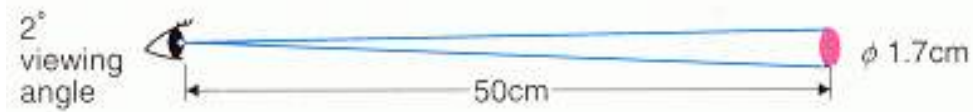


disponible en < <http://en.wikipedia.org/wiki/Color> > [noviembre de 2004]

² Traducción Francesa de “Comisión Internacional sobre iluminantes”, principal organización internacional encargada de los problemas sobre medición del color.

³ Siglas en ingles de Red(rojo), Green(verde) y Blue (azul) colores base en el modelo RGB

Figura 8. observador de 2°.

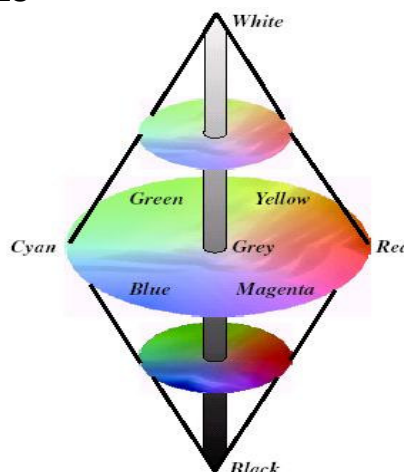


disponible en < <http://home.wanadoo.nl/paulschils/10.00.htm> > [noviembre de 2004]

En diagrama cromático del CIE (figura 7) se puede observar que la información de color se encuentra solamente en dos de sus componentes (x & y), puesto que en la componente Y se representa la claridad de los colores, esto le da una ventaja sobre el modelo RGB en el cual la información de color esta contenida en sus tres componentes.

- **Modelo De Color HLS:** Este modelo utiliza los tres componentes básicos del color: matiz (Hue), saturación (Saturation) y brillo (Luminance). Una representación gráfica incluyendo estas tres propiedades genera el denominado *espacio y color*, el cual se ve en la figura 9.

Figura 9. espacio y color HLS



RUSS John. The Image Processing Handbook. Boca Raton, FL. CRC PRESS, IEEE PRESS. 2002.

En el perímetro del disco están situados los colores en su forma mas pura, es decir con la ausencia de luz blanca. Al variar la distancia al centro del disco, los colores comienzan a mezclarse con la luz blanca, presentándose así los colores pasteles, esto se debe a que la distancia al centro (radio) señala los cambios en la componente de saturación (saturation) de los colores.

El barrido angular que se hace con una línea tomada desde el centro por toda el área del disco corresponde a cambios en el color. Esta coordenada angular corresponde al matiz (hue) del color.

Por ultimo el brillo (luminance), corresponde a la altura del doble cono y tiene un valor de 0 a 1, esto significa que a una altura cero se encuentra el color negro, a una de 0.5 se encuentra un gris de intensidad media, y a una altura de valor 1 se localiza el color blanco puro, lo cual indica que en la línea perpendicular que pasa por el centro de la circunferencia se encuentran los niveles de grises.

3.1.2 Algoritmo de color

A continuación se describe el algoritmo de segmentación por color basado en los modelos HLS y xyY, implementado en el presente proyecto.

Inicialmente se tiene una imagen RGB de la etapa de adquisición y luego se realiza una conversión de RGB al modelo de color seleccionado con anterioridad.

La transformación de RBG a HLS se describe con las siguientes ecuaciones [4]:

$$H = \begin{cases} \theta & \text{si } B \leq G \\ 360 - \theta & \text{si } B > G \end{cases} \quad (1)$$

donde:

$$\theta = \cos^{-1} \left(\frac{\frac{1}{2}[(R-G) + (R-B)]}{\sqrt{(R-G)^2 + (R-B)(G-B)}} \right) \quad (2)$$

$$S = 1 - \frac{3}{R+G+B} [\min(R, G, B)] \quad (3)$$

La transformación de RGB a xyY [9] se describe como:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.4124543 & 0.35758 & 0.180423 \\ 0.2126710 & 0.71516 & 0.072169 \\ 0.0193340 & 0.119193 & 0.950227 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (4)$$

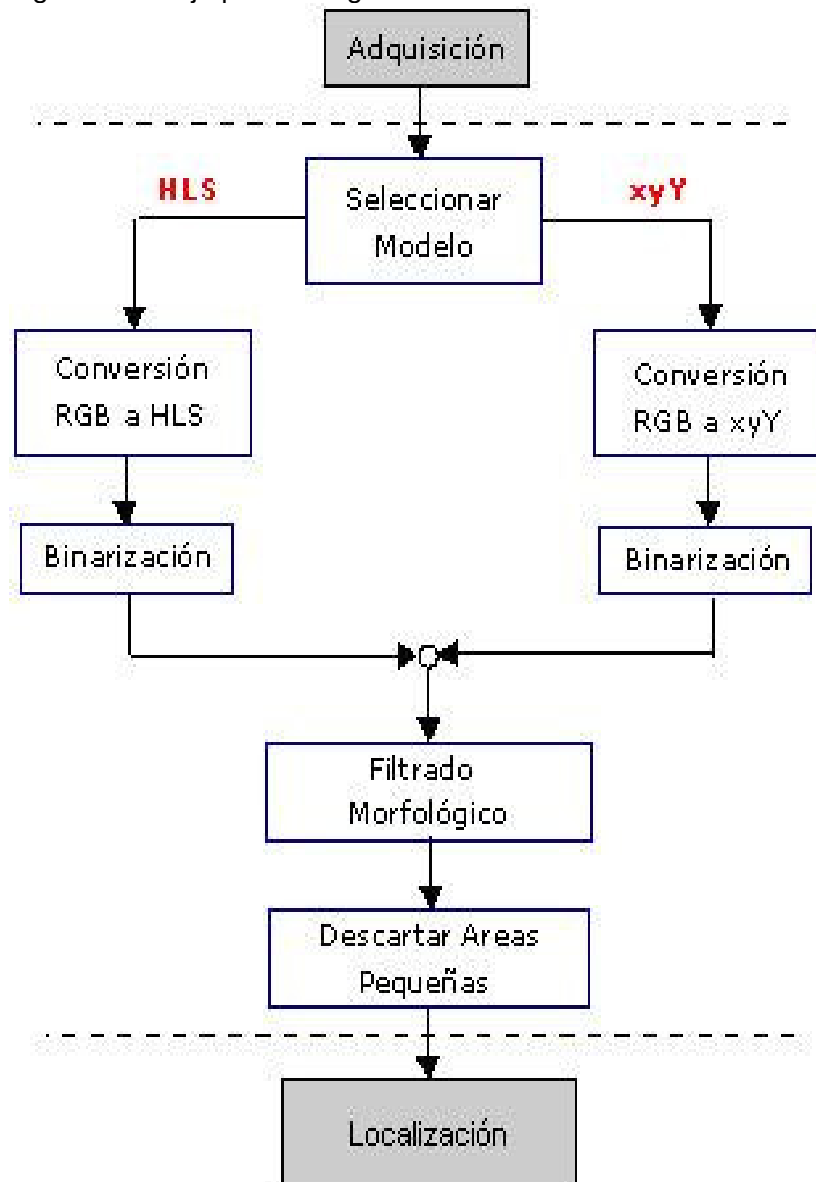
$$x = \frac{X}{X+Y+Z} \quad y = \frac{Y}{X+Y+Z} \quad (5)$$

donde la ecuación 4 corresponde a la transformación de RGB a los valores triestímulo XYZ, para el iluminante D65 [9] y un observador de 2º, y la ecuación 5 es la normalización de las componentes x e y.

Una vez realizada la transformación al modelo de color adecuado, se binariza la imagen obtenida de acuerdo al color que se quiera identificar. A esta imagen binarizada se le aplica un filtrado morfológico basado en erosiones y relleno de huecos [14] con el fin de obtener una buena segmentación de los objetos de interés en la escena. Los umbrales de binarización dependen en su valor de cada modelo de color y hacen parte de la base del conocimiento, es decir de los parámetros con los cuales se realice la segmentación. Finalmente se eliminan falsos candidatos si el área del objeto es menor a un área umbral predeterminada en el algoritmo.

En la figura 10 se muestra el diagrama para el algoritmo de color descrito anteriormente.

Figura 10. Diagrama de flujo para el algoritmo de color



En las figuras 11 y 12 se observa el resultado de aplicar el algoritmo de color a algunas imágenes tomadas en el laboratorio. En el inciso b) de cada figura la imagen ya binarizada presenta ruido debido a condiciones de iluminación en el escenario, en el inciso c) se detalla el resultado final del proceso de detección y segmentación del color elegido después de haber aplicado el filtro morfológico a las imágenes del inciso b).

Cabe resaltar que estas imágenes no fueron adquiridas en el escenario planteado en este proyecto, sino en un escenario no controlado, y puede verse que la iluminación no deja de ser un factor preponderante en el momento de realizar la detección del color, no obstante se observa que con la utilización de los modelos de color implementados en este proyecto este problema se soluciona de forma completa.

Figura 11. Segmentación del color amarillo con modelo HLS.
a) Imagen original. b) Binarización. c) Imagen con filtrado morfológico

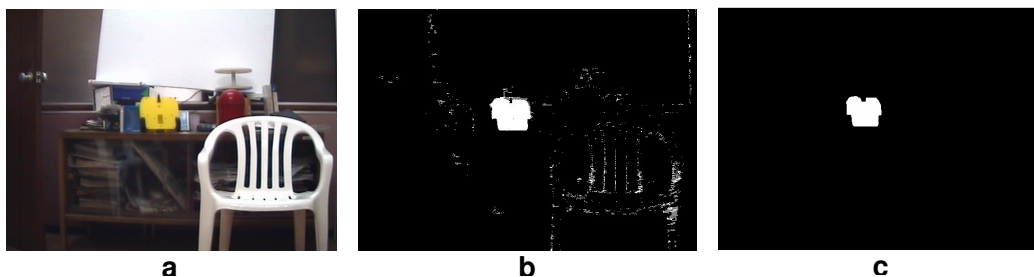
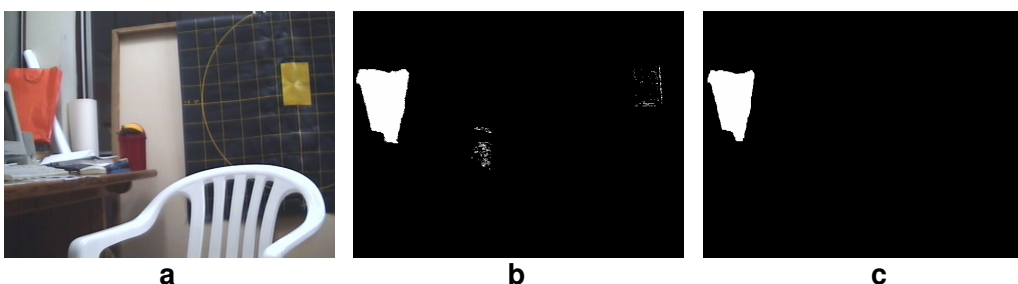


Figura 12. Segmentación del color naranja con modelo xyY
a) Imagen original. b) Binarización. c) Imagen con filtrado morfológico



3.2 MÉTODO BASADO EN GRADIENTES

Estos métodos proveen una solución al problema de estimación de movimiento, a partir de la observación de los cambios en el brillo de una secuencia de imágenes. Aprovechando este fundamento, se puede detectar los bordes de un elemento en movimiento en cualquier escena dinámica.

En la estimación del movimiento por medio de métodos basados en gradientes uno de los principales problemas es la presencia de elementos no móviles que pueden estar en la escena y que interfieren con la correcta detección de los bordes de los elementos en movimiento. Por tal motivo en el proyecto aquí expuesto se implementó un algoritmo que se fundamenta en el algoritmo de Jain el cual soluciona en gran medida este problema.

3.2.1 Algoritmo de Jain

Este algoritmo utiliza dos gradientes, el espacial y el temporal, los cuales se combinan por medio de un operador lógico AND, como se muestra en la siguiente ecuación:

$$E_T(x, y, t) = \frac{dF(x, y, t)}{dS} \bullet \frac{dF(x, y, t)}{dt} \quad (6)$$

En donde $dF(x, y, t)/dS$ y $dF(x, y, t)/dt$ son respectivamente los gradientes espacial y temporal de la función de intensidad $F(x, y, t)$ ⁴. Para obtener el gradiente espacial se puede utilizar cualquier detector de bordes usado en procesamiento digital de imágenes, mientras que para determinar el gradiente temporal una opción consiste en realizar la diferencia entre los niveles de grises de dos imágenes diferentes de la misma escena. Cabe resaltar que una de las imágenes utilizadas en la

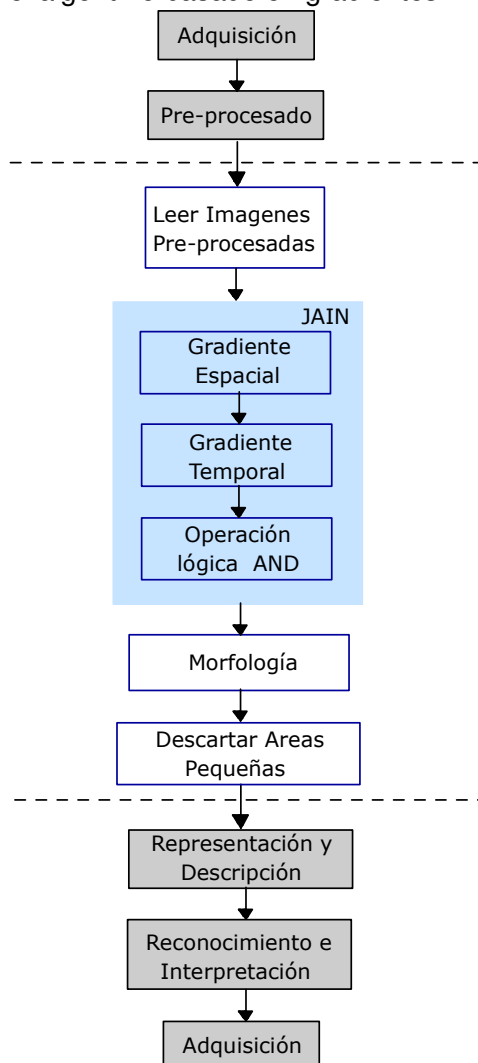
⁴ Corresponde a los niveles de intensidad en cada píxel presente en una imagen.

diferencia de niveles de grises para calcular el gradiente temporal debe ser la misma que se utiliza para calcular el gradiente espacial.

3.2.2 Algoritmo basado en gradientes

Este algoritmo se basa fundamentalmente en el algoritmo de Jain y en algunas operaciones morfológicas como se puede apreciar en la figura 13.

Figura 13. Diagrama para el algoritmo basado en gradientes



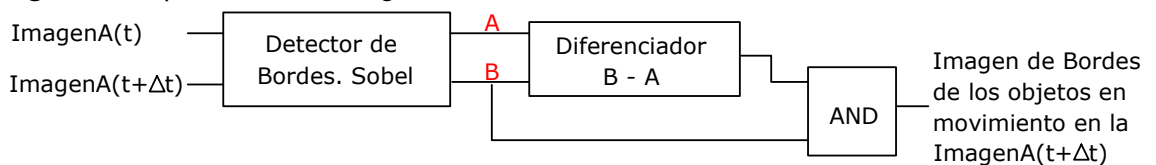
A continuación se describe el algoritmo de detección y segmentación de movimiento basado en gradientes implementado en el proyecto aquí expuesto:

En primera instancia, se leen dos imágenes en escala de grises que provienen de la etapa de pre-procesado, las cuales han sido adquiridas en instantes diferentes de una secuencia de imágenes.

$$\underbrace{\text{imagen } A(t) \dots\dots \text{ imagen } A(t + \Delta t) \dots\dots\dots}_{\text{SECUENCIA DE IMAGENES}}$$

Luego se aplica el método de Jain, el cual consiste inicialmente en hacer una detección de bordes a cada imagen por medio del detector de bordes de sobel[14], después se realiza la diferencia entre la segunda y la primera de estas imágenes de bordes y a este resultado se aplica la operación lógica AND con la segunda imagen de bordes, obteniendo como resultado solamente los bordes del objeto en movimiento en la segunda imagen. El proceso se ilustra en la figura 14.

Figura 14. Operaciones del algoritmo de Jain



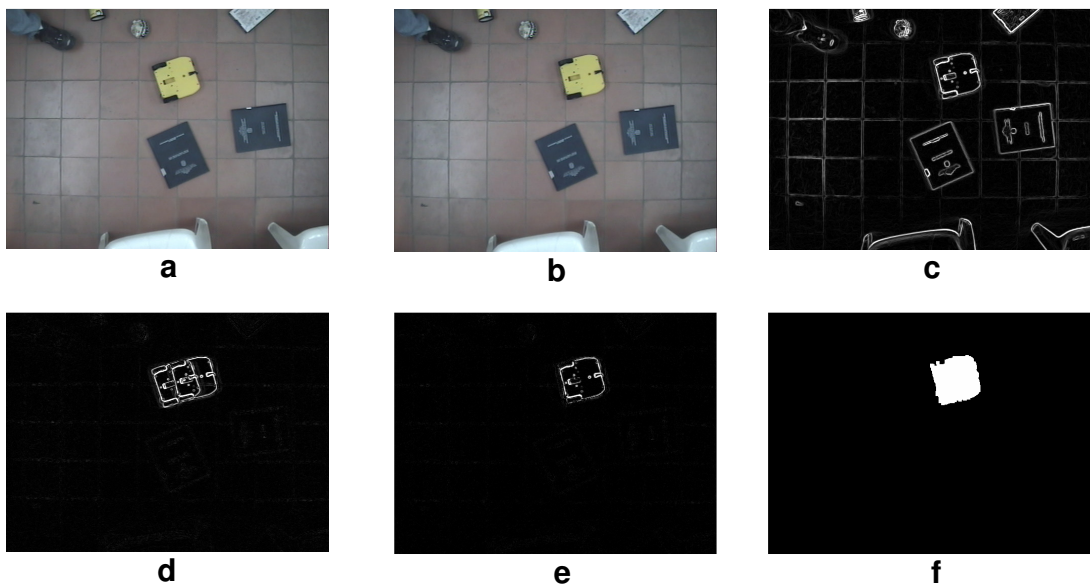
Finalmente por medio de operaciones morfológicas se realiza el relleno de la región correspondiente al objeto en movimiento y además se descartan las regiones con un área menor a un valor umbral fijado previamente.

Uno de los inconvenientes del algoritmo descrito anteriormente es que solo segmenta objetos en movimiento en una escena, por lo tanto, cuando un objeto que esta en movimiento se queda estático o sale de la escena, el algoritmo no

obtendrá resultado alguno. Por esta razón se optó por realizar un algoritmo alternativo con el cual recuperar la información de localización de los objetos que ahora se encuentran fijos o que no se encuentran en la escena. Este algoritmo alternativo se expondrá en detalle en el siguiente capítulo.

Figura 15. Segmentación de movimiento con gradientes.

a) imagen $A(t)$. b) imagen $B(t+\Delta t)$. c) Gradiente espacial. d) Gradiente temporal. e) bordes del objeto en movimiento f) objeto segmentado



En la figura 15 se observa la segmentación del objeto en movimiento (carro amarillo) realizada con el algoritmo basado en gradientes y en una escena no controlada. En los incisos c, d y e de la figura 15 se muestra el resultado de las operaciones del algoritmo de Jain. Finalmente en la figura 15-f se observa el resultado de aplicar operaciones morfológicas a la imagen 15-e obteniendo así la región interna del objeto para su posterior localización.

3.3 MÉTODO BASADO EN CORRELACIÓN.

La correlación es la medida de la relación de igualdad que existe entre dos conjuntos de datos, los valores que toma, indican el grado de semejanza entre ellos.

3.3.1 Principios fundamentales

La correlación aplicada al análisis de imágenes y más específicamente a la localización de objetos en movimiento dentro de una escena, consiste en comparar secciones de una imagen para ver el nivel de semejanza con otra imagen, llamada modelo patrón [8].

La función de correlación normalmente utilizada en procesamiento digital de imágenes tiene la siguiente expresión matemática:

$$r = \sum_{i=1}^{i=N} I_i M_i \quad (7)$$

Donde M es el modelo patrón e I es la imagen a la cual se le va a encontrar la semejanza con el modelo patrón.

Esta función es máxima cuando la parte analizada de la imagen es muy semejante al modelo patrón. Infortunadamente, este método es impreciso si los niveles de brillo en la imagen son muy altos. En este caso es muy posible que la correlación alcance su máximo valor, cuando las imágenes no tienen ninguna correspondencia entre ellas; por tal motivo en el presente proyecto se utilizara la correlación normalizada, como una solución práctica a este inconveniente.

Correlación Normalizada. Es una función más compleja, en la cual el resultado no se ve afectado por cambios lineales en la imagen o valores de los píxeles en el modelo. Además, la función alcanza su valor máximo únicamente cuando la imagen y el modelo tienen una correspondencia exacta [10]. La función de correlación normalizada se muestra a continuación:

$$r = \frac{N \sum IM - \sum I \sum M}{\sqrt{(N \sum I^2 - (\sum I)^2)(N \sum M^2 - (\sum M)^2)}} \quad (8)$$

Donde N corresponde al número de píxeles analizados e I y M corresponden a la imagen y modelo patrón, respectivamente.

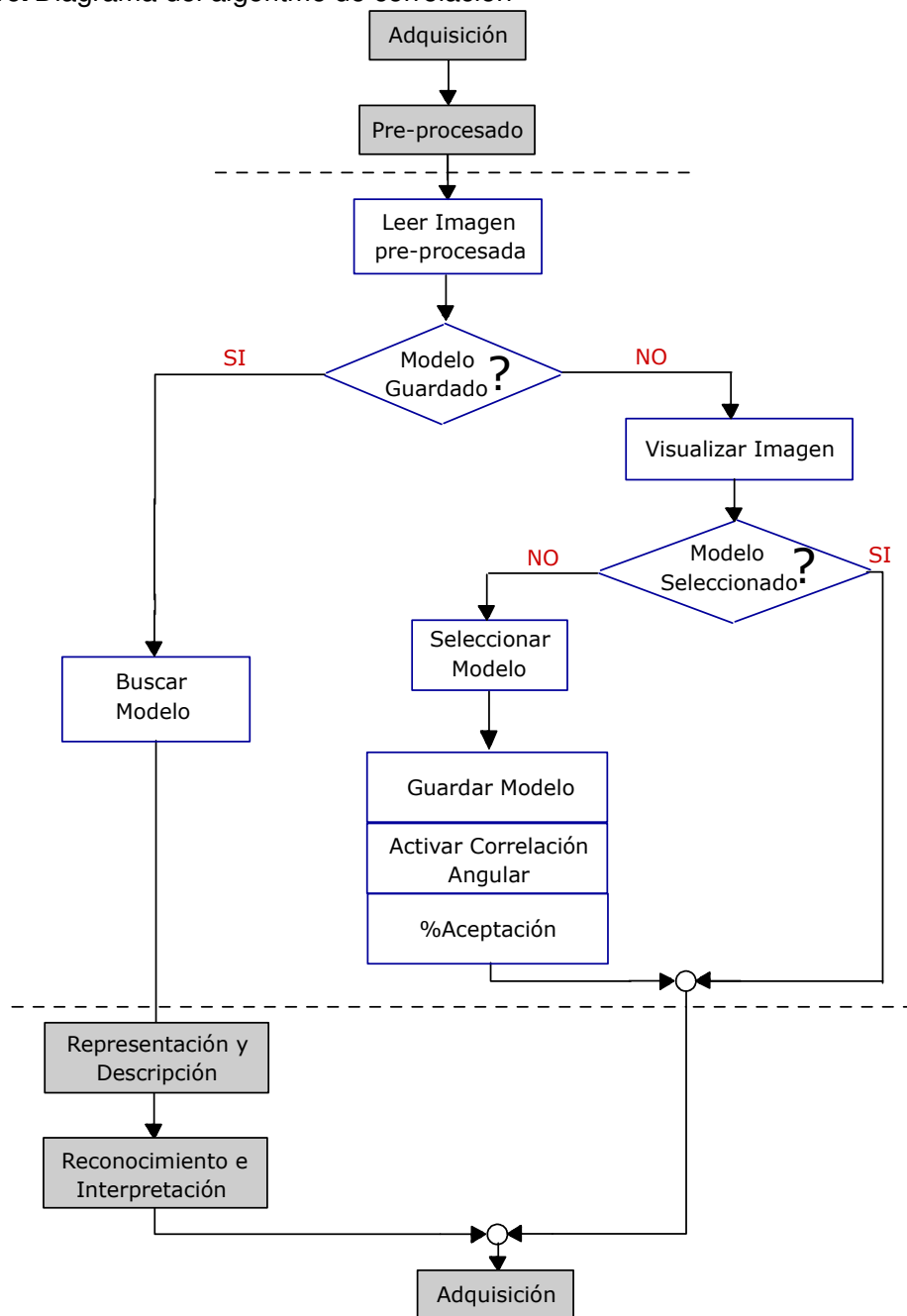
3.3.2 Algoritmo basado en Correlación

El objetivo principal de este algoritmo es seleccionar como modelo patrón, un objeto presente en una primera imagen en escala de grises de la escena, para posteriormente, por medio de correlación, realizar un seguimiento del móvil en las imágenes subsecuentes.

Para tal fin, se implementó un algoritmo eficaz, que para la correlación tiene en cuenta al modelo no solamente en la orientación original sino también, en posiciones rotadas, con lo cual, el modelo solo se almacene en memoria una vez. El Algoritmo utiliza un valor de aceptación para tomar o rechazar un resultado el cual esta muy ligado a las condiciones de iluminación y ha sido fijado previamente de acuerdo a las condiciones del laboratorio.

El diagrama del algoritmo para el seguimiento por medio de correlación se describe a continuación y se visualiza en la figura 16.

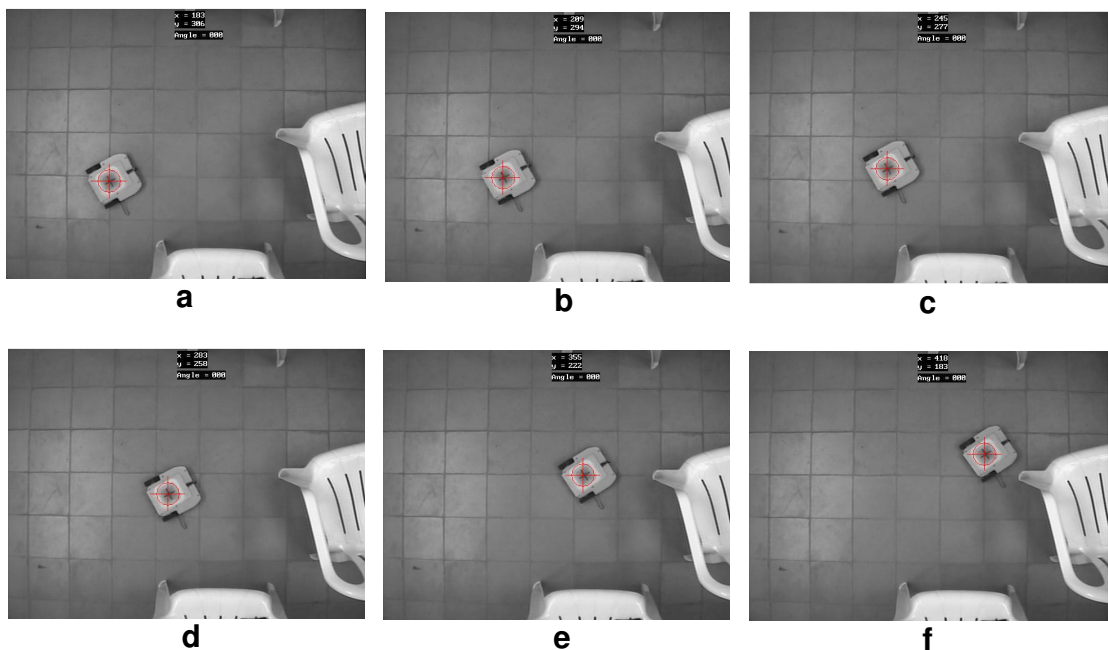
Figura 16. Diagrama del algoritmo de correlación



Inicialmente se lee una imagen en escala de grises que ha sido pre-procesada. Seguidamente se pregunta si el modelo ya está almacenado en memoria o no. Si la respuesta es afirmativa se ejecuta la función de correlación para buscar dicho modelo en la imagen adquirida, de lo contrario se visualiza la imagen con el

propósito que el usuario pueda seleccionar por medio del clic izquierdo del mouse el centro del modelo. Finalmente si se ha seleccionado satisfactoriamente un modelo, éste se guarda, se habilita la función de correlación angular, la cual tiene en cuenta los movimientos rotacionales del móvil y se fija un valor umbral con el cual se decide si un resultado es aceptable o no.

Figura 17. Resultados del algoritmo de correlación



En las figuras 17-a a 17-f, se observa la secuencia de seguimiento de un móvil por medio del algoritmo de correlación, la marca roja indica el lugar donde la correlación arrojó el mayor nivel de semejanza y esta posición coincide satisfactoriamente con el modelo elegido.

4. LOCALIZACIÓN E IDENTIFICACION

La información en píxeles proporcionada por las imágenes toma importancia en el momento en que la relacionamos con alguna magnitud física como el área, longitud etc, con la cual se pueda sacar conclusiones acerca de lo que ocurre en el mundo real que es captado por la cámara. La localización e identificación es la etapa que relaciona finalmente la información obtenida por el sensor que en este caso es la cámara, con la información física real en la escena.

4.1 LOCALIZACIÓN DE OBJETOS EN LA IMAGEN

Después de realizado el proceso de detección y segmentación de movimiento es necesario realizar la localización del centroide del objeto segmentado con respecto al origen de coordenadas de la imagen. Aunque el centroide, está relacionado con una propiedad física de los objetos como es su masa, tiene una aplicación particular en el procesamiento digital de imágenes. A continuación se describe de forma breve el concepto de momentos, necesario para realizar el cálculo del centroide aplicado al procesamiento digital de imágenes.

- **Momentos.** Son propiedades numéricas que dependen de las coordenadas y del valor que toman las variables en función de ellas. Para una función discreta⁵ el momento (p,q) se define de las siguiente manera :

$$M(p, q) = \sum_x \sum_y x^p y^q f(x, y) \quad (9)$$

⁵ En procesamiento digital de imágenes, corresponde a los niveles de intensidad en cada una de las posiciones dentro de la imagen.

- **Centroide.** se obtiene por medio de los momentos de orden 1 $M(1,0)$, $M(0,1)$ y de orden cero $M(0,0)$. Este último corresponde al área en píxeles del objeto segmentado. El cálculo realizado con estos momentos se muestra en la siguiente ecuación.

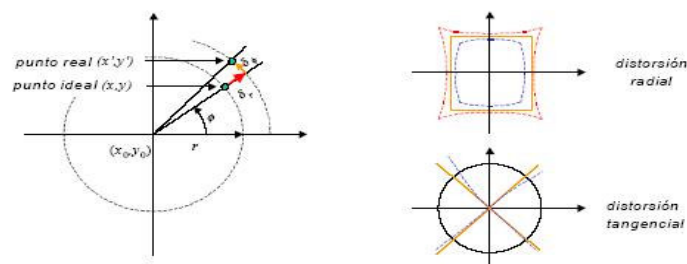
$$X = \frac{M(1,0)}{M(0,0)} \quad Y = \frac{M(0,1)}{M(0,0)} \quad (10)$$

4.2 CALIBRACIÓN DE LA CÁMARA

La calibración de la cámara que se considera en el presente trabajo, es el proceso de estimación y corrección de las distorsiones causadas por la cámara. Una calibración precisa es necesaria en muchas aplicaciones de visión computacional y más aún en la que se expone en el presente proyecto, ya que los resultados de posición entregados por la aplicación pueden ser erróneos debido a distorsiones presentes en las imágenes con las que se realiza el proceso de detección de movimiento.

Las distorsiones de la cámara pueden clasificarse como radial y tangencial. La distorsión radial hace que los puntos de la imagen se desplacen en forma radial a partir de su centro óptico, mientras que la tangencial ocurre a partir del eje óptico formando ángulos rectos con las líneas radiales. En la figura 18 se observa los dos tipos de distorsiones

Figura 18. Distorsiones en las lentes



Cuando se realiza la calibración de una cámara se obtienen dos clases de parámetros: los intrínsecos y los extrínsecos. Los parámetros intrínsecos corresponden a propiedades ópticas de la cámara, mientras que los parámetros extrínsecos dan cuenta de la posición de la cámara con respecto al sistema de coordenadas que la relaciona con el mundo real.

Para la calibración de la cámara DFK50H13/N utilizada en el presente proyecto se empleó la toolbox de Matlab CAMERA CALIBRATION TOOLBOX⁶, la cual haciendo uso de al menos 20 imágenes tomadas en diferentes posiciones de un patrón de calibración semejante a un tablero de ajedrez, entrega los parámetros intrínsecos de la cámara.

Los parámetros intrínsecos de la cámara DFK50H13/N calculados por la toolbox son los siguientes:

- Longitudes Focales
 $F_{cx} = 1077.18144$ pixel $F_{cy} = 1074.90424$ pixel
- Centro óptico
 $C_{ox} = 328.63616$ píxel $C_{oy} = 194.06424$ píxel
- Coeficientes de distorsión radial
 $K_{c1} = -0.31630$ $K_{c2} = -0.31778$ $K_{c6} = 0$
- Coeficientes de distorsión tangencial
 $K_{c3} = 0.00186$ $K_{c4} = 0.00094$

En los parámetros intrínsecos se observa que los coeficientes de distorsión radial son de mayor valor comparados con los de distorsión tangencial, lo cual hace de

⁶ Toolbox de libre descarga para aplicaciones no comerciales. Documentación y software disponible en (http://www.vision.caltech.edu/bouguetj/calib_doc/)

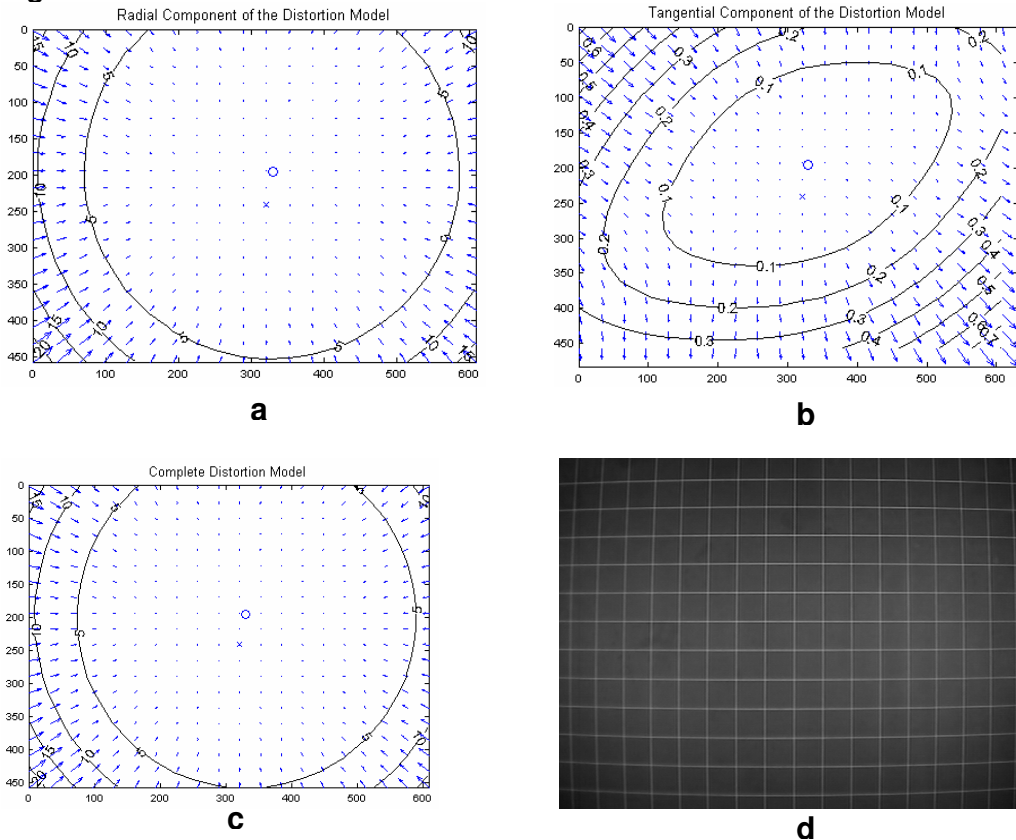
la distorsión radial, la más notoria en las imágenes capturadas con la cámara DFK50H13/N.

En la figura 19 se observa el modelo de distorsión generado por la toolbox de calibración para la cámara utilizada en el proyecto. En el incisos 19-a y 19- b se observan los modelos de distorsión radial y tangencial respectivamente. En el 19-c se observa el modelo de distorsión completa.

Cada circunferencia en esta grafica señala la incidencia de distorsión, la cual es función de la distancia al centro óptico. A medida que la distancia aumenta, la cantidad de distorsión observada aumenta, presentándose en el centro de la imagen un nivel de distorsión casi nulo.

Finalmente en el inciso 19-d se observa una imagen capturada con la cámara posicionada de acuerdo con lo expuesto en la descripción del escenario; en la cuadrícula se observa la incidencia de las distorsiones al momento de la captura.

Figura 19. Modelo de distorsión de la cámara



4.3 CORRECCIÓN DE DISTORSIONES

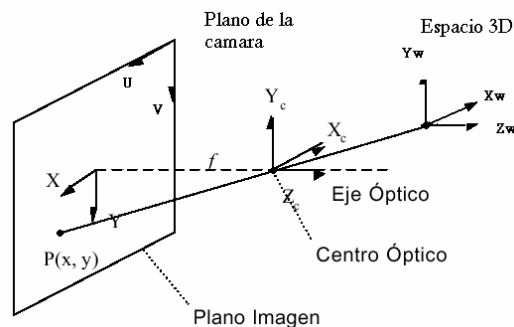
Cuando una imagen es capturada por una cámara, cada punto físico en el espacio real tridimensional se transforma en un píxel del espacio bidimensional de la imagen. Debido a las curvaturas del lente de la cámara, esta transformación tiende a distorsionarse, lo cual dificulta el cambio del espacio de la imagen al espacio real 3D. En el presente proyecto se implementó una estrategia con base en métodos numéricos para solucionar esta tarea; esto se expone a continuación.

En primera instancia es necesario realizar la transformación de las coordenadas de la imagen a las coordenadas del espacio de la cámara, para lo cual se utilizan las ecuaciones del modelo pin-hole de la cámara [12]. La descripción matemática del modelo se muestra en la ecuación 11 y un esquema general del mismo, en la figura 20.

$$\begin{aligned} P_U &= P_X f_{CX} + C_{OX} \\ P_V &= P_Y f_{CY} + C_{OY} \end{aligned} \quad (11)$$

Donde P_U y P_V son las coordenadas de un punto con respecto al origen del plano de la imagen y P_X y P_Y son las coordenadas del mismo punto con respecto al origen de coordenadas del plano de la cámara. El resto de parámetros corresponden a la distancia focal y a las coordenadas del centro óptico de la cámara obtenidos en la calibración.

Figura 20. Modelo Pin-Hole de la cámara



Después de realizar la transformación de las coordenadas del plano imagen al plano de la cámara se utiliza la ecuación 12, que permite corregir la distorsión causada por el lente de la cámara con origen en el plano de la cámara.

$$\begin{aligned}
 P_{dx} &= P_{ix} \left(1 + K_{c1}r^2 + K_{c2}r^4 + K_{c5}r^6\right) + 2K_{c3}P_{ix}P_{iy} + 2K_{c4}P_{ix}^2 + K_{c4}r^2 \\
 P_{dy} &= P_{iy} \left(1 + K_{c1}r^2 + K_{c2}r^4 + K_{c5}r^6\right) + 2K_{c4}P_{ix}P_{iy} + 2K_{c3}P_{iy}^2 + K_{c3}r^2 \quad (12) \\
 r &= P_{ix}^2 + P_{iy}^2
 \end{aligned}$$

En donde P_{dx} y P_{dy} corresponden a las coordenadas con distorsión, P_{ix} y P_{iy} corresponden a las coordenadas sin distorsión y el resto de parámetros corresponde a los coeficientes de distorsión calculados en la calibración de la cámara.

Con estas dos ecuaciones se busca encontrar el punto P_{ix} y P_{iy} , dadas las coordenadas P_{dx} y P_{dy} . Para resolver las ecuaciones se utilizó el método de Newton-Raphson para ecuaciones no lineales [2].

Después de tener sin distorsión las coordenadas del punto, se lleva a este nuevamente al origen de coordenadas de la imagen por medio del modelo pin-hole de la cámara.

Es importante aclarar que la corrección de las distorsiones no se realiza sobre todos los puntos de la imagen adquirida, sino solamente sobre los puntos de interés hallados, es decir sobre el centroide de los objetos en movimiento y estáticos. Esto se hace con el fin de optimizar recursos computacionales y disminuir el tiempo total de procesamiento.

4.4 TRANSFORMACIÓN DE PÍXELES A CENTÍMETROS

Una vez realizada la corrección de las distorsiones el siguiente paso es transformar los píxeles a medidas de longitud con el fin de dar la ubicación de los objetos con respecto a un origen de coordenadas en el mundo real, en unidades de longitud. Para este fin se uso un factor de escala encontrado bajo las condiciones del escenario descrito en el capítulo 2. Este factor de escala es:

$$fe = \frac{10 \text{ Centimetros}}{39 \text{ pixel}}$$

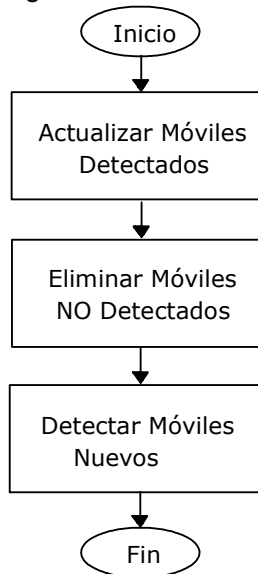
Para encontrar este factor se ubico un cuadrado de 10 x 10 cm en la pista del escenario; luego se realizo la captura de la imagen, para posteriormente medir en píxeles los lados del cuadrado en la imagen. El factor de escala encontrado es equivalente tanto en el ancho como en el alto del cuadrado, por lo tanto se usa el mismo valor de escala para la transformación.

4.5 ALGORITMO DE IDENTIFICACIÓN

En la escena se pueden tener múltiples móviles (máximo seis), por tanto se requiere de un algoritmo que esté constantemente identificando a cada uno de los móviles con el fin de realizar un correcto seguimiento de ellos y no llegar a confundirlos uno con otro.

La etapa de identificación consta de tres fases fundamentales: actualizar móviles detectados, eliminar móviles no detectados y detectar móviles nuevos. En la figura 21 se muestra el diagrama de flujo para este algoritmo.

Figura 21. Diagrama de flujo para algoritmo de identificación



A continuación se describe cada una de las fases del algoritmo de identificación.

Actualización de móviles detectados Consiste en actualizar la información de posición y orientación de cada uno de los móviles que habían sido detectados en la imagen anterior y que se han detectado en la imagen actual.

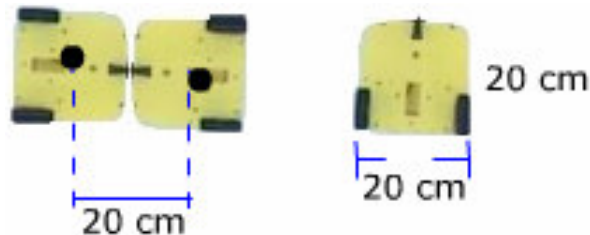
Eliminación de móviles NO detectados: consiste en eliminar la información de posición y orientación de un móvil que no ha sido detectado en la imagen actual pero que se había detectado en la imagen anterior.

Detección de móviles nuevos. En esta fase se agrega la información de posición y orientación de un móvil que se ha detectado en la imagen actual pero que no se había detectado en la imagen anterior.

La manera de conocer si un móvil que se había detectado en la imagen anterior aparece nuevamente en la imagen actual o ha desaparecido en esta última, se explica a continuación:

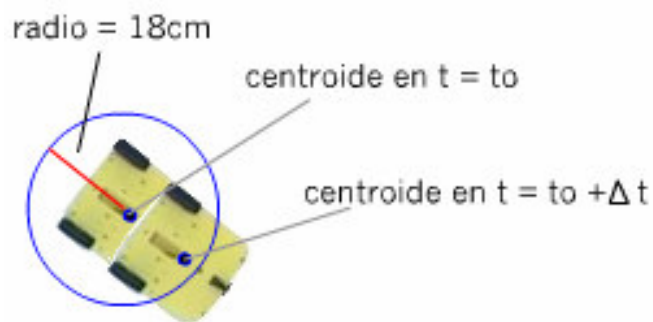
Se parte del hecho que las dimensiones de los móviles utilizados en el presente proyecto son de 20cm x 20cm y que cuando dos móviles están lo mas cercano posible sin haber chocado, sus centros de gravedad se encuentran a una distancia de 20cm aproximadamente (figura 22).

Figura 22. Dimensiones de los móviles



Consecuentemente, para que pueda realizarse un correcto seguimiento de cada uno de los móviles en la escena, es necesario que el centroide de un objeto se encuentre dentro de una circunferencia de radio 18 cm con origen en el centroide del mismo objeto en la imagen anterior. (figura 23).

Figura 23. Identificación de un móvil



Al fijarse el radio de la circunferencia en 18 cm, se asegura que nunca se va a confundir un móvil que se desplaza, con otro objeto cercano. Como contraparte, esta elección limita la velocidad máxima a la cual puede desplazarse un móvil para

poder ser identificado y realizar un correcto seguimiento del mismo. Esta velocidad es:

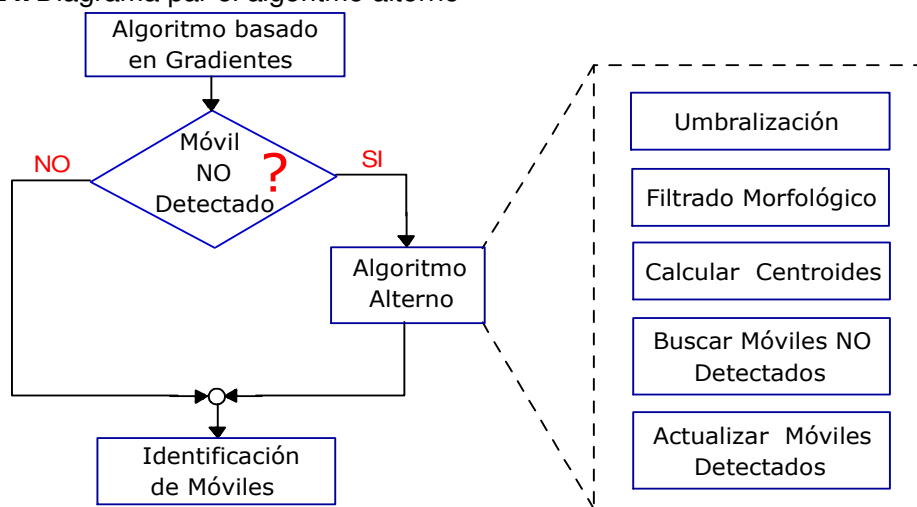
$$\begin{aligned} v_{\max} &= \frac{x_{\max}}{\Delta t} \\ v_{\max} &= \frac{18 \left[\frac{cm}{s} \right]}{\Delta t} \end{aligned} \quad (13)$$

donde v_{\max} y Δt corresponden respectivamente a la velocidad máxima de desplazamiento del móvil y al tiempo que requiere la aplicación para adquirir una imagen y procesarla. El tiempo de procesamiento depende del algoritmo en ejecución.

El algoritmo descrito anteriormente, funciona correctamente siempre y cuando los móviles puedan ser localizados en todo instante, sin importar si están en movimiento o no. Cuando se ejecuta el algoritmo basado en correlación o el algoritmo basado en color, siempre se logran localizar los objetos estáticos o en movimiento, sin embargo esto no ocurre cuando se utiliza el algoritmo basado en gradientes, el cual localiza solo los objetos en movimiento.

Para subsanar este inconveniente, se optó por implementar un algoritmo alternativo, con el cual se pueda establecer la localización de un objeto móvil cuando se quede estático o salga de la escena. El diagrama para el algoritmo alternativo se muestra en la figura 24.

Figura 24. Diagrama par el algoritmo alterno



El algoritmo alterno mostrado en la figura 24, solo se ejecuta cuando el número de móviles detectados en la imagen actual es inferior al detectado en la imagen anterior.

Este algoritmo inicialmente consiste en hacer una umbralización seguida de un filtrado morfológico para segmentar los móviles en la escena. Luego se realiza el calculo del centroide de cada objeto segmentado. Finalmente se efectúa la búsqueda de los móviles no detectados por medio de comparación con los móviles detectados en la imagen anterior y se actualiza la información de estos móviles. La actualización consiste en establecer la posición de los móviles que ahora están estáticos o eliminar los móviles ya no están en la escena.

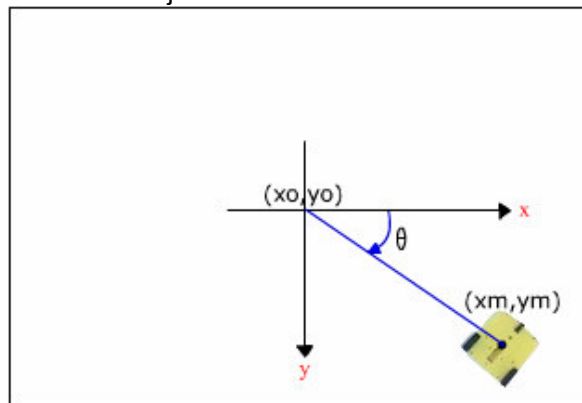
4.6 CÁLCULO DE ORIENTACIÓN DEL MÓVIL

En la aplicación implementada, la orientación se puede calcular respecto a dos sistemas de referencia. El primero es un sistema de referencia fijo el cual se puede establecer en cualquier lugar de la escena y el segundo es de referencia variable en cada instante.

4.6.1 Sistema de referencia fijo

En este sistema es necesario fijar el origen de coordenadas en un punto cualquiera de la escena. En la figura 25, se puede observar el sistema de referencia fijado en el centro de la escena.

Figura 25. Sistema de referencia fijado en el centro de la escena



En el sistema de referencia de la figura 25, el ángulo para un móvil aumenta en el sentido de las manecillas del reloj respecto al eje positivo x , desde de 0^0 hasta 360^0 . Este ángulo se puede calcular por medio de la relación mostrada en la ecuación 14.

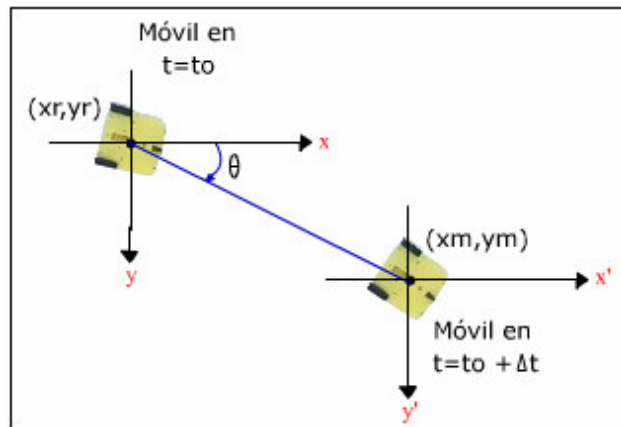
$$\theta = \tan^{-1}\left(\frac{ym - yo}{xm - xo}\right) \text{ [radianes]} \quad (14)$$
$$\theta = \frac{180}{\pi} \theta \text{ [grados]}$$

Donde el punto (xo, yo) es el origen de coordenadas fijo y el punto (xm, ym) hace referencia al centroide del móvil en cada instante.

4.6.2 Sistema de referencia variable

La referencia se fija en el centroide del objeto y cambia en cada instante conforme este se mueve. Este sistema se muestra en la figura 26.

Figura 26. Sistema de referencia variable



En la figura 27 se puede apreciar un móvil ubicado en instantes diferentes de tiempo. Cuando $t = t_0$ el móvil se encuentra en el punto (x_r, y_r) , un instante después, en $t = t_0 + \Delta t$, el móvil está ubicado en el punto (x_m, y_m) . La orientación del movimiento del objeto se puede calcular por medio de la ecuación 15.

$$\theta = \tan^{-1}\left(\frac{y_m - y_r}{x_m - x_r}\right) [\text{radianes}] \quad (15)$$
$$\theta = \frac{180}{\pi} \theta [\text{grados}]$$

5. PRUEBAS Y ANÁLISIS DE RESULTADOS

En este capítulo se presentan las pruebas realizadas a los algoritmos implementados. Estas pruebas son:

- Posición.
- Orientación.
- Seguimiento de trayectorias.
- Tiempo de latencia

El análisis de cada uno de los resultados obtenidos en las pruebas mencionadas permitirá medir el desempeño de la aplicación en la localización tanto de objetos móviles como estáticos.

Para realizar estas pruebas se utilizó una aplicación implementada en visual C++ 6.0 (FENIX) (ver anexo C), la cual genera un archivo de texto con la información de posición y orientación de los objetos de interés. Además para emular el objeto en movimiento o estático se usó en todas las pruebas un rectángulo de cartulina cuyas dimensiones son 17cm x 12cm con el fin de localizar fácilmente su centroide en la intersección de sus diagonales.

El error, el valor medio y la desviación estándar para las pruebas se calcula por medio de las siguientes relaciones:

$$\text{error} = \text{valor medido} - \text{valor real}$$

$$\text{valor medio} = \frac{1}{n} \sum_{i=0}^n x_i$$

$$\text{desviación estándar} = \sqrt{\frac{1}{n-1} \sum_{i=0}^n (x_i - \text{valor medio})^2}$$

donde **n** es el número de muestras.

El cálculo del error permite medir el nivel de exactitud, mientras la desviación estándar establece el nivel de precisión de las mediciones.

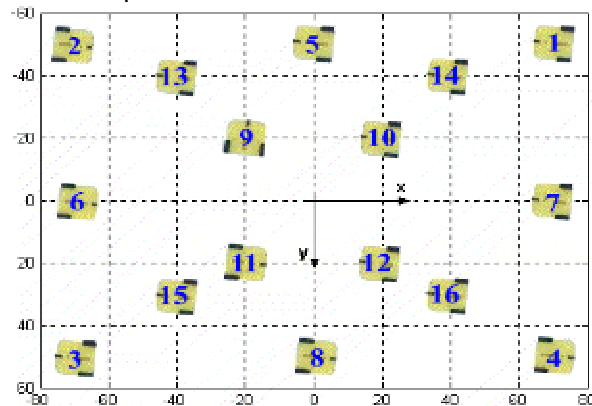
5.1 POSICIÓN

El objetivo principal de esta prueba es medir el nivel de exactitud y precisión que tiene la aplicación desarrollada cuando se usa para localizar (posición) los objetos de interés en la escena.

Para realizar esta prueba, se dividió la pista en una cuadrícula de 10cm x 10cm simulando en el mundo real un sistema de coordenadas cartesiano con origen en el centro de la pista. A continuación se ubicó en la pista un objeto en diferentes posiciones, de tal manera que su centroide coincidiera con una posición (x,y) conocida.

Por último se localizó el objeto con la aplicación desarrollada, tomando para cada posición 50 datos. Las posiciones elegidas están distribuidas de tal manera que cubran la mayor parte del área de trabajo como se observa en la figura 27.

Figura 27. Posiciones usadas para determinar la distribución del error.



Para esta prueba, el algoritmo basado en gradientes no arroja resultado alguno, ya que el objeto que se va a localizar no se encuentra en movimiento, por lo cual este algoritmo se excluye de la prueba.

Los resultados numéricos de la prueba para cada algoritmo se muestran en una tabla, en la cual se puede observar el valor medio, la desviación estándar y el error para el conjunto de datos de cada una de las posiciones elegidas. Un ejemplo de esto se observa en la tabla 1, en la cual se muestran los resultados para el algoritmo basado en el modelo de color HLS.

Tabla 1. Resultados para el algoritmo basado en el modelo de color HLS.

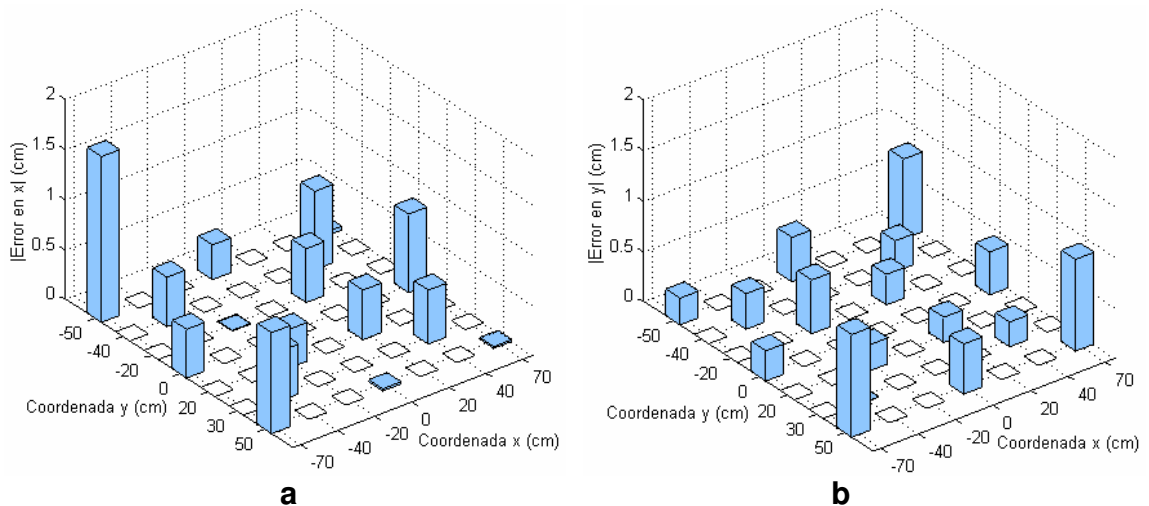
Nº	Posición Real (cm)	Medidas en x (cm)			Medidas en y (cm)		
		Valor Medio (cm)	Desviación estándar	Error (cm)	Valor Medio (cm)	Desviación estándar	Error (cm)
1	(70,-50)	70.0381	0	0.0381	-50.7900	0	-0.7900
2	(-70,-50)	-68.3434	0	1.6566	-49.7353	0	0.2647
3	(-70,50)	-68.9902	0	1.0098	48.9834	0	-1.0166
4	(70,50)	70.0168	0	0.0168	49.0923	0	-0.9077
5	(0, -50)	0.3469	0.0962	0.3469	-50.4403	0.0003	-0.4403
6	(-70,0)	-69.5129	0.0001	0.4871	-0.2994	0.0711	-0.2994
7	(70,0)	70.7840	0.0878	0.7840	-0.4263	0.0006	-0.4263
8	(0,50)	-0.0218	0.0512	-0.0218	50.4955	0	0.4955
9	(-20,-20)	-19.9929	0	0.0071	-20.5314	0	-0.5314

10	(20,-20)	20.5385	0	0.5385	-20.3028	0	-0.3028
11	(-20,20)	-20.3871	0.1298	-0.3871	20.2580	0	0.2580
12	(20,20)	20.4957	0	0.4957	20.2488	0	0.2488
13	(-40,-40)	-39.5058	0	0.4942	-40.3517	0	-0.3517
14	(40,-40)	40.7835	0.0004	0.7835	-40.3073	0.0520	-0.3073
15	(-40,30)	-40.4978	0	-0.4978	30.0018	0	0.0018
16	(40,30)	40.5379	0	0.5379	30.2446	0	0.2446

Para ver las tablas de resultados de cada uno de los algoritmos en forma detallada ver el anexo A.

En la figura 29 se puede apreciar la distribución del error de posición, por medio un grafico de barras 3D, en cada una de las posiciones elegidas para esta prueba. En la figura 29-a y 29-b se muestra el error en las coordenadas x e y respectivamente.

Figura 28. Distribución 3D del error para el algoritmo basado en el modelo de color HLS
a) Error en la coordenada x b) Error en la coordenada y



Los resultados anteriores, muestran que para la mayoría de las posiciones, el error de posición que resulta de utilizar el algoritmo basado en el modelo de color HLS, tiene un valor cercano e inferior a 1cm.

5.2 ANÁLISIS COMPARATIVO DE RESULTADOS PARA LA PRUEBA DE POSICIÓN

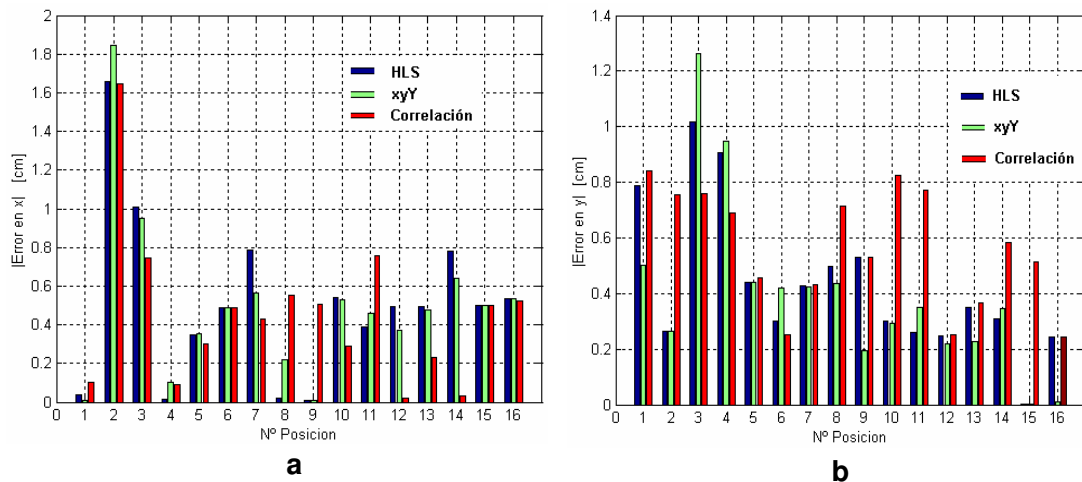
Como se aprecia en la tabla 2, para los dos algoritmos implementados, el error no supera 1 cm en la mayoría de las posiciones elegidas, lo cual es un valor aceptable, dadas las dimensiones del área de trabajo (122cm x 165cm).

Tabla 2. Resultados comparativos para el error de posición.

Nº	Localización Real (cm)	Error en x (cm)			Error en y (cm)		
		HLS	xyY	Correlación	HLS	xyY	Correlación
1	(70,-50)	0.0381	0.0080	0.0995	0.7900	0.5023	0.8399
2	(-70,-50)	1.6566	1.8423	1.6433	0.2647	0.2656	0.7537
3	(-70,50)	1.0098	0.9482	0.7473	1.0166	1.2628	0.7595
4	(70,50)	0.0168	0.0999	0.0907	0.9077	0.9497	0.6886
5	(0, -50)	0.3469	0.3525	0.3023	0.4403	0.4403	0.4581
6	(-70,0)	0.4871	0.4870	0.4881	0.2994	0.4189	0.2498
7	(70,0)	0.7840	0.5631	0.4308	0.4263	0.4249	0.4310
8	(0,50)	0.0218	0.2211	0.5530	0.4955	0.4373	0.7150
9	(-20,-20)	0.0071	0.0072	0.5069	0.5314	0.1924	0.5304
10	(20,-20)	0.5385	0.5282	0.2880	0.3028	0.2925	0.8248
11	(-20,20)	0.3871	0.4589	0.7571	0.2580	0.3503	0.7719
12	(20,20)	0.4957	0.3726	0.0178	0.2488	0.2181	0.2514
13	(-40,-40)	0.4942	0.4738	0.2336	0.3517	0.2280	0.3664
14	(40,-40)	0.7835	0.6383	0.0306	0.3073	0.3475	0.5849
15	(-40,30)	0.4978	0.4978	0.5004	0.0018	0.0018	0.5156
16	(40,30)	0.5379	0.5379	0.5248	0.2446	0.0119	0.2439

En la figura 29-a y 29-b se aprecia la distribución del error de posición en las coordenadas x e y, para los tres algoritmos implementados. Las barras azules señalan el error correspondiente al algoritmo basado en color con el modelo HLS, las verdes con el modelo de color xyY y las rojas al algoritmo de correlación.

Figura 29. Distribución comparativa del error de posición utilizando los algoritmos implementados a) Error en las coordenadas en x . b) Error en las coordenadas en y



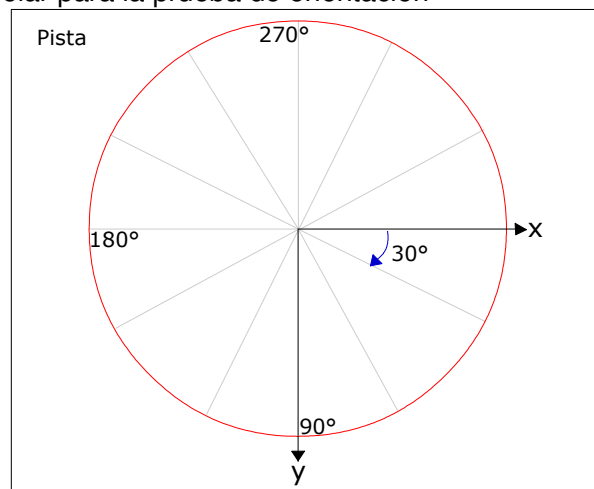
En la distribución del error de posición se observa que este es mayor que un centímetro para la coordenada x en la posición 2 y para la coordenada y en las posiciones 3 y 4. Estos errores se deben a que tales puntos se encuentran localizados en los extremos de la imagen, lugar en el cual se presenta el mayor nivel de distorsión, como se expuso en el capítulo 4.

Por último se considera que la aplicación desarrollada tiene una alta precisión en localización ya que para los algoritmos implementados la desviación estándar calculada es inferior a 0.1459, siendo el más preciso de todos el algoritmo de color con el modelo HLS el cual en todas las posiciones seleccionadas tiene desviaciones estándar inferiores a 0.0711 (ver anexo A)

5.3 ORIENTACIÓN

Para realizar esta prueba, se simuló un sistema de coordenadas polar fijado en el centro de la pista y se realizaron divisiones de 30° en un rango de 0° a 360°. (ver figura 30).

Figura 30. Sistema polar para la prueba de orientación



Luego se ubicó un objeto en diferentes posiciones, de tal manera que el ángulo formado por éste respecto al eje x del sistema de coordenadas, varíe desde 0° hasta 330° con espaciamiento de 30°. Para cada posición se tomaron 50 datos del ángulo los cuales se utilizaron para su correspondiente análisis.

Los resultados de esta prueba, se encuentran detallados en el anexo A, en el cual se puede encontrar una tabla con el valor medio, desviación estándar y el error para cada una de las diferentes orientaciones calculadas, con los diferentes algoritmos.

La tabla 3 muestra el error de orientación inherente a cada algoritmo.

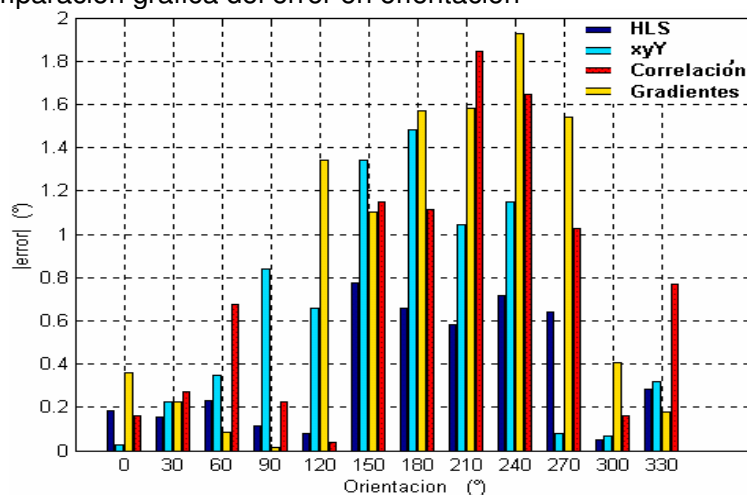
Tabla 3. Error de orientación para cada uno de los algoritmos implementados

Orientación Real(°)	Error (°)			
	HLS	xyY	Correlación	Gradientes
0	0.1822	-0.0282	0.3590	0.1585
30	-0.1525	0.2279	-0.2271	-0.2721
60	0.2316	0.3499	-0.0833	-0.6759
90	0.1166	0.8363	0.0118	0.2248

120	-0.0812	0.6562	1.3432	-0.0363
150	0.7737	1.3432	1.1034	1.1494
180	0.6569	1.4840	1.5681	1.1140
210	0.5831	1.0430	1.5811	1.8446
240	0.7156	1.1486	1.9291	1.6463
270	-0.6403	0.0796	1.5414	1.0272
300	-0.0508	-0.0696	0.4069	0.1634
330	-0.2823	-0.3178	-0.1765	0.7685

En la figura 31 se muestra el error en orientación correspondiente al algoritmo basado en el modelo de color HLS con barras verticales de color azul oscuro, con barras de color azul claro para el algoritmo basado en el modelo de color xyY, con barras amarillas para el algoritmo basado en correlación y el algoritmo basado en gradientes se identifica con barras rojas.

Figura 31. Comparación gráfica del error en orientación



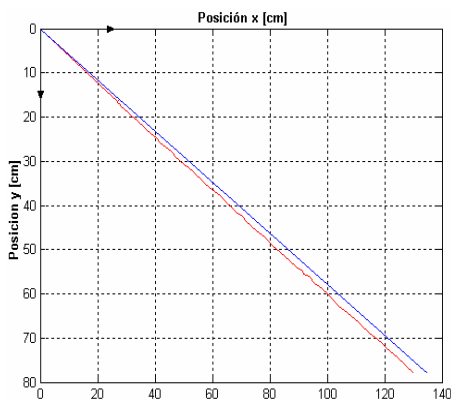
En la anterior figura se puede comparar gráficamente el error en orientación para cada uno de los algoritmos. Se observa que para todos los algoritmos el error es inferior a 2°. Su valor es menor en el rango entre 0° y 90° mientras que entre 150° y 270° se presenta el mayor valor. El algoritmo basado en el modelo de color HLS tiene la mayor exactitud, con un error inferior a 0.8° en todo el rango.

5.4 SEGUIMIENTO

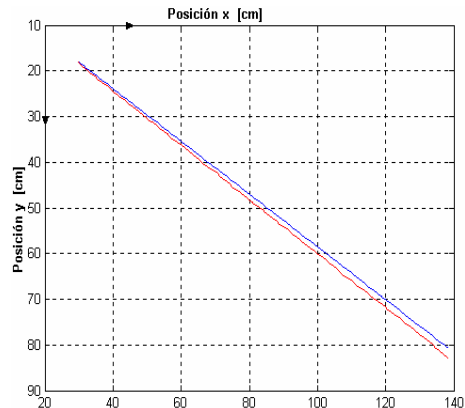
Esta prueba se realizó con el fin de comprobar el nivel de fidelidad de la aplicación desarrollada al realizar el seguimiento de un móvil que describe una trayectoria conocida.

Para la ejecución de la prueba se optó por usar dos trayectorias: la primera es una línea recta con una pendiente de 30° con respecto al eje x. En la figura 32 en color azul se observa la trayectoria ideal y en color rojo la trayectoria calculada con cada uno de los algoritmos implementados

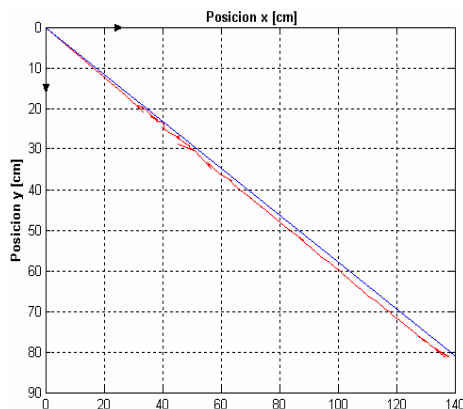
Figura 32. Seguimiento de una trayectoria recta
a) con el modelo de color HLS b) con el modelo de color xyY c) con el algoritmo basado en gradientes d) con el algoritmo basado en correlación



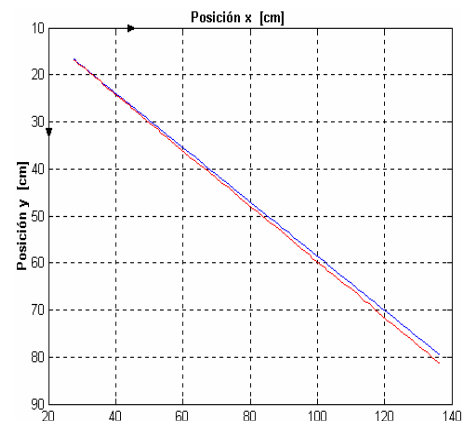
a



b



c

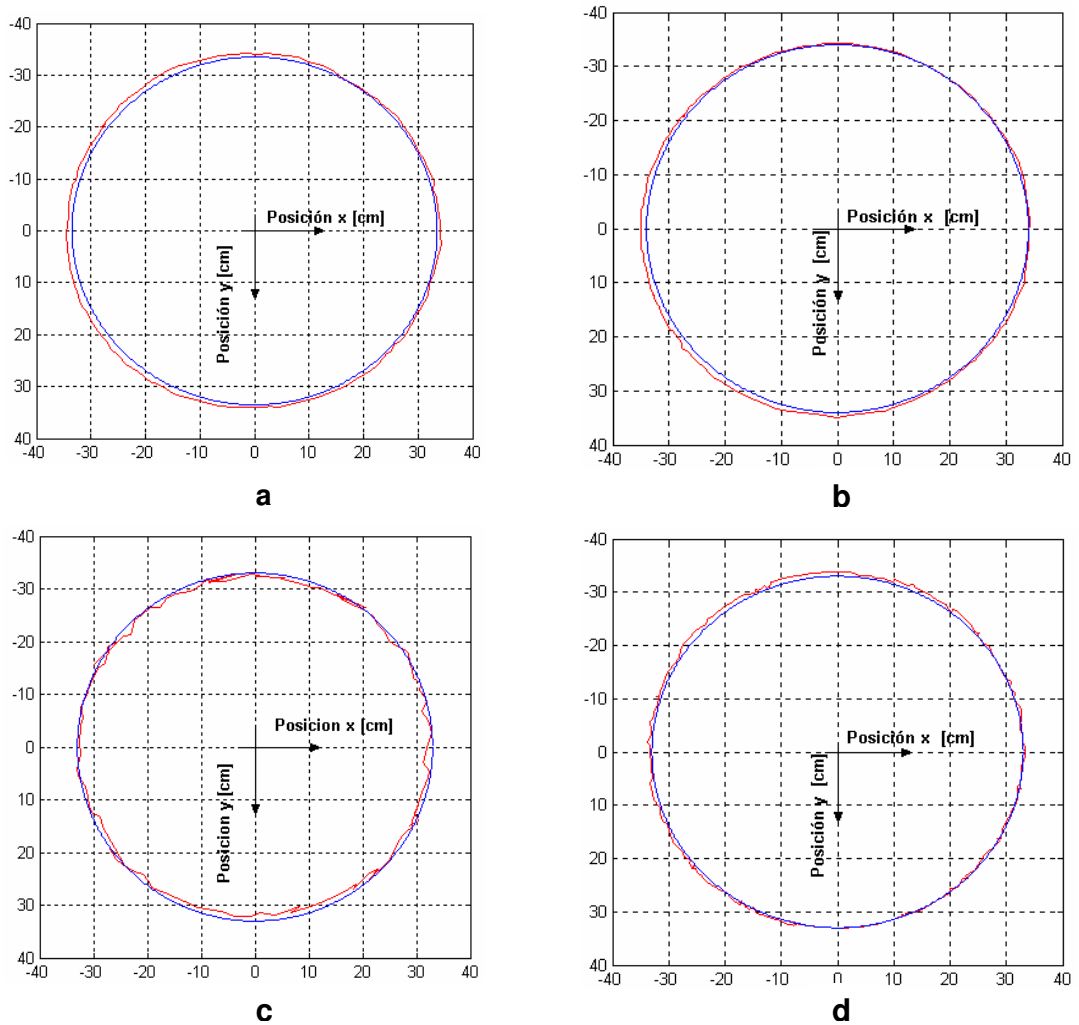


d

Aunque el seguimiento de una trayectoria rectilínea es altamente fiel para los cuatro algoritmos implementados, se observo un leve cambio en la pendiente de la recta. Esto es ocasionado por errores humanos durante el proceso de desplazamiento del objeto y medición de la pendiente de la trayectoria prevista para la prueba.

La segunda trayectoria descrita es circular. Los resultados para el seguimiento con los algoritmos implementados se observan en la figura 33.

Figura 33. Seguimiento de una trayectoria circular
a) con el modelo de color HLS b) con el modelo de color xyY c) con el algoritmo basado en gradientes d) con correlación



La curva en color azul representa la trayectoria ideal y en color rojo, la trayectoria calculada por la aplicación. En este caso el seguimiento tiene una alta precisión salvo el realizado con el algoritmo de gradientes donde se presentan variaciones producidas por distorsiones en la segmentación y el filtrado morfológico.

5.5 TIEMPO DE LATENCIA

Esta prueba tiene como objetivo determinar la velocidad de respuesta de cada uno de los algoritmos implementados.

Esta prueba se realizó colocando sobre la pista seis objetos móviles y seis obstáculos. Se escogió este número de objetos debido a que es el máximo que la aplicación puede identificar simultáneamente. Al trabajar con este número de objetos (móviles y obstáculos) se obtiene el tiempo crítico para cada algoritmo ya que al incrementar el número de objetos de interés en la escena también incrementa el tiempo de procesamiento.

En la tabla 4 se muestran los tiempos de adquisición, procesamiento y el tiempo total de latencia el cual es la suma de los dos anteriores. Cuando los objetos móviles se detectan por correlación o gradientes, los obstáculos pueden ser detectados con base en el modelo de color xyY o el HLS. Cuando los objetos móviles son detectados por el algoritmo basado en un modelo de color, xyY o HLS, los obstáculos son detectados con base en el mismo modelo de color.

Tabla 4. Tiempo de latencia

Método Tiempo de latencia (seg)	Correlación		xyY	HLS	Gradientes		Móviles
	xyY	HLS			xyY	HLS	Obstáculos
Adquisición	0,0735	0,0735	0,0735	0,0735	0,0735	0,0735	
Procesamiento	0.8108	0.3018	0.9926	0.1661	0.7742	0.3000	
Total	0,8821	0,3748	1,0643	0,2429	0,8515	0,3712	

En la tabla anterior se observa que el tiempo total de latencia varía desde 0.2429s hasta 1.0643s. El algoritmo basado en el modelo de color HLS es el más rápido con un tiempo de latencia de 0.2429s.

El algoritmo basado en el modelo de color xyY es el más lento, con un tiempo de latencia de 1.0643s. Esto se debe al proceso de normalización de las componentes X e Y de los valores triestímulo, el cual consume un tiempo considerable.

El tiempo de latencia limita la velocidad máxima con la cual pueden moverse los objetos para poder ser localizados e identificados correctamente con la aplicación desarrollada. Esto se muestra a continuación:

Tabla 5. Velocidad máxima de un objeto para ser identificado

	Correlación		xyY	HLS	Gradientes		Móviles
	xyY	HLS			xyY	HLS	Obstáculos
Velocidad máxima (cm/s)	20,405	48.025	16.912	74.104	21.139	48.491	

Esta prueba se realizó en un PC con las siguientes características:

Sistema Operativo: Microsoft Windows XP Profesional versión 2002.

Procesador: Pentium 4 de 1.50 GHz.

Memoria RAM: 256 MB.

Cabe resaltar que el tiempo consumido en procesamiento y por ende la máxima velocidad a la cual puede desplazarse un objeto móvil, dependen de factores

como la frecuencia del reloj del procesador, la velocidad de los buses de la board del PC, la velocidad de la memoria RAM, etc.

6. CONCLUSIONES Y OBSERVACIONES

El resultado final del proyecto es una aplicación en Microsoft Visual C++ 6.0 con la cual se puede realizar la localización de los objetos de interés en una escena de condiciones controladas. Como resultado se obtiene la posición y la orientación para objetos móviles y la posición para obstáculos. Para el efecto se utilizan tres algoritmos basados respectivamente en gradientes espacial y temporal, correlación y color. Además se realizó un algoritmo de adquisición que permite capturar de forma eficiente las imágenes necesarias, utilizando una tarjeta digitalizadora DFG/Compress PIRANHA.

Se realizó la configuración de la tarjeta DFG/COMPRESS PIRANHA para adquirir video e imágenes independientes. De esta forma la etapa de adquisición puede interactuar automáticamente con la etapa de procesamiento, de manera que el análisis de movimiento en las imágenes se realiza una vez cada una de ellas es capturada. La etapa de adquisición puede ser utilizada en diferentes aplicaciones, debido al tipo de programación orientado a objetos que se utilizó al implementar FÉNIX, el cual permite modificar y ampliar la aplicación sin alterar las demás funciones. De esta manera, se puede crear nuevas funciones para otras tareas diferentes.

El algoritmo basado en gradientes implementado en este proyecto es una nueva estrategia que se plantea para detectar los objetos en movimiento y avanzar en una futuras investigaciones sobre el flujo óptico[12] con el cual se pueden realizar mediciones mas precisas sobre el movimiento tales como velocidad y aceleración . En comparación con otros métodos para la detección del movimiento como la extracción del fondo y la diferencia de dos imágenes, el algoritmo basado en gradientes combina información temporal y espacial lo cual solucionó los

inconvenientes de ruido en la detección de los objetos en movimiento cuando se encuentran en la escena elementos que no son de interés en la detección.

La etapa de segmentación basada en los modelos de color HLS y xyY, es bastante confiable y robusta gracias al ambiente parcialmente controlado, donde el fondo de la escena tiene un color diferente de los utilizados en el reconocimiento de los objetos móviles y los obstáculos. Además, el color de los obstáculos es diferente al color de los objetos móviles, por lo tanto la identificación entre estos dos tipos de objetos (móviles y obstáculos) se hace fácil. Estos modelos de color tienen la ventaja de que la información cromática se determina en dos componentes, por lo cual el proceso de segmentación con estos modelos se realiza en un menor tiempo respecto a utilizar el modelo RGB, en el cual la información del color se encuentra en sus tres componentes.

Debido a que se trabajó con imágenes de dimensiones 640x480 píxeles sin algún tipo de compresión, la mayoría de las operaciones entre imágenes fueron bastante básicas como operaciones lógicas, sumas, restas y umbralizaciones, con el fin de no consumir demasiado tiempo en procesamiento. Esto se evidenció en la prueba del tiempo de latencia, donde para los tres algoritmos implementados el tiempo de procesamiento es inferior a 1 segundo.

Para el algoritmo basado en correlación, se logró cumplir las restricciones de bajo tiempo en procesamiento y efectividad en la localización, ajustando el nivel de aceptación (valor con el cual se acepta o rechaza un resultado), el tamaño del modelo patrón y el área de búsqueda. Hay que aclarar que a pesar que la función de correlación en imágenes necesita un gran número de multiplicaciones, se logró ganar velocidad a cambio de exactitud en el resultado, aunque manteniendo este último en un nivel bastante aceptable.

Las pruebas realizadas a la aplicación desarrollada arrojaron resultados con un error de posición y de orientación inferiores a 1 cm y 2º respectivamente. Lo

anterior evidencia la confiabilidad en la información de localización obtenida con los tres algoritmos y la utilidad del proceso de calibración para corregir las distorsiones de la cámara utilizada en el proyecto.

Se pueden identificar hasta seis objetos móviles diferentes sin necesidad de marcas específicas, lo cual hace la aplicación atractiva para aplicaciones de robots cooperantes.

La interfaz grafica que acompaña a la aplicación desarrollada es amigable al usuario ya que el entorno es semejante al utilizado en las diferentes herramientas Windows habitualmente usadas. Esto brinda al usuario la posibilidad de manejar las diferentes opciones de la aplicación de una forma sencilla.

La utilidad de los procesos morfológicos en problemas relacionados con tratamiento de imágenes es innegable; sin embargo es importante recalcar que su uso debe ser controlado y orientado a las necesidades del problema a solucionar, considerando que en casos donde se desee conservar los detalles del objeto, la aplicación de un proceso morfológico muy fuerte puede resultar perjudicial y conllevar a un deterioro de la imagen.

La aplicación desarrollada es útil en trabajos relacionados con el tema de planeación de trayectorias de robots móviles en un escenario como el planteado en el presente proyecto, ya que brinda la información confiable de localización de los robots y de los obstáculos presentes en el área de trabajo.

7. RECOMENDACIONES PARA TRABAJOS FUTUROS

Si se desea capturar una mayor área de trabajo, se puede ubicar la cámara a mayor altura o utilizar un lente con distancia focal inferior a 8mm. Si el lente utilizado es diferente al utilizado en el presente proyecto, se debe realizar un nuevo proceso de calibración de parámetros de la cámara y el lente.

Se recomienda explorar el uso de las funciones de compresión de imágenes de la tarjeta DFG/Compress PIRANHA [15], con el fin de lograr un avance en cuanto a la velocidad de transferencia de datos.

Construir un sistema de iluminación artificial apropiado ayudaría en el proceso de adquisición de las imágenes, segmentación y finalmente en la localización de los objetos de interés en la escena.

Un mayor grado de precisión y exactitud de los resultados podría obtenerse si se investiga a profundidad en los temas relacionados con calibración de cámaras y corrección de las distorsiones producidas por los lentes.

8. REFERENCIAS BIBLIOGRAFICAS

- [1]. BALLARD, Danna y BROWN, C. Computer Vision [on line]. New Jersey: Prentice Hall, 1982 [citado en mayo de 2004]. disponible en <<http://www.cs.rochester.edu/u/dana/>>
- [2]. CHAPRA, Steven y CANALE, Raymond. Métodos Numéricos para ingenieros. México: McGraw-Hill interamericana editores S.A .1999
- [3]. DE JALON, J. D et al. Aprenda C++ Como Si Estuviera En Primero [on line]. España: Universidad Superior de Ingenieros Industriales de San Sebastián y Universidad de Navarra, 1998[citado en mayo de 2004] disponible en <<http://www.abcdatos.com/tutoriales/tutorial/l2525.html>>
- [4]. GONZALEZ, Rafael y WOODS, Richard. Tratamiento digital de imágenes. Estados Unidos de América: Editorial Addison-Wesley iberoamericana.1996
- [5]. GUREWICH, Nathan y GUREWICH, Ori. Aprendiendo Visual C++ 5 En 21 Días. México: Prentice-Hall Hispanoamericana, S.A. Cuarta edición, 1998.
- [6]. JAIN, Ramesh; RANGACHAR, Kasturi y BRIAN, Shunck. Machine Vision. Singapore: McGraw-Hill Book Co.,1995.
- [7]. KRUGLINSKI, David J; SHEPHERD, George y WINGO, Scot. Programación Avanzada Con Visual C++. España: McGRAW HILL / INTERAMERICANA DE ESPAÑA. Primera edición en español, 1999.

[8]. KUZINA, T. Y. Investigación e implementación de los algoritmos del campo de visión dinámica por computadora [on line]. México: Universidad de Las Américas Departamento de ingeniería en sistemas computacionales, 2000 [citado en mayo de 2004]. Disponible en <http://140.148.3.250/udl/servlet/mx.udlap.ict.tales.html.Block?Thesis=42&Type=T>

[9]. MAYNERO, Carlos. Desarrollo De Un Sistema De Captura Y Análisis De Imágenes Para La Determinación Del Color Del Esmalte De Dentales. Aplicación A La Confección De Fundas Dentales[on line]. España: Universidad de Valencia, 2002[citado en julio de 2004]. Disponible en <http://www.maynero.com/carlos/proyecto/>

[10]. MATROX IMAGING LIBRARY (MIL). User Guide. Versión 5.1. 1998. Guía en formato digital.

[11]. MATROX IMAGING LIBRARY (MIL). Command Reference. Versión 5.1. 1998. Guía en formato digital.

[12]. PAJARES, Gonzalo y DE LA CRUZ, Jesús. Visión Por Computador: Imágenes Digitales Y Aplicaciones. México: Alfaomega, 2002.

[13]. PEREGRINA, Darío. Seguimiento de objetos por medio de visión activa [on line]. México: Instituto Nacional de Astrofísica, Óptica y Electrónica, 2002[citado en junio de 2004]. Disponible en <http://ccc.inaoep.mx/~labvision/tesis.htm>

[14]. RUSS John. The Image Processing Handbook. Boca Raton, FL. CRC PRESS, IEEE PRESS. 2002.

[15]. THE IMAGING SOURCE DFG / COMPRESS. Users Manual. Versión 1.1 US Edition, 2000. Guía en formato digital.

ANEXO A

RESULTADOS DE LAS PRUEBAS DE POSICIÓN Y ORIENTACIÓN

A.1 RESULTADOS PRUEBA DE POSICIÓN

A continuación se muestran los resultados detallados para cada algoritmo en una tabla. Además se puede observar una representación grafica de la distribución del error en las coordenadas x e y.

A.1.1 Algoritmo basado en el modelo de color HLS

En la tabla A.1, para el modelo de color HLS, en la mayoría de las posiciones seleccionadas, el error en valor absoluto de las coordenadas x e y no supera 0.8 cm y la desviación estándar para las mismas es menor a 0.1298.

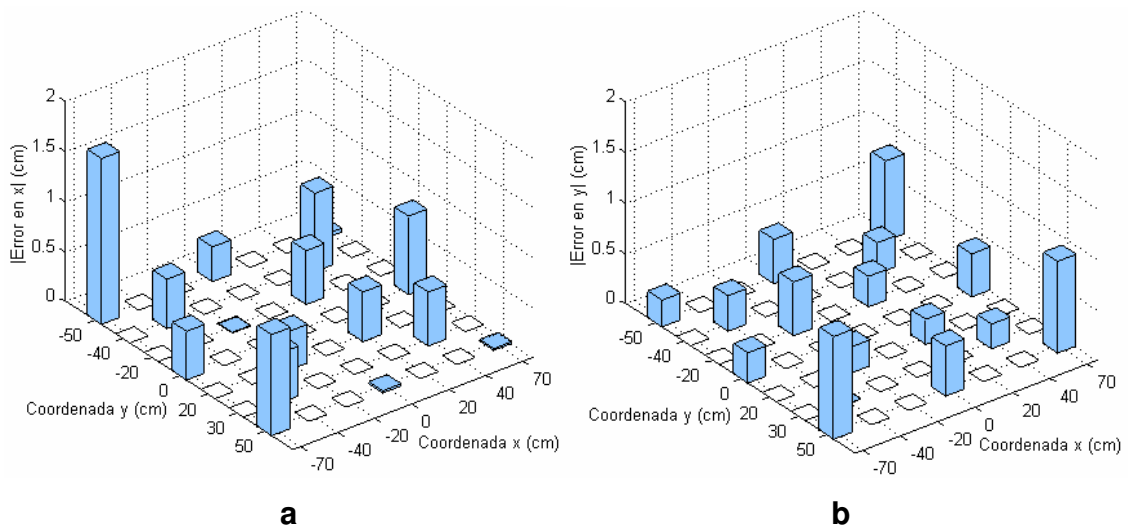
Tabla A.1 Estadísticas de posición utilizando el modelo de color HLS.

Nº	Posición Real (cm)	Medidas en x (cm)			Medidas en y (cm)		
		Valor Medio (cm)	Desviación estándar	Error (cm)	Valor Medio (cm)	Desviación estándar	Error (cm)
1	(70,-50)	70.0381	0	0.0381	-50.7900	0	-0.7900
2	(-70,-50)	-68.3434	0	1.6566	-49.7353	0	0.2647
3	(-70,50)	-68.9902	0	1.0098	48.9834	0	-1.0166
4	(70,50)	70.0168	0	0.0168	49.0923	0	-0.9077
5	(0, -50)	0.3469	0.0962	0.3469	-50.4403	0.0003	-0.4403
6	(-70,0)	-69.5129	0.0001	0.4871	-0.2994	0.0711	-0.2994
7	(70,0)	70.7840	0.0878	0.7840	-0.4263	0.0006	-0.4263
8	(0,50)	-0.0218	0.0512	-0.0218	50.4955	0	0.4955
9	(-20,-20)	-19.9929	0	0.0071	-20.5314	0	-0.5314
10	(20,-20)	20.5385	0	0.5385	-20.3028	0	-0.3028
11	(-20,20)	-20.3871	0.1298	-0.3871	20.2580	0	0.2580
12	(20,20)	20.4957	0	0.4957	20.2488	0	0.2488

13	(-40,-40)	-39.5058	0	0.4942	-40.3517	0	-0.3517
14	(40,-40)	40.7835	0.0004	0.7835	-40.3073	0.0520	-0.3073
15	(-40,30)	-40.4978	0	-0.4978	30.0018	0	0.0018
16	(40,30)	40.5379	0	0.5379	30.2446	0	0.2446

Figura A.1 Distribución del error en la escena para el modelo de color HLS

a) Error en la coordenada x. b) Error en la coordenada y



A.1.1 Algoritmo basado en el modelo de color xyY

Los resultados en la tabla A.2, muestran que el modelo de color xyY presenta un error en valor absoluto de las coordenadas x e y que no supera 0.7 cm en la mayoría de las posiciones elegidas. La desviación estándar es menor a 0.1538 en ambas coordenadas.

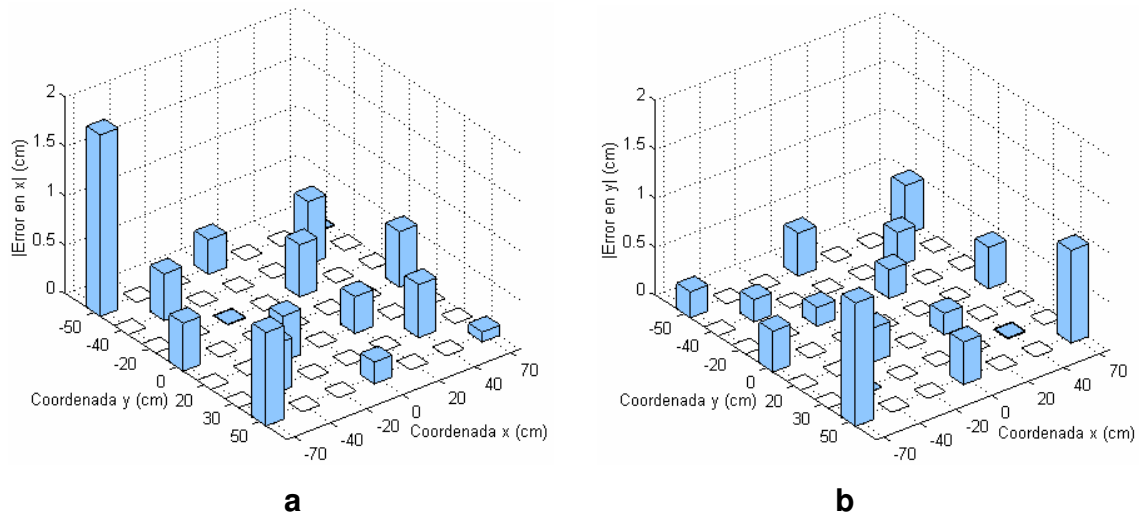
Tabla A.2 Estadísticas de posición utilizando el modelo de color xyY.

Nº	Posición Real (cm)	Coordenada x (cm)			Coordenada y (cm)		
		Valor Medio	Desviación estándar	error	Valor Medio	Desviación estándar	error
1	(70,-50)	70.0080	0.0759	0.0080	-50.5023	0.0746	-0.5023
2	(-70,-50)	-68.1577	0.1182	1.8423	-49.7344	5.6549e-004	0.2656

3	(-70,50)	-69.0518	0.1120	0.9482	48.7372	0.0513	-1.2628
4	(70,50)	69.9001	0.1340	-0.0999	49.0503	0.1221	-0.9497
5	(0,-50)	0.3525	0.1015	0.3525	-50.4403	2.7752e-004	-0.4403
6	(-70,0)	-69.5130	1.1365e-004	0.4870	-0.4189	0.1309	-0.4189
7	(70,0)	70.5631	0.0564	0.5631	-0.4249	0.1128	-0.4249
8	(0,50)	-0.2211	0.1011	-0.2211	50.4373	0.1099	0.4373
9	(-20,-20)	-19.9928	2.9042e-005	0.0072	-20.1924	0.1223	-0.1924
10	(20,-20)	20.5282	0.0514	0.5282	-20.2925	0.0514	-0.2925
11	(-20,20)	-20.4589	0.0959	-0.4589	20.3503	0.1256	0.3503
12	(20,20)	20.3726	0.1308	0.3726	20.2181	0.1538	0.2181
13	(-40,-40)	-39.5262	0.0711	0.4738	-40.2280	0.1315	-0.2280
14	(40,-40)	40.6383	0.1315	0.6383	-40.3475	0.1297	-0.3475
15	(-40,30)	-40.4978	0	-0.4978	30.0018	0	0.0018
16	(40,30)	40.5379	0	0.5379	29.9881	3.6260e-015	-0.0119

Figura A.2 Distribución del error en la escena para el modelo de color xyY

a) Error en la coordenada x. b) Error en la coordenada y.



Aunque ambos modelos de color presentan una exactitud semejante ya que el error para la mayoría de los puntos se encuentra por debajo de 0.8 cm, el modelo HLS tiene una mayor precisión debido a que la desviación estándar para los puntos localizados es inferior a la obtenida con el modelo xyY.

A.1.3 Algoritmo basado en correlación

Tabla A.3 Estadísticas de posición utilizando el algoritmo basado en correlación

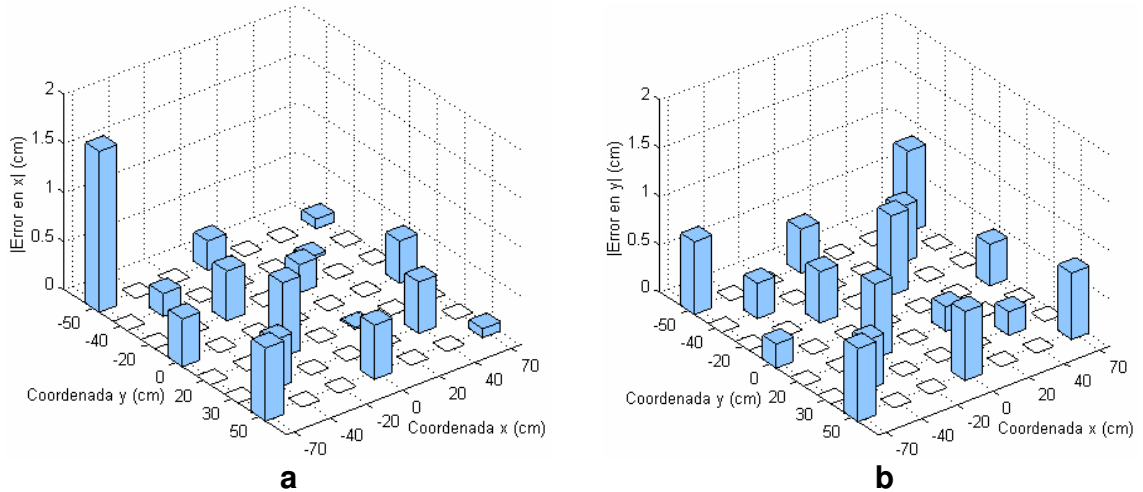
	Posición Real cm)	Coordenada x (cm)			Coordenada y (cm)		
		Valor Medio	Desviación estándar	error	Valor Medio	Desviación estándar	error
1	(70,-50)	69.9005	0.1459	-0.0995	-50.8399	0.1049	-0.8399
2	(-70,-50)	-68.3567	0.0259	1.6433	-49.2463	1.3347e-004	0.7537
3	(-70,50)	-69.2527	0	0.7473	49.2405	0	-0.7595
4	(70,50)	69.9093	0.1250	-0.0907	49.3114	0.0380	-0.6886
5	(0,-50)	0.3023	0.0507	0.3023	-50.4581	0.0730	-0.4581
6	(-70,0)	-69.5119	0.0538	0.4881	-0.2498	0.1114	-0.2498
7	(70,0)	70.4308	0.1169	0.4308	-0.4310	7.9977e-004	-0.4310
8	(0,50)	-0.5530	0.0828	-0.5530	50.7150	0.1180	0.7150
9	(-20,-20)	-20.5069	8.6529e-006	-0.5069	-20.5304	0.0353	-0.5304
10	(20,-20)	20.2880	0	0.2880	-20.8248	4.2745e-014	-0.8248
11	(-20,20)	-20.7571	0	-0.7571	20.7719	0	0.7719
12	(20,20)	19.9822	0	-0.0178	20.2514	0	0.2514
13	(-40,-40)	-39.7664	0	0.2336	-40.3664	0	-0.3664
14	(40,-40)	40.0306	0	0.0306	-40.5849	0	-0.5849
15	(-40,30)	-40.5004	0	-0.5004	30.5156	0	0.5156
16	(40,30)	40.5248	0	0.5248	30.2439	0	0.2439

En la tabla A.3 se observa que para el modelo de color xyY, en la mayoría de las posiciones seleccionadas, el error en valor absoluto de las coordenadas x e y no supera 0.8399 cm, así mismo la desviación estándar es menor a 0.1538 en ambas coordenadas.

A continuación se puede observar en la figura A.3 la distribución del error en las coordenadas x e y en la escena de trabajo.

Figura A.3 Distribución del error en la escena para el algoritmo de correlación

a) Error en la coordenada x. b) Error en la coordenada y



A.2 RESULTADOS PARA LA PRUEBA DE ORIENTACIÓN

A continuación se muestran los resultados de la prueba de orientación para los tres algoritmos implementados. Para cada uno de ellos se utilizó una tabla con el valor medio, la desviación estándar y el error de orientación, realizando un barrido de 0° a 360° . También se observa una representación grafica del error de orientación.

A.2.1 Algoritmo de Color basado en el modelo HLS

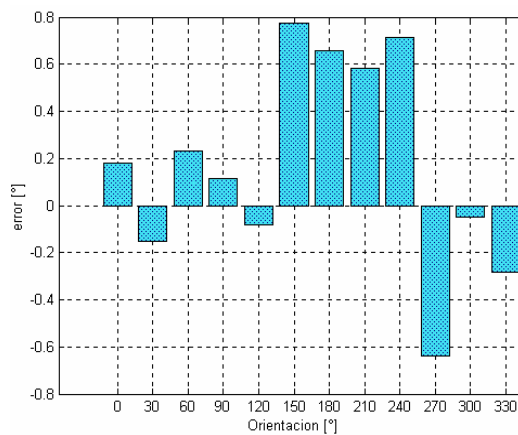
Tabla A.4 Estadísticas de orientación para el modelo HLS

Orientación Real ($^\circ$)	Orientación Calculada ($^\circ$)		
	Valor Medio	Desviación estándar	error
0	0.1822	0.1030	0.1822
30	29.8475	0	-0.1525
60	60.2316	0	0.2316
90	90.1166	0.1561	0.1166
120	119.9188	0.1028	-0.0812
150	150.7737	0.0449	0.7737

180	180.6569	0.1264	0.6569
210	210.5831	0	0.5831
240	240.7156	0.0523	0.7156
270	269.3597	0.0021	-0.6403
300	299.9492	0.0738	-0.0508
330	329.7177	0.0251	-0.2823

En la tabla anterior se puede observar que el error en valor absoluto es inferior a 0.8° y que el máximo valor de la desviación estándar es 0.1561.

Figura A.1 Error en orientación para el modelo HLS



A.2.2 Algoritmo de Color basado en el modelo xyY

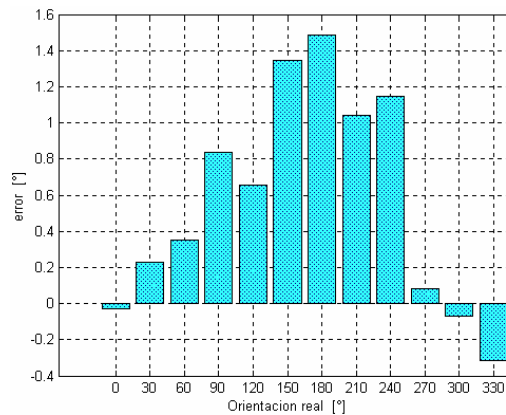
Tabla A.5 Estadísticas de orientación para el modelo xyY

Orientación Real (°)	Orientación Calculada (°)		
	Valor Medio	Desviación estándar	error
0	-0.0282	0.1094	-0.0282
30	30.2279	0.1301	0.2279
60	60.3499	0.1006	0.3499
90	90.8363	0.2003	0.8363
120	120.6562	0.0674	0.6562
150	151.3432	0.0661	1.3432
180	181.4840	0.0028	1.4840

210	211.0430	0.0934	1.0430
240	241.1486	0.2014	1.1486
270	270.0796	0.1163	0.0796
300	299.9304	0.1053	-0.0696
330	329.6822	0.0772	-0.3178

Para este modelo de color, puede verse que el error en valor absoluto es inferior a 1.485° y que el máximo valor de la desviación estándar es 0.2014.

Figura A.2 Error en orientación para el modelo xyY



A.2.3 Algoritmo basado en correlación

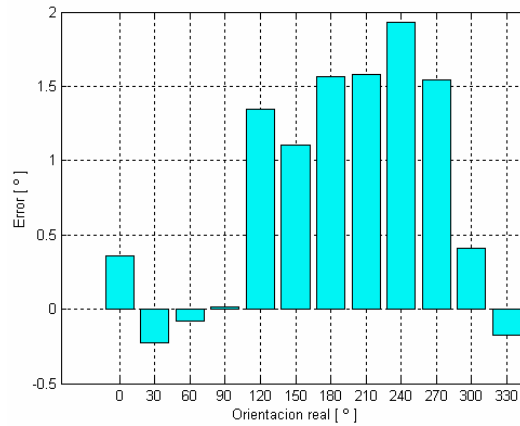
Tabla A.6 Estadísticas de orientación para el algoritmo basado en Correlación

Orientación Real (°)	Coordenada x (cm)		
	Valor Medio	Desviación estándar	Error
0	0.3590	0	0.3590
30	29.7729	0	-0.2271
60	59.9167	0	-0.0833
90	90.0118	0	0.0118
120	121.3432	0	1.3432
150	151.1034	0	1.1034
180	181.5681	0.0058	1.5681
210	211.5811	0	1.5811

240	241.9291	0	1.9291
270	271.5414	0	1.5414
300	300.4069	0	0.4069
330	329.8235	0	-0.1765

El algoritmo basado en correlación presenta un error en valor absoluto inferior a 1.9292° y un máximo valor de la desviación estándar 0.0058.

Figura A.3 Error en orientación para el algoritmo basado en correlación



A.2.4 Algoritmo basado en gradientes

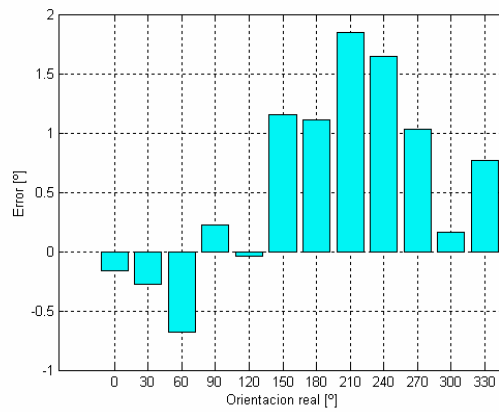
Tabla A.6 Estadísticas de orientación para el algoritmo basado en gradientes

Orientación Real(°)	Orientación medida (°)		
	Valor Medio	Desviación estándar	error
0	0.1585	0.3357	0.1585
30	29.7279	0.3470	-0.2721
60	59.3241	0.3589	-0.6759
90	90.2248	0.2832	0.2248
120	119.9637	0.5169	-0.0363
150	151.1494	0.3880	1.1494
180	181.1140	0.3546	1.1140
210	211.8446	0.3217	1.8446
240	241.6463	0.3452	1.6463

270	271.0272	0.3107	1.0272
300	300.1634	0.2708	0.1634
330	330.7685	0.5001	0.7685

En la tabla anterior se observa que el máximo error en valor absoluto 1.8446° y que el máximo valor de la desviación estándar es 0.5169.

Figura A.4 Error en orientación para el algoritmo basado en gradientes



ANEXO B
PRINCIPALES CARACTERÍSTICAS DE LA TARJETA DIGITALIZADORA
DFG/COMPRESS PIRANHA Y DE LA CÁMARA DFK 50H13

B.1 TARJETA DIGITALIZADORA DFG/COMPRESS PIRANHA

La DFG/Compress es una tarjeta de compresión de video de 32 bit PCI, adaptable a multimedia, CTV y aplicaciones médicas. Todas las fuentes de video convencionales con S-VHS (YC) y video compuesto están soportadas por ésta tarjeta de compresión. Las cámaras pueden ser conectadas usando las cuatro entradas de la tarjeta. La transferencia de datos de imagen al PC o a la tarjeta gráfica VGA se hace usando transferencia DMA en burst mode. Usando ésta técnica la CPU no es sobrecargada y de esta manera el PC opera normalmente.

El flujo de imágenes en vivo de salida en el monitor puede ser guardado al disco duro como una secuencia de imágenes AVI en una determinada resolución (máximo 768 x 576) y un número de colores (máximo 32 bit). Adicionalmente, información hardware, como los datos actuales y el tiempo, pueden ser guardados en la imagen de video. Una vez está información ha sido insertada en el flujo de video, no puede ser removida posteriormente. La señal de video sale de un conector de video S-Video (compuesto). Videos previamente grabados en disco duro y el flujo de video en vivo desde una cámara conectada a la tarjeta pueden salir de este conector.

Alternativamente, los datos de video pueden ser escritos a la VGA (y posteriormente al monitor del PC). Para esta funcionalidad, los drivers de la DFG/Compress usan la interfaz DirectDraw de Microsoft.

B.1.1 CARACTERÍSTICAS

- 4 fuentes monocromáticas / 4 fuentes de color compuesto o 4 fuentes Y/C pueden ser conectadas (multiplexadas).
- Soporte multi-estándar (CCIR, PAL; NTSC; RS170, SECAM).
- 4 entradas de disparo.
- 8 entradas y 8 salidas digitales (TTL) – opcionalmente circuitería acoplada ópticamente.
- Soporta distintas resoluciones (representación de la imagen, máximo 768 x 576).
- Conversor de color de Y/C (YUV) a RGB en tiempo real.
- Soporta formatos de verdadero color, RGB de 24 bit y RGB de 32 bit.
- Máxima resolución de 768 x 576 a 50 Hz (PAL / CCIR) y 640 x 480 a 60 Hz (NTSC / RS170).
- Soporte overlay por medio de la interfaz DirectDraw de Microsoft.
- Escalado por hardware de la imagen en vivo.
- EEPROM para escribir y leer datos.
- Interfaz programable para Windows NT 4.0.
- Capacidad de guardar secuencias de video a archivos AVI.
- Compatible con PCI 2.1.
- Transferencia directa de datos a la VGA y a la memoria del sistema.
- Interface maestra PCI DMA.

B.1.2 REQUERIMIENTOS DEL SISTEMA

Los siguientes requerimientos son necesarios para que la DFG/Compress puede ser usada:

- PC con procesador de 100 MHz o mayor.

- Compatibilidad con PCI 2.1.
- Un slot PCI libre de 32 bit.
- Por lo menos 32 MB de RAM (recomendado 64 MB para Windows NT)-
- Windows NT 4.0.

Adicionalmente se recomienda lo siguiente:

- Disco duro SCSI para guardar secuencias de video.
- Monitor externo para verificar el video grabado y la salida de video.

B.1.3 TARJETA VGA

Se recomienda una tarjeta VGA con por lo menos 2 MB de DRAM para aplicaciones monocromáticas y 4 MB de memoria para aplicaciones a color. Solamente con estas especificaciones, se puede lograr resultados con la máxima resolución y número de colores. Es importante que el driver de DirectDraw de la VGA esté instalado. A pesar de que es posible visualizar imágenes sin DirectDraw, existen muchas limitaciones a la frecuencia de repetición de imágenes y al desplegar el overlay. Los mejores resultados pueden ser obtenidos usando tarjetas con chipset ATI (RagePro).

B.1.4 INSTALACIÓN HARDWARE

A continuación se describe como instalar la DFG/Compress en el PC. Antes de la instalación, se debe aislar el PC desconectando el conector de potencia del enchufe en la pared. Para descargar cualquier carga estática almacenada, se debe tocar la cubierta del PC. Luego de esto, cualquier carga estática guardada es seguramente descargada. La instalación de la DFG/Compress es muy fácil, se debe seguir los pasos mostrados a continuación:

1. Apague el computador y desconecte el cable de potencia. En computadores nuevos con tarjeta madre ATX siempre que el computador se apaga, el bus PCI puede aun estar energizado, por lo cual al insertar la DFG/Compress en la tarjeta madre lleva a la destrucción de ambas tarjetas.
2. Abra el computador quitando los tornillos.
3. Quite el metal del slot PCI el cual no está en uso.
4. Inserte a DFG/Compress en el slot PCI disponible en el computador. Es importante no tocar algún componente en la DFG/Compress.
5. Ajuste la tapa del slot de la tarjeta con un tornillo.
6. Asegúrese que la tarjeta ha sido correctamente instalada en el slot.
7. Tape de nuevo el computador.
8. Conecte la entrada de video.

Se pueden conectar hasta cuatro entradas (compuesta o S-Video) a la DFG/Compress, usando un conector DSUB de 44 pines.

B.1.5 ADQUISICIÓN DE VIDEO

Hasta cuatro fuentes de video pueden ser conectadas a la DFG/Compress. Las entradas pueden ser S-VHS (S-Video / YC, Hi8) o fuentes de video compuesto (VHS, Video8, FBAS, NTSC, RS170, CCIR). LA conexión física entre la fuente de video y la tarjeta se hace con un cable opcional a la salida de un conector DSUB de 44 pines en un lado y un conector BNC (para video compuesto) o conectores Hosiden de 4 pines (S-VHS) en el otro lado.

Resolución de color: La adquisición y compresión de video se hace en un formato de verdadero color YUC 4:2:2. La visualización en la VGA puede ser en diferentes modos: 15 bit, 16 bit (alto color), verdadero color de 24 bit (16.7 millones de colores) o en modo de verdadero color de 32 bit.

Entrada de video: 4 entradas externas: S-Video / YC (S-VHS, Hi8) o compuesto.

Frecuencia de scan de video: PAL/SECAM: 14.75 MHz, NTSC: 12,27 MHz.

Conversión A/D: PAL: 1/50 s (20 ms) por campo, NTSC: 1/60 s (16.7 ms) por campo.

Resolución geométrica: PAL: 768 x 576 píxel en 2 campos, NTSC: 640 x 480 píxel en 2 campos.

B.1.6 ADQUISICIÓN DE IMAGEN CON UNA CÁMARA:

Si se desea adquirir datos de video con solo una cámara, no es necesario usar el conector DSUB de 44 pines. En este caso la DFG/Compress tiene un conector adicional BNC. Los datos desde la cámara pueden ser transferidos en tiempo real (25 imágenes/seg en PAL/SECAM y 30 imágenes/seg en NTSC), por medio de la memoria FIFO, directamente al disco duro del PC. En éste modo de operación, el multiplexor interno no se activará.

B.1.7 SALIDA DE VIDEO

El DFG/Compress tiene dos salidas de video de 50Hz/60Hz que son independientes entre si. Si una fuente 50 Hz SVHS de video se conecta a una de las entradas de la tarjeta, exactamente la misma señal está disponible en sus salidas. En otras palabras, usando la tarjeta de este modo no se deteriora la señal de video en lo absoluto. Usando la salida de video 1, una de las cuatro entradas de video puede ser colocada directamente a la salida, desviándose así del proceso de digitalización de la tarjeta. La salida de la entrada seleccionada de video no tiene influencia en ninguna grabación de video, la cual puede estar

corriendo al mismo tiempo. En la salida de video 2 se fija el video corriente proveniente de una cámara o la señal de video sin comprimir, dependiendo de cual modo se encuentra seleccionado en la tarjeta.

Modo grabación (recording): Entre la entrada de video y la salida de video la señal es análoga. Cualquier revestimiento presente no es enviado a la salida de video.

Modo cinta (playback): La imagen descomprimida es una secuencia de video AVI. Cualquier revestimiento presente esta contenido en la imagen descomprimida. Adicionalmente la salida de video puede ser vista en un monitor VGA.

Resolución en color: La descompresión y la salida de la señal de video 2 es siempre en YUV 4:2:2 formato de color verdadero. El despliegue en la tarjeta VGA, puede ser en 15 bits, 16 bits (alto color), 24 bits en modo de verdadero color (16.7 millones de colores) o 32 bits en modo de verdadero color.

B.1.8 ORGANIZACIÓN DE LA MEMORIA PARA IMÁGENES SIN COMPRIMIR

Las imágenes sin comprimir pueden capturarse singularmente o como una secuencia. La adquisición de la secuencia no tiene nada que ver con el almacenamiento en memoria de la secuencia de video comprimido. Durante la adquisición, la secuencia de imágenes es almacenada en una o más memorias de imagen (memorias). Antes de esto puede elegirse un lugar de almacenamiento, la memoria de imagen debe ser primero asignada con el driver e insertada dentro de la lista de la secuencia. La preparación de la memoria de imagen toma lugar con el driver, cuando se ha localizado físicamente y de forma lineal (una al lado de la otra) las posiciones de memoria para la secuencia que se almacena. Note que una memoria con estas características puede ser definida solamente cuando su

sistema esta iniciando. Solo una limitada parte de la memoria principal esta disponible para la asignación de la memoria de imagen y no puede ser modificada en su tamaño mientras el sistema se encuentra en funcionamiento.

B.1.9 INSTALACION DEL SOFTWARE

La DFG/Compress SDK (funciones API) y el driver de video para Windows están contenidos ambos en el DLL *piranha.dll*. Durante la instalación el driver de video para Windows se introduce en el registro del directorio correspondiente. Este es el mismo durante la instalación de video para Windows y el SDK.

Aviso Importante:

Para instalar la tarjeta DFG/Compress bajo Windows NT, los derechos del administrador se necesitan. En el caso de instalar la tarjeta por primera vez o realizar un cambio de tarjeta en su equipo.

Numero de serial:

El numero serial de la tarjeta DFG/Compress esta impreso en un sticker blanco sobre la tarjeta del circuito y además esta almacenado en la EEPROM en la tarjeta. Cada tarjeta DFG/Compress tiene un único número de serial el cual puede ser mostrado con el programa *lcbInfo.exe*

Instalación por primera vez:

Windows NT 4.0:

Para instalar el software simplemente inserte el disco 1 en su unidad de disquetes y corra el SETUP, el menú de este programa es guiado y auto explicativo. La rutina de instalación copia los drivers en los siguientes directorios:

icbvcd32.sys	a	\<windir>\system32\drivers
icbmem32.sys	a	\<windir>\system32\drivers
piranha.dll	a	\<windir>\system32
icbhook.dll	a	\<windir>\system32
piranha.ini	a	\<windir>
icblogic.rbf	a	\<windir>

La estructura del directorio es creada de la siguiente forma:

<Install_dir>\Piranha\Program	Programas de ayuda
\Piranha\Samples\Exe	ejemplos de programas
\Piranha\Samples\Source	código de los programas de ejemplo
\Piranha\Help	información adicional

(<windir> es el directorio de Windows. <Install_dir> es el directorio en el cual el software de la DFG/Compress es instalado.)

Windows 9X:

Bajo Windows 9x la DFG/Compress será reconocida automáticamente como un nuevo dispositivo multimedia, preguntando así por el driver correspondiente para la instalación, cuando pregunte, por favor inserte el disco en la unidad de disquetes. El driver de la DFG/Compress se encuentra en este disco, entonces simplemente siga las instrucciones en pantalla y cuando pregunte que hacer, reinicie su sistema, esto es necesario, ya que el driver será reconocido solamente cuando su sistema se encuentre iniciando. Cuando su sistema esta corriendo de nuevo, inserte el disco y ejecute el programa *SETUP* y de nuevo siga las

instrucciones en pantalla. Durante la instalación los siguientes archivos son copiados en su PC:

icbvcd32.vxd	a	\<windir>\system
piranha.dll	a	\<windir>\system
icbhook.dll	a	\<windir>\system
piranha.ini	a	\<windir>
icblogic.rbf	a	\<windir>

Una vez su sistema reinicia, inicie el programa de demostración. Si su PC no reinicia correctamente o se bloquea mientras esta iniciando, no se preocupe! , simplemente oprima el botón de reset que se encuentra en el exterior de su PC, apague y espere 60 segundos y de nuevo prenda su PC, Windows deberá comenzar correctamente.

B.1.10 MODOS DE SALIDA DE LA TARJETA DFG/COMPRESS

Modo de mapa de bits (BMP/DIB):

En este modo el video capturado de la DFG/Compress es almacenado en la memoria principal del PC y luego copiado a la tarjeta VGA, aunque de esta forma, la tasa de actualización en pantalla es más lenta, Sin embargo la mayoría de computadoras modernas están en capacidad de desplegar imágenes en tiempo real usando la DFG/Compress. El modo mapa de bits es particularmente ventajoso, debido a la alta compatibilidad que tiene con todas las tarjetas VGA y directo acceso de los datos de imagen a la memoria principal. Las funciones de cubierta (mezclado en una imagen de texto y gráficos) tienen que ser programadas por el usuario. Como Windows toma el control del despliegue de las imágenes, la imagen en vivo puede ser parcial o totalmente cubierta por muchas otras ventanas y cajas de dialogo.

DirectDraw Primary Surface Mode:

En este modo los datos de la imagen son escritos en el área visible de la tarjeta VGA. Brinda el máximo desempeño de 25 imágenes por segundo con la DFG/Compress. En este caso no es posible usar cubiertas (mezclado en una imagen de texto y gráficos) debido a que los datos de cubierta deben ser constantemente sobrescritos en el siguiente frame lo cual nunca permitiría su visualización. En contraste con muchas otras tarjetas de compresión, en la DFG/Compress es posible posicionar ventanas, cajas de dialogo y otras aplicaciones de Windows sobre la ventana que contiene la imagen en vivo de la DFG/Compress sin perturbarla en lo absoluto, la tarjeta de compresión puede llevar a cabo esto, no desplegando automáticamente la imagen en vivo bajo las ventanas que la cubren y manteniendo una alta frecuencia de actualización.

DirectDraw Overlay Surface Mode:

En este modo, se puede lograr simultáneamente, la velocidad de salida del modo DirectDraw Primary Surface Mode y el despliegue de los datos de cubierta. Esta es la ventaja más grande, ya que las tarjetas VGA las implementan a un costo muy bajo. La imagen de video es digitalizada a un área no visible de la tarjeta VGA. El chip de la VGA permite desplegar la imagen de video o el contenido original de la ventana de salida en la que se despliegan los datos, exactamente cualquiera de estos dos modos se usa dependiendo del color que se haya programado. Definiendo el tipo de color y usando este para dibujar en la ventana de salida, la imagen solo podrá ser mostrada en las partes donde el tipo de color elegido sea usado, naturalmente si toda la ventana esta llena solamente del tipo de color elegido, la imagen se visualizara completamente. Ahora, textos y gráficos usados en la cubierta que no usen el tipo de color elegido permanecerán disponibles, esto se conoce como *cubierta no-destructiva*. El mezclado de la cubierta es realizado por el chip de la VGA y apenas se requieren algunos ciclos

de CPU. Este modo no es soportado por todos los chips VGA y la DFG/Compress solo soporta modos de 15 y 16 bits en los VGA.

B.1.11 COLOR Y FORMATOS DE ALMACENAMIENTO

Cada uno de los formatos de color soportado por la tarjeta DFG/Compress usa diferentes formatos con lo cual se almacenan las imágenes. Como se muestra en la siguiente tabla:

Format	DWORD	Pixel Data [31:0]			
		Byte 3 [31:24]	Byte 2 [23:16]	Byte 1 [15:8]	Byte 0 [7:0]
RGB32	dw0	0	R0	G0	B0
	dw1	0	R1	G1	B1
RGB24	dw0	B1	R0	G0	B0
	dw1	G2	B2	R1	G1
	dw2	R3	G3	B3	R2
RGB16	dw0	{R1[15:11], G1[10:5], B1[4:0]}*		{R0[15:11], G0[10:5], B0[4:0]}*	
RGB15	dw0	{0, R1[14:10], G1[9:5], B1[4:0]}*		{0, R0[14:10], G0[9:5], B0[4:0]}*	

* Cuando una imagen esta en formato RGB15 o RGB16 los bits superiores al bit interno 8 son usados en los colores R G y B.

B.2 CÁMARA DFK 50H13

A continuación se muestran las principales características de la cámara DFK 50H13.

Formato de la señal: NTSC

Sensor de imagen: 1/3" interlineado CCD.

Imagen efectiva: 768 (H) x 494 (V).
Frecuencia Horizontal: 15.734 kHz
Frecuencia Vertical: 59.94 Hz
Mínima iluminación: 1 lux.
Radio S/N: mas de 48dB (AGC=OFF).
 γ : 0.45.
Balance blanco: automático / manual.
Montaje de lente: CS o C (adptador).
AGC: ON / OFF, seleccionable.
Salida de Video: VBS 1.0 Vp-p a 75Ω , Y/C.
I/O serial: RS232C.
Alimienación: 10 ~ 12 V DC.
Potencia consumida: 180 mA.
Temperatura de operación: -10°C ~ $+40^{\circ}\text{C}$.
Dimensiones: 51mm (ancho) X 51 mm (alto) X 55.5 mm (profundidad).
Peso: 190g.

ANEXO C

INSTALACIÓN E INTERFASE GRAFICA DE “FENIX”

FENIX es una aplicación software desarrollada en *Microsoft Visual C++ 6.0*, la cual tiene como objetivo el análisis dinámico de imágenes en una escena particular.

C.1 INSTALACIÓN

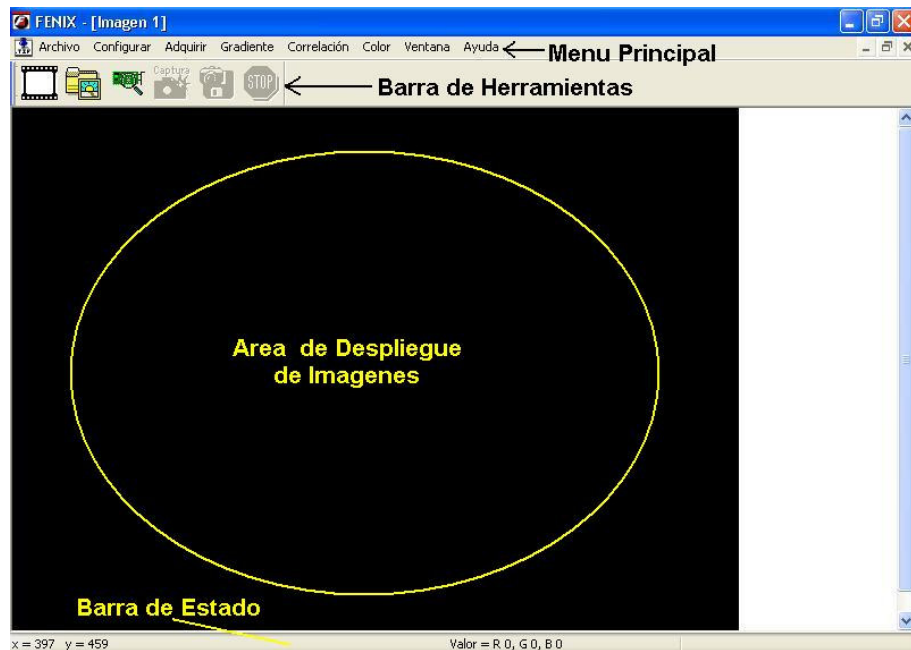
A continuación se describen los pasos a seguir para una correcta instalación de la aplicación FENIX.

1. Instale correctamente la tarjeta digitalizadora PIRANHA. En el anexo B se encuentra de forma detallada la instalación de la misma.
2. Copie la carpeta *Fenix* con todos sus componentes, en algún lugar del disco duro del PC.
3. Ejecute el archivo *fenix.exe*, el cual se encuentra dentro de la carpeta *Fenix* copiada a su PC.

C.2 INTERFASE GRAFICA

En la figura C.1 se observa la ventana principal de FENIX.

Figura C.1. Ventana principal de FENIX



La ventana principal de FÉNIX tiene diferentes características: en la parte superior se encuentra el menú principal, del cual se puede seleccionar todas las funciones de la aplicación. También se muestra la barra de herramientas que tiene acceso directo a algunas funciones. En la parte central se observa el área para desplegar imágenes. Finalmente en la parte inferior se muestra la barra de estado.

C.2.1 MENÚ PRINCIPAL

En la figura C.2 se observan las diferentes opciones del menú principal las cuales son: archivo, configurar, adquirir, gradiente, correlación, color, ventana y ayuda.

Figura C.2. Menú principal



C.2.1.1 Archivo

En la figura C.3 se aprecian las opciones del ítem archivo.

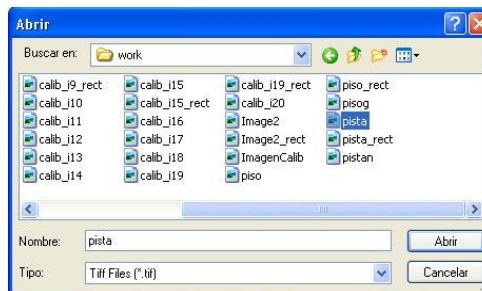
Figura C.3. Ítem archivo



Nuevo: Crea una nueva área (ventana) para desplegar una imagen.

Abrir: Carga una imagen en formato tiff ,desde una archivo existente, por medio del dialogo *abrir* de Windows (figura C.4).

Figura C.4. Abrir Imagen



Cerrar: Cierra una imagen activa.

Salir: Cierra la aplicación en su totalidad.

C.2.1.2 Ítem Configurar

Permite fijar los parámetros necesarios para la correcta ejecución de los algoritmos implementados en la aplicación. Consta de las siguientes opciones: adquisición, video, calibración, modelo de color y resultados. En la figura C.5 se muestra la manera como se despliegan la opciones en pantalla.

Figura C.5. Ítem configurar



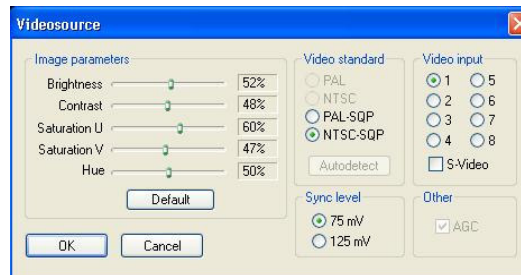
Adquisición: permite seleccionar por medio de una ventana de dialogo (figura C.6) la tarjeta con la cual se va a realizar el proceso de digitalización. Para el proyecto desarrollado la única tarjeta configurada fue la PIRANHA, aunque se dejan las opciones de otras tarjetas en el dialogo para posteriores mejoras de la aplicación.

Figura C.6. Dialogo de selección de la tarjeta digitalizadora



Video: permite manipular por medio de una ventana de dialogo (figura C.7) en la tarjeta piranha las propiedades de video como contraste, brillo, selección del canal de video, estándar de video etc.

Figura C.7. Dialogo de propiedades del video de la tarjeta digitalizadora PIRANHA



Calibración: En esta opción (figura C.8) se fijan los parámetros intrínsecos obtenidos en el proceso de calibración de la cámara (figura C.9) y el origen de coordenadas del mundo real en la imagen. (figura C.10).

Figura C.8. Ítem calibración



En la figura C.9 se observa el dialogo de entrada para los parámetros intrínsecos de calibración. En este dialogo también se introduce a la aplicación el factor de escala necesario para la conversión de píxeles a centímetros.

Figura C.9. Dialogo de entrada para los parámetros de calibración

Parámetros de Calibración

Fenix

Parámetros Intrínsecos

Longitud Focal
fc1 1077.18144 fc2 1074.90424

Punto Principal
cc1 328.63616 cc2 194.06424

Coefficientes de Distorsión
kc1 -0.3163 kc2 -0.31778 kc3 0.00186 kc4 0.00094

Factor de Escala
0.2564102 cm/pixel

Aceptar Cancelar

en la figura C.10 se muestra el dialogo de entrada para las coordenadas en píxeles del origen del sistema de referencia del mundo real en la imagen.

Figura C.10. Dialogo de entrada para el origen de coordenadas

Sistema de Referencia

Fenix

Coordenadas (píxeles)

Xo 320 Yo 240

Aceptar Cancelar

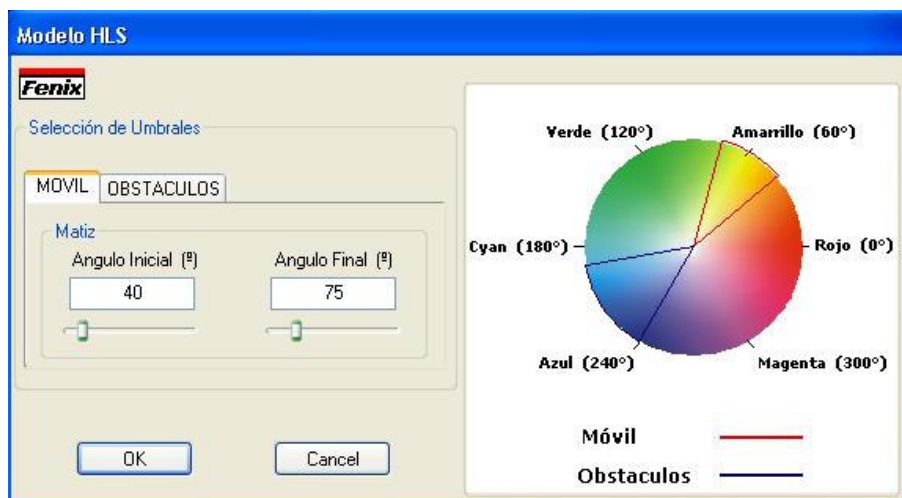
Modelo de color: Con este ítem (figura C.11) se puede seleccionar y configurar el modelo de color a utilizar. Se tienen las opciones de modelos HLS y xyY

Figura C.11. Ítem modelo de color



En la figura C.12 se observa el dialogo de configuración del modelo de color HLS. En la parte izquierda se configura el matiz del color a localizar en los móviles y los obstáculos. En la parte derecha se muestra de forma grafica y dinámica los cambios realizados en los valores de configuración del modelo

Figura C.12. Dialogo de configuración del modelo de color HLS.



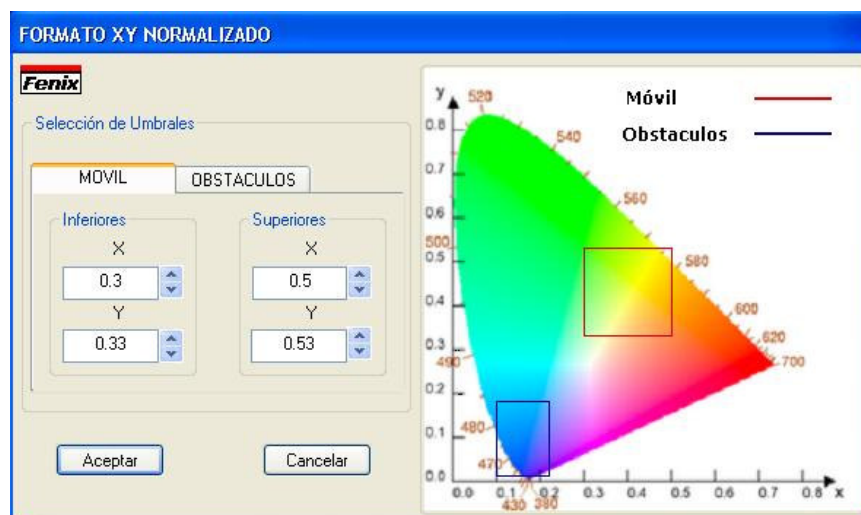
Al seleccionar el modelo de color xyY, se despliega la ventana de dialogo mostrada en la figura C.13, la cual tiene dos partes principales.

La primera se ubica a la izquierda y contiene la información de los umbrales utilizados para la segmentación de los objetos móviles y de los obstáculos. Para

las componentes x e y del modelo xyY se tiene un umbral inferior y un umbral superior.

La segunda está ubicada a la derecha y contiene información visual del color con el cual se desea segmentar cada tipo de objetos de la escena. Para esto se tiene una imagen de fondo del modelo de color xyY y sobre ésta se dibujan dos rectángulos. Uno de ellos encierra el color deseado para los objetos móviles y el otro encierra el color para los obstáculos.

Figura C.13. Dialogo para seleccionar umbrales en el modelo de color xyY



Resultados: este ítem tiene dos opciones: la primera se refiere a la unidad en la cual se desea desplegar la información de posición para los objetos de interés en la escena. Esta unidad de medida puede ser píxeles o centímetros (figura C.14).

Figura C.14. Selección de la unidad de medida para la posición de los objetos de Interés



La segunda indica el tipo de referencia (fija o móvil) para calcular la orientación de los objetos móviles (figura C.15).

Figura C.15. Selección de la referencia para calcular la orientación



La aplicación por defecto selecciona medidas de posición en centímetros y una referencia móvil para la orientación.

C.2.1.3 Ítem Adquirir

Este ítem tiene tres opciones: adquirir video, capturar un frame y guardar un frame desplegado, como se muestra a continuación:

Figura C.16. Ítem adquirir



Video: Esta opción permite adquirir video en una ventana de dialogo como se muestra en la figura C.17.

Figura C.17. Ventana de dialogo para desplegar el video



Frame: captura un frame (imagen) y lo despliega en la parte central de la ventana principal de la aplicación (figura C.18).

Guardar: guarda una imagen en formato tiff, por medio de la ventana “*guardar*” de Windows (figura C.19).

Figura C.18. Frame capturado y desplegado

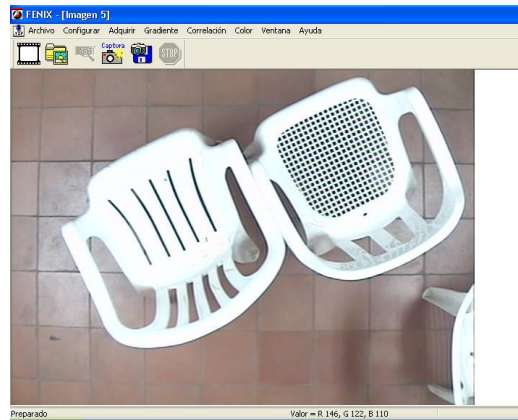


Figura C.19. Ventana para guardar una imagen.



C.2.1.4 Ítems gradiente, correlación y color

Estos ítems son los encargados del inicio de ejecución de los tres principales algoritmos implementados por medio de la opción *calcular* (figura C.20). Cabe resaltar que la aplicación solo deja ejecutar solo un algoritmo a la vez.

Figura C.20. Gradiente, correlación y color



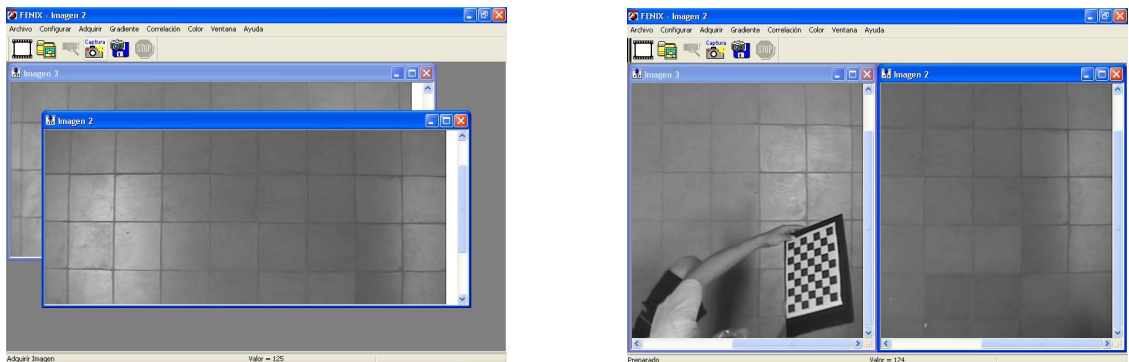
C.2.1.5 Ítem Ventana

Con este ítem se organizan las imágenes que se encuentran desplegadas en diferentes ventanas en forma de cascada (figura C.22-a) o de mosaico (figura C.22-b). También se muestra un lista con las imágenes actualmente desplegadas.

Figura C.21. Ítem ventana



Figura C.22 Organizar imágenes en forma de a) cascada b) mosaico.



a

b

C.2.2 BARRA DE HERRAMIENTAS

Esta barra está compuesta de iconos con los cuales tienen acceso directo a algunas de las funciones descritas anteriormente (figura C.23).

Figura C.23. Barra de herramientas



A continuación se relaciona cada icono con la función que ejecuta.



Despliega una nueva imagen en la cual se puede adquirir una imagen



Abre una imagen ya existente.



Abre el dialogo de selección de la tarjeta digitalizadora.



Captura una imagen desde la cámara



Guarda una imagen

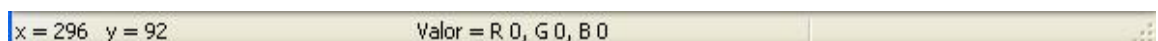


Detiene el procesamiento del algoritmo en ejecución.

C.2.3 BARRA DE ESTADO

La barra de estado muestra dos tipos de información: a la izquierda las coordenadas por donde pasa el cursor del mouse sobre la imagen y la derecha el valor RGB del píxel en dichas coordenadas, como se muestra en la figura C.24.

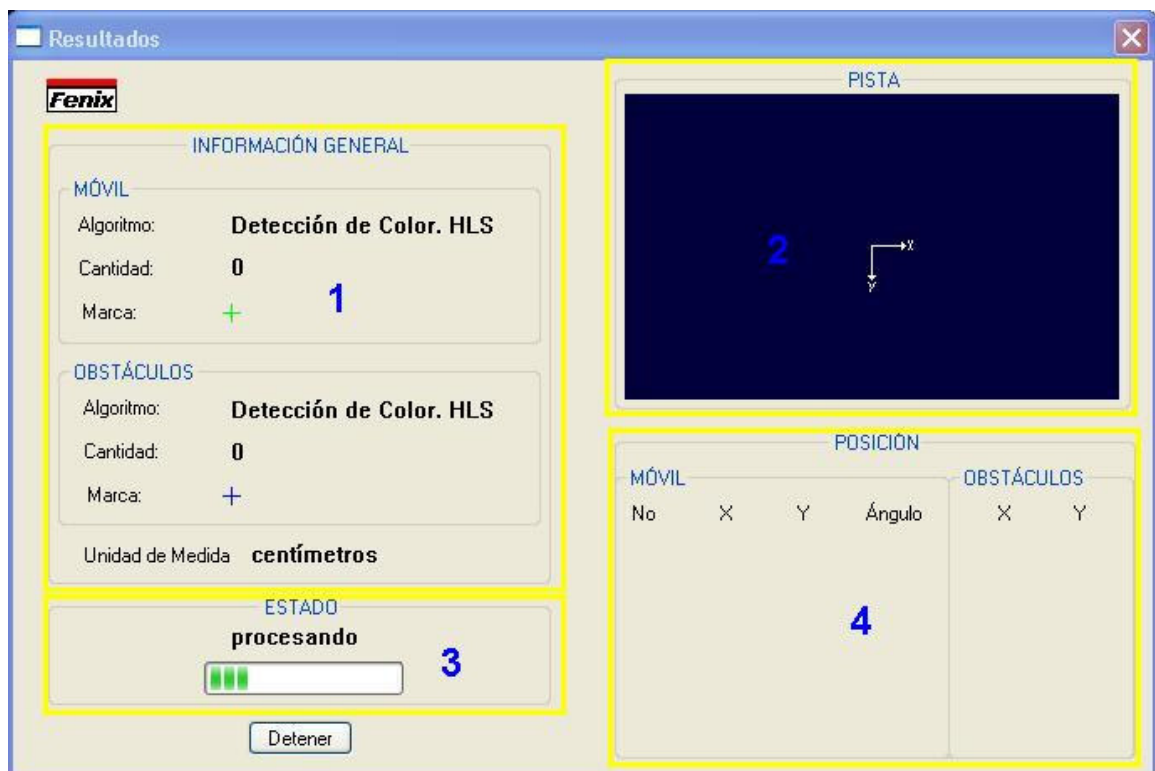
Figura C.24. Barra de estado



C.2.4 DIALOGO DE RESULTADOS

Es donde se despliega la información final de la aplicación FENIX. Este dialogo se divide en cuatro zonas como se muestra en la figura C.25.

Figura C.25 Dialogo de resultados



Zona 1. Información General: Muestra la información del algoritmo utilizado, cantidad de objetos de interés en la escena y unidad de medida para la posición.

Zona 2. Pista: área en la cual se simula la escena real. Los objetos de interés en la escena se simbolizan por medio de marcas.

Zona 3. Estado: muestra el estado de ejecución de los algoritmos (procesando / detenido).

Zona 4. Localización: se muestra la información de posición en la unidad de medida seleccionada, para los objetos móviles y obstáculos en la escena. Además el ángulo de orientación respecto al eje x para cada uno de los objetos móviles.