

Localización Dinámica de Móviles y Obstáculos para Aplicaciones en Robótica

Diego Alexander Tibaduiza Burgos ^{*†} Roberto Martínez Ángel^{*} Yamit D. Amaya Quintero^{*}, John Ruiz Hernández^{*}

Resumen

Usando algoritmos de procesamiento de imágenes y detección de movimientos se logró obtener la posición de dos móviles y de dos obstáculos fijos en un escenario en el cual se controlan las condiciones de color de los mismos. Para tal fin, se implementaron tres algoritmos, el primero de ellos se basa en los gradientes espacial y temporal para realizar la detección de los bordes de los objetos en movimiento dentro de la escena, el segundo es una detección en color, con la cual se identifica el color de los móviles y el tercero se basa en correlación de imágenes para realizar la búsqueda del objeto en movimiento. La detección de los obstáculos (objetos estáticos) para los tres algoritmos se realizó por medio de reconocimiento de un color característico en los obstáculos. Después de detectados los objetos de interés en la escena, se realiza su localización en la imagen por medio del centroide, para posteriormente entregar su posición y orientación en unidades de longitud.

Palabras Clave: *Análisis dinámico de imágenes, color, correlación, gradientes, identificación, segmentación.*

Abstract

Using algorithms of processing of images and detection of movements was possible to obtain the position of two mobile objects and two fixed obstacles in a scene where they interact under controlled conditions of color. For such an end, three algorithms were implemented, the first of them is based on the spatial and temporal gradients to carry out the detection of the borders of the objects in movement inside the scene, the second is a detection in color, with which the color of the mobile objects is identified and the third is based on correlation of images to accomplish the search of the object in movement. The detection of the obstacles (static objects) for the three algorithms was carried out through recognition of a characteristic color in the obstacles. After having detected the objects of interest in the scene, is carried out their localization in the image through the centroide (center), for later on to give their position and orientation in centimeters.

Keywords: *Dynamic analysis of images, color, correlation, gradients, identification, segmentation.*

1 Introducción

Al aplicar algoritmos de procesamiento digital de imágenes es posible obtener características ocultas en la adquisición que permiten al usuario tener una mayor información acerca de lo que se está monitorizando. “Localización Dinamica de Móviles y Obstáculos en una Escena Controlada

[†] Docente investigador. Universidad Autónoma de Bucaramanga, Calle 48 No. 39-234, Bucaramanga, Colombia. dtibaduiza@unab.edu.co

^{*} Universidad Industrial de Santander, Cra 27 Calle 9, Bucaramanga, Colombia. rmangel@uisedu.co

Para Aplicaciones en Robótica”, es un trabajo desarrollado con el fin de avanzar en la investigación del análisis dinámico de imágenes para lograr una interacción más eficiente entre el movimiento de los objetos en una escena y su localización por medio de un computador, lo cual adquiere importancia en aplicaciones relacionadas con robótica móvil y planeación de trayectorias.

En la actualidad competencias como la Robocopa, entre otras, han incentivado el desarrollo de estrategias como las planteadas en este artículo para poder determinar posición y orientación de objetos en una escena controlada para obtener resultados en tiempos mucho más rápidos sin dejar a un lado la robustez propia de un procesamiento digital de imágenes.

Dentro de esta competencia se reglamenta el color de cada objeto dentro de la pista para que cada equipo desarrolle algoritmos de color que permitan conocer las posiciones de cada objeto dentro de la pista. Siguiendo estas particularidades, se implementaron tres algoritmos, el primero de ellos se basa en los gradientes espacial y temporal para realizar la detección de los bordes de los objetos en movimiento dentro de la escena, en la que se encuentran obstáculos móviles y fijos con unas características definidas de color. El segundo es una detección en color, con la cual se identifica el color de los móviles y el tercero se basa en correlación de imágenes para realizar la búsqueda del objeto en movimiento, se presentará una breve descripción de cada uno y se mostrarán los resultados obtenidos con el fin de determinar cual es el mas apropiado para la situación planteada. Este trabajo fue implementado en un sistema robótico para detectar dos robots móviles de color amarillo y dos obstáculos fijos de color azul. Uno de los móviles se uso para realizar planeamiento de trayectorias, por lo cual el algoritmo de visión es usado para indicarle al algoritmo de planeamiento las posiciones de cada objeto en la escena en tiempo real y evitar colisiones.

2 Generalidades del Análisis Dinámico de Imágenes

2.1 Definición

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.

Al ser parte de la visión computacional, el análisis dinámico de imágenes se realiza siguiendo las mismas etapas de la visión computacional, en las que aplicando procesamiento digital de imágenes se extrae la información necesaria para cumplir con la tarea de localización y seguimiento. Un esquema general puede ser apreciado en la Fig. 1.

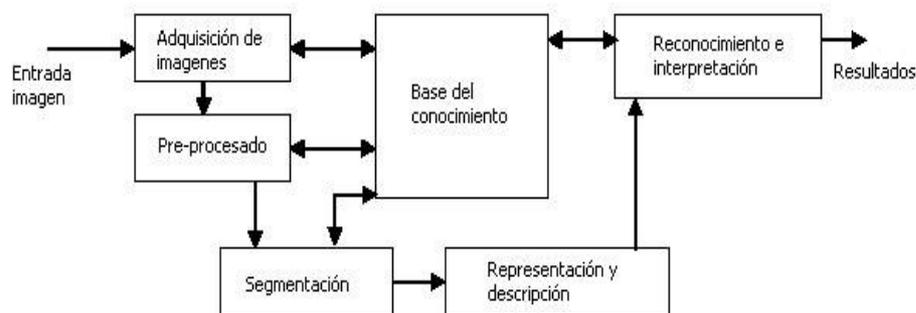


Fig. 1. Etapas de un sistema de visión computacional

2.2. Etapas de un Sistema de Visión Computacional

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 haciendo así innecesario el digitalizador.

Pre-procesamiento. 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.

Segmentación. En esta etapa los objetos de interés en la imagen son separados del fondo para su posterior estudio.

Representación y Descripción. La representación es el proceso mediante el cual se obtienen características convenientes para representar un objeto segmentado en la escena.

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.

3 Etapas Fundamentales de la Aplicación Implementada

Las etapas implementadas en este trabajo, para la localización dinámica de objetos en movimiento se observan en la Fig. 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.

3.1 Adquisición y Pre-procesamiento

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.

3.1.1 Descripción de la Escena

El escenario en general (Fig. 3.) consta de una pista de madera, con unas medidas de 2.44 m * 1.52 m y 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 iluminación de la escena se realiza con lámparas fluorescentes existentes en el laboratorio, por tanto este factor no está controlado.

La cámara se ubicó a una altura de 2.80 m de altura de la pista, procurando que su eje óptico quedara perpendicular al piso, de esta forma se obtiene un área efectiva de captura de 2 m².

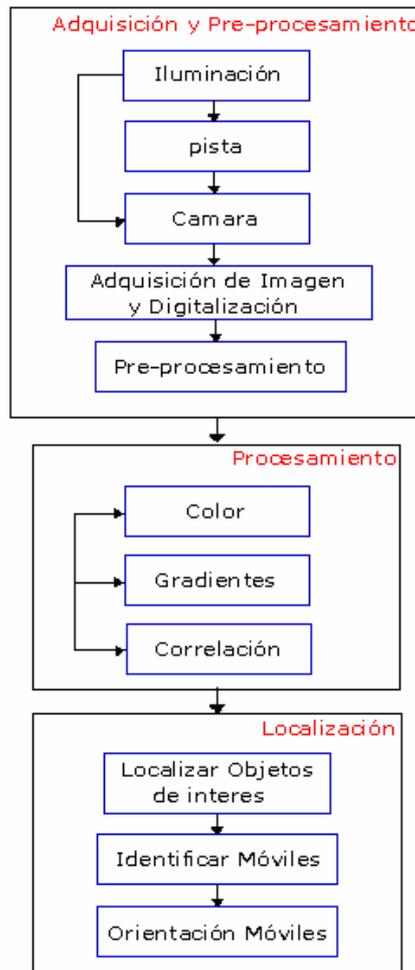


Fig. 2. Etapas fundamentales de la aplicación

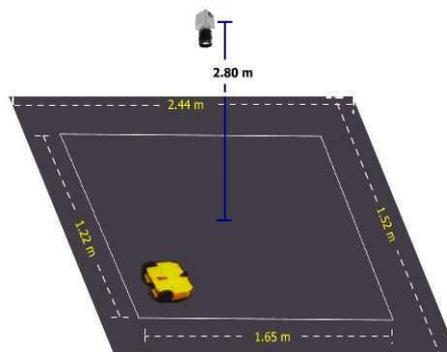


Fig. 3. Distribución general de la escena

3.1.2 Elementos de Adquisición

Para la adquisición de las imágenes se utilizó una cámara analógica DFK 50H13 con un lente de 8mm y una tarjeta de compresión de video PCI de 32 bits, DFG/Compress PIRANHA. Con esto se logra trabajar imágenes de 640 x 480 pixeles.

3.1.3 Algoritmo de Adquisición Implementado

El algoritmo de adquisición 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. 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.

3.1.4 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, los cuales eliminan en gran parte el ruido presente en la imagen sin alterar de forma notoria la calidad visual de la misma.

3.2 Detección y Segmentación de Movimiento

En esta sección, se muestran los diferentes algoritmos implementados para la detección y segmentación del movimiento.

3.2.1 Algoritmo 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. Se implementaron los modelos de color xyY [5] y HLS [6], estos modelos se muestran en la Fig. 4, la gráfica a corresponde al modelo de color xyY, y la grafica b corresponde al modelo HLS.

La segmentación de color se selecciona indicando el área sobre la grafica en el programa realizado, los objetos móviles trabajados son de color amarillo y los obstáculos de color azul. Esta umbralización de color se realiza para toda la imagen.

El diagrama de flujo para el algoritmo basado en color se muestra en la Fig 5.

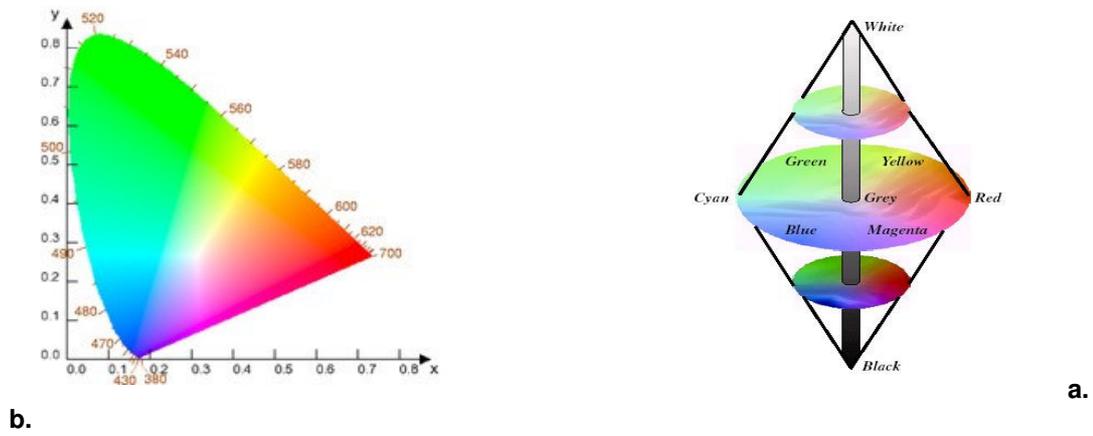


Fig. 4. Esquemas de los algoritmos de color usados

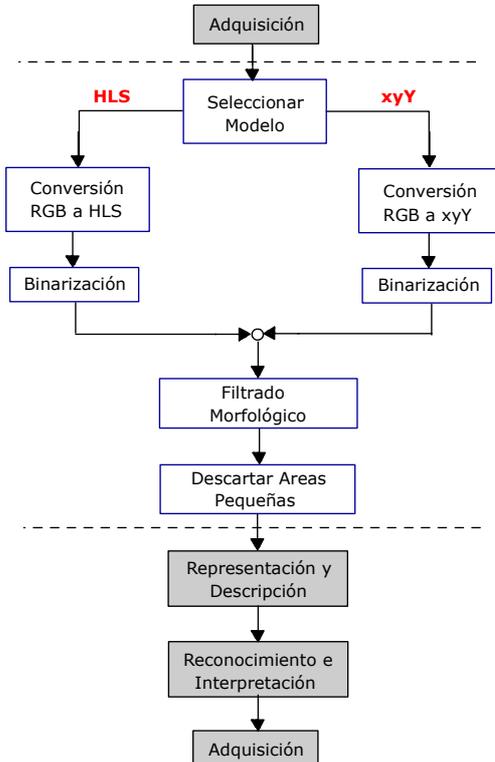


Fig. 5. Algoritmo basado en color

En la Fig. 6 y 7 se observa el resultado de aplicar el algoritmo de color a algunas imágenes tomadas en el laboratorio. Para cada modelo se segmenta un objeto de diferente color.

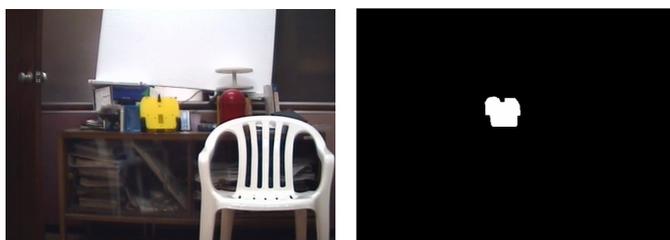


Fig. 6. Segmentación del objeto amarillo mediante el modelo de color HLS

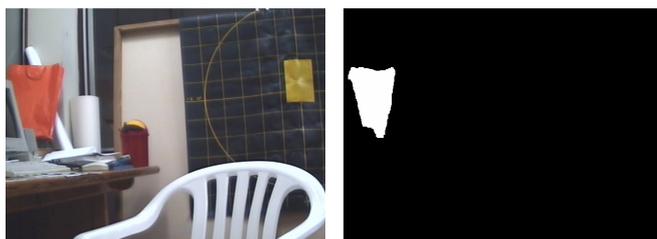


Fig. 7. Segmentación del objeto naranja mediante el modelo de color xyY

3.2.2 Algoritmo 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 [3] el cual soluciona en gran medida este problema.

Para el algoritmo de Jain 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.

Luego se aplica el algoritmo de Jain, el cual consiste inicialmente en hacer una detección de bordes a cada imagen por medio del detector de bordes de Sobel [7]. Esta detección de bordes permite hallar el gradiente de cada una de las imágenes, es necesario tener en cuenta que al realizar el gradiente se obtiene una imagen de magnitud que corresponde a la misma imagen de bordes con la que se trabaja el algoritmo y en la cual los valores serán todos positivos ya que son magnitudes. Después se realiza la diferencia con valor absoluto 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. 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. En la Fig. 8. se muestra el algoritmo basado en gradientes que se implementó.

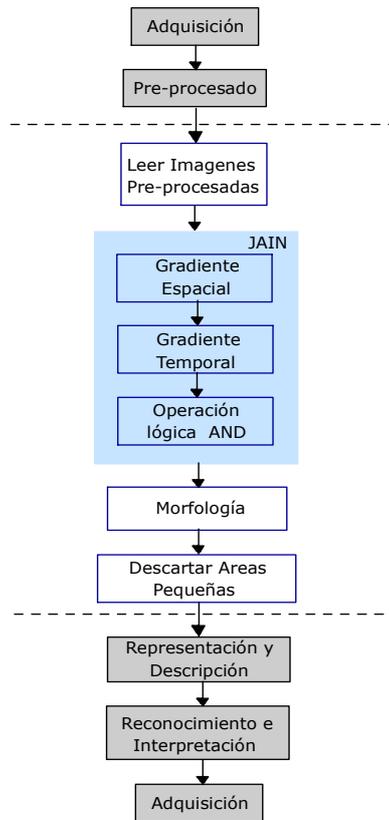


Fig. 8. Algoritmo basado en gradientes

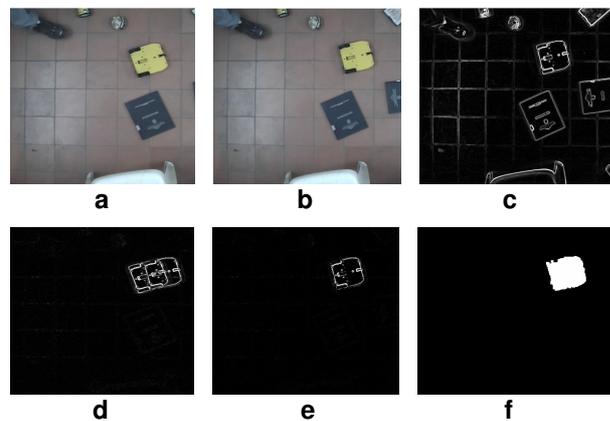


Fig. 9. Segmentación del objeto en movimiento con el algoritmo basado en gradientes

En la Fig. 9. 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 Fig. 9. se muestra el resultado de las operaciones del algoritmo de Jain. Finalmente en la Fig. 10-f. se observa el resultado de aplicar operaciones morfológicas a la imagen 9-e obteniendo así la región interna del objeto para su posterior localización.

Las operaciones morfológicas son las siguientes:

Primero se hacen dilataciones con el fin de acentuar los bordes débiles que han resultado del gradiente temporal y obtener el contorno completo del objeto deseado, luego se hace el relleno del objeto para poder localizar su centroide posteriormente, finalmente se realizan erosiones con el fin de eliminar pequeños puntos que pudieron formar o quedaron debido a las dilataciones hechas inicialmente.

El elemento estructurante tanto para las dilataciones como para las erosiones tiene un tamaño de 3x3 y todos sus valores son 1, hablando en binario.

Antes de realizar las operaciones morfológicas se realiza una binarización de la imagen con el fin que este proceso sea más eficiente, computacionalmente hablando.

3.2.3 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 está 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 Fig. 10.

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 ratón 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.

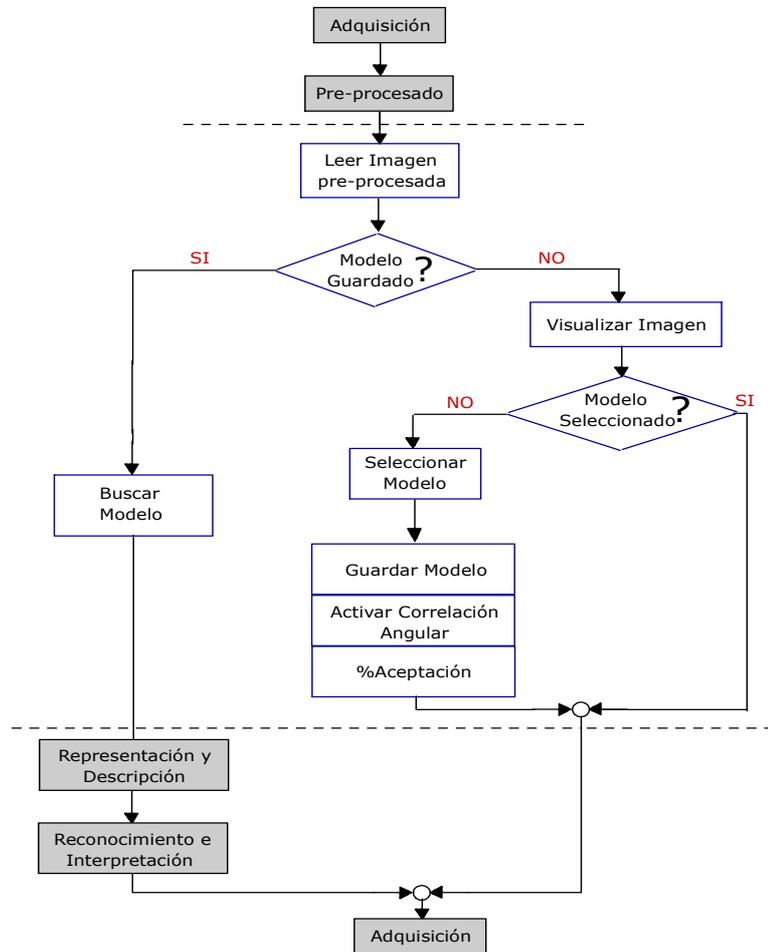


Fig. 10. Algoritmo basado en correlación

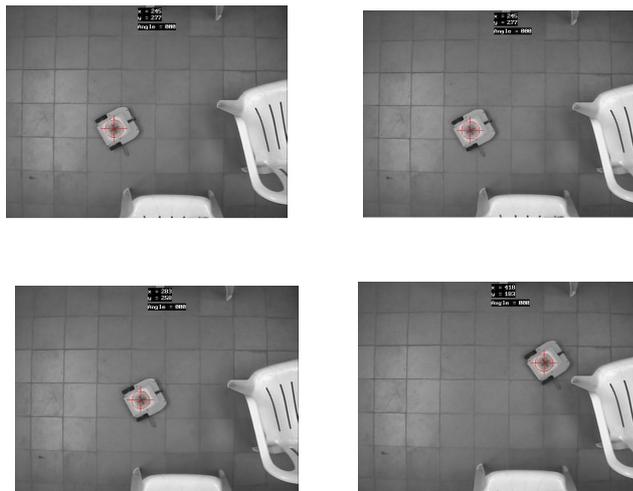


Fig. 11. Seguimiento del objeto en movimiento por medio del algoritmo basado en correlación

En las fotos de la Fig 11, 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 Identificación

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 [2] del objeto segmentado con respecto al origen de coordenadas de la imagen. Aunque el centroide físico está relacionado con una propiedad física de los objetos como es su masa, tiene una aplicación particular en el procesamiento digital de imágenes considerando cada pixel como una masa puntual ficticia cuyo valor fuera el nivel de gris en dicho punto.

4.2 Calibración de la Cámara

Para la calibración de la cámara DFK50H13/N utilizada en el presente proyecto se empleó la toolbox de Matlab Camera Calibration Toolbox, 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

Coefficientes de distorsión radial

$K_{c1} = -0.31630$

$K_{c2} = -0.31778$

$K_{c6} = 0$

Coefficientes de distorsión tangencial

$K_{c3} = 0.00186$

$K_{c4} = 0.00094$

En la Fig. 12. se observa el modelo completo de distorsión generado por la toolbox de calibración para la cámara utilizada en el proyecto

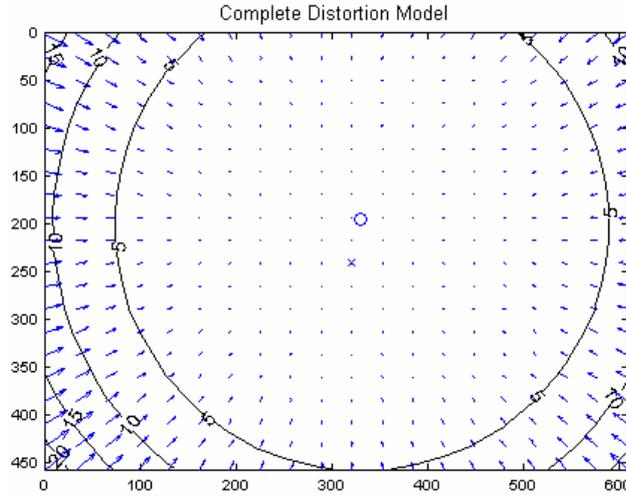


Fig. 12. Modelo de distorsión completa

4.3 Corrección de Distorsiones

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. La descripción matemática se muestra a continuación

$$P_U = P_X f_{CX} + C_{OX} \quad (1)$$

$$P_V = P_Y f_{CY} + C_{OY} \quad (2)$$

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 obtenidas en la calibración.

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

$$P_{dX} = P_{iX} \left(1 + K_{C1}r^2 + K_{C2}r^4 + K_{C5}r^6 \right) + 2K_{C3}P_{iX}P_{iY} + 2K_{C4}P_{iX}^2 + K_{C4}r^2 \quad (3)$$

$$P_{dY} = P_{iY} \left(1 + K_{C1}r^2 + K_{C2}r^4 + K_{C5}r^6 \right) + 2K_{C4}P_{iX}P_{iY} + 2K_{C3}P_{iY}^2 + K_{C3}r^2 \quad (4)$$

$$r = \sqrt{P_{iX}^2 + P_{iY}^2} \quad (5)$$

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. El proceso de obtención de las coordenadas sin distorsiones se realiza en tiempo real.

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 usó un factor de escala encontrado bajo las condiciones del escenario descrito anteriormente. Este factor de escala es:

$$fe = \frac{10 \text{ Centímetros}}{39 \text{ pixel}} \quad (6)$$

Para encontrar este factor se ubicó un cuadrado de 10 x 10 cm en la pista del escenario; luego se realizó 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

La etapa de identificación consta de tres fases fundamentales: actualizar móviles detectados, eliminar móviles no detectados y detectar móviles nuevos.

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 más cercano posible sin haber chocado, sus centros de gravedad se encuentran a una distancia de 20cm aproximadamente (Fig. 13.).

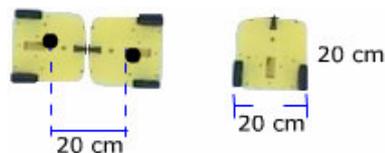


Fig. 13. Dimensiones de los móviles

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. (Fig. 14.).

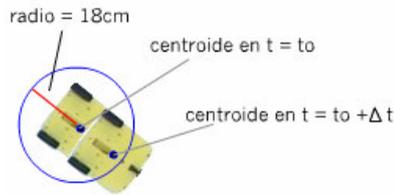


Fig. 14. Identificación de un móvil

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:

$$v_{\max} = \frac{x_{\max}}{\Delta t} \quad (7)$$

$$v_{\max} = \frac{18 \left[\frac{cm}{s} \right]}{\Delta t} \quad (8)$$

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 logra 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 alterno se muestra en la Fig. 15.

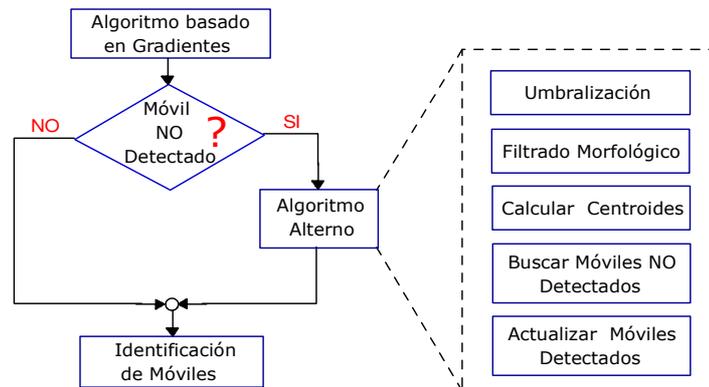


Fig. 15 Diagrama par el algoritmo alternativo

El algoritmo alternativo mostrado en la Fig. 15, 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 cálculo 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

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 el sistema de referencia de la Fig. 16, el ángulo para un móvil aumenta en el sentido de las manecillas del reloj respecto al eje positivo x, desde de 00 hasta 360o. Este ángulo se puede calcular por medio de la relación mostrada a continuación:

$$\theta = \tan^{-1} \left(\frac{ym - yo}{xm - xo} \right) \text{ [radianes]} \quad (9)$$

$$\theta = \frac{180}{\pi} \theta \text{ [grados]} \quad (10)$$

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.

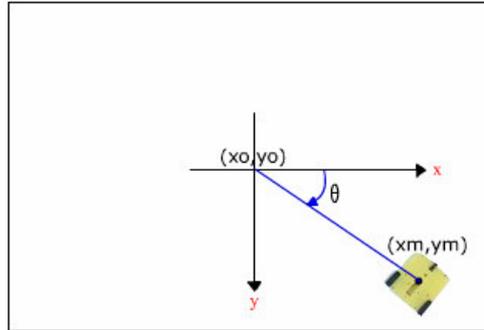


Fig. 16. Sistema de referencia fijado en el centro de la escena

4.6.2 Sistema de Referencia Variable

La referencia se fija en el centroide del objeto, orientada paralelamente a los ejes (x,y) y cambia en cada instante conforme éste se mueve. Este sistema se muestra en la Fig. 17.

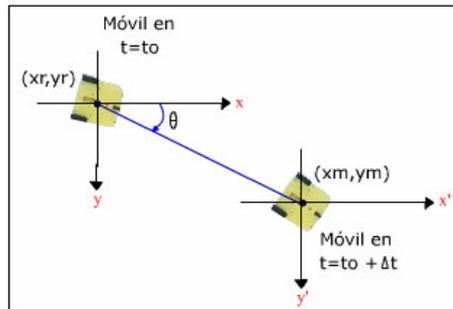


Fig. 17. Sistema de referencia variable

En la Fig. 17. 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 siguiente ecuación:

$$\theta = \tan^{-1} \left(\frac{y_m - y_r}{x_m - x_r} \right) \text{ [radianes]} \quad (11)$$

$$\theta = \frac{180}{\pi} \theta \text{ [grados]} \quad (12)$$

5 Pruebas y Resultados

Las pruebas realizadas a los algoritmos implementados fueron:

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

Para realizar estas pruebas se utilizó una aplicación implementada en visual C++ 6.0 (FENIX), la cual genera un archivo de texto con la información de posición y orientación de los objetos de interés.

5.1 Posición

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 Fig. 18.

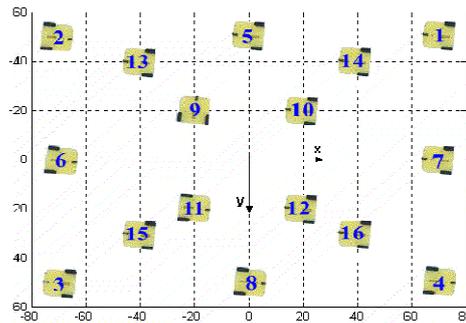


Fig. 18. Posiciones usadas para determinar la distribución del error

En la Fig. 19 y 20 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.

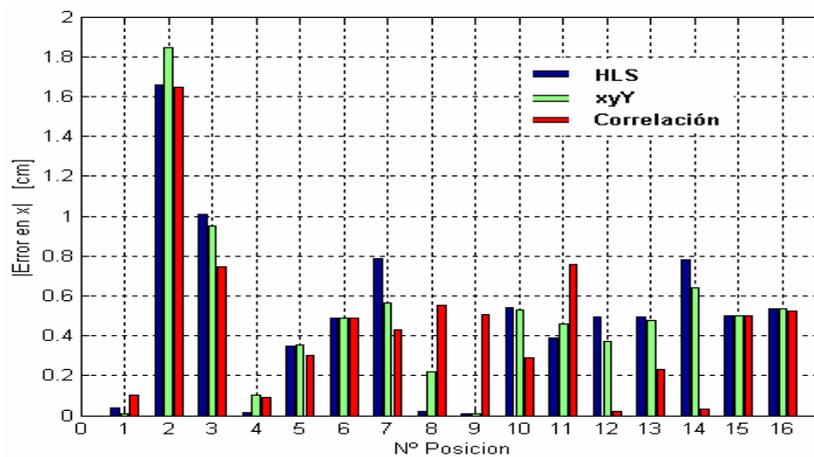


Fig. 19. Distribución comparativa del error de posición utilizando los algoritmos implementados para la coordenada en x

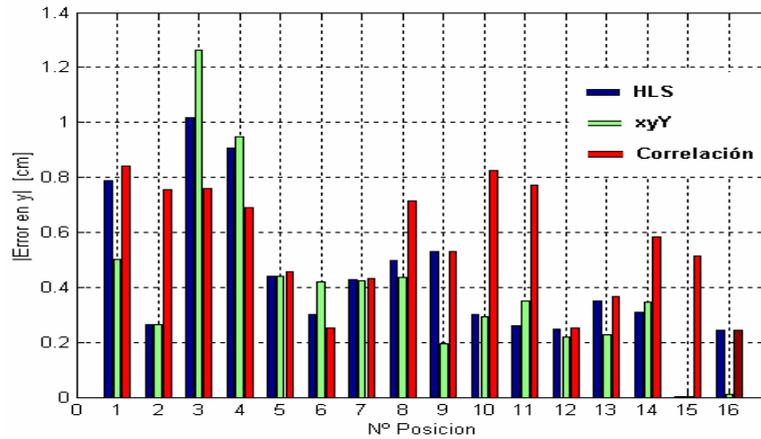


Fig. 20. Distribución comparativa del error de posición utilizando los algoritmos implementados para la coordenada en y

Como se observa en la Fig. 19 y 20 la aplicación desarrollada arrojó resultados con un error de posición y de orientación inferiores a 1 cm y 2° respectivamente.

5.2 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°.

En la Fig. 21 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.

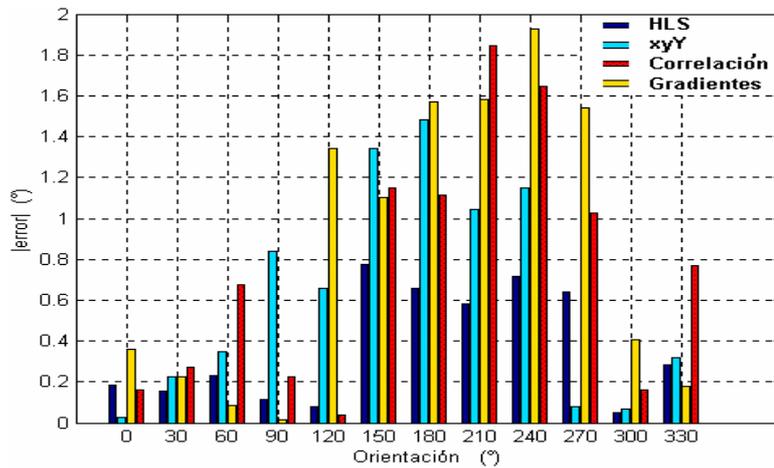


Fig. 21. Sistema polar para la prueba de orientación

5.3 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 Fig. 22. en color azul se observa la trayectoria ideal y en color rojo la trayectoria calculada con cada uno de los algoritmos implementados.

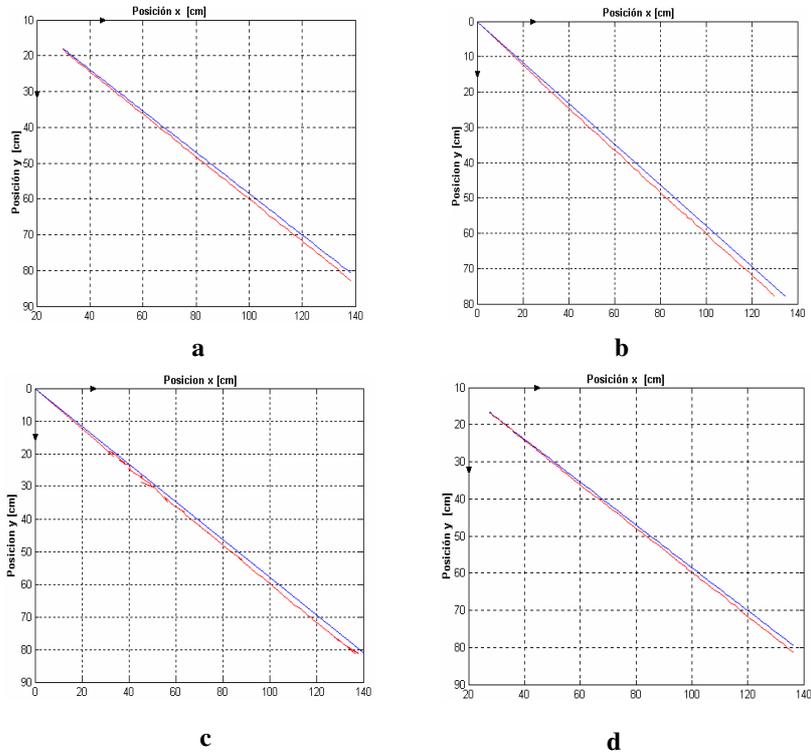


Fig. 22. Seguimiento de una trayectoria recta a) con el modelo de color HLS b) con el modelo de color xyY c) con el algoritmo basado en gradientes d) con el algoritmo basado en correlación

La segunda trayectoria descrita es circular. Los resultados para el seguimiento con los algoritmos implementados se observan en la Fig. 23.

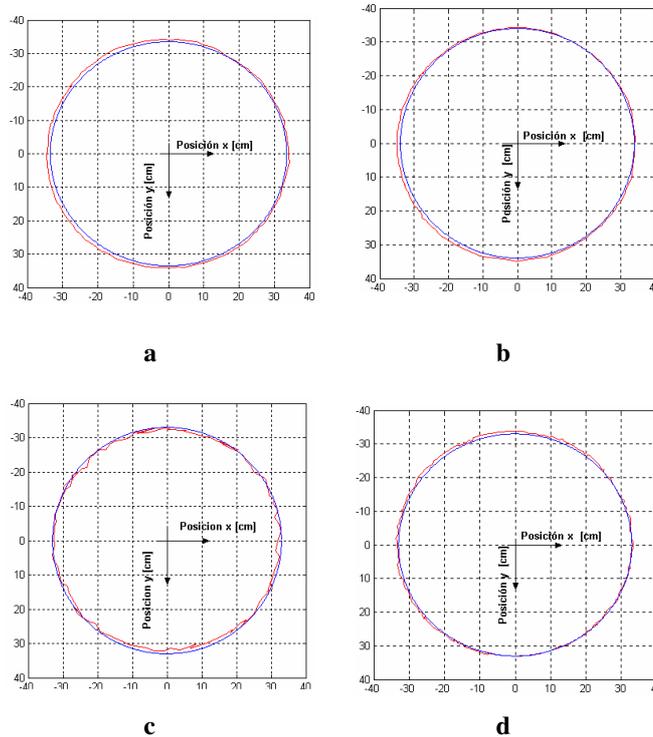


Fig. 23. Seguimiento de una trayectoria circular a) con el modelo de color HLS b) con el modelo de color xyY c) con el algoritmo basado en gradientes d) con correlación

5.4 Tiempo de latencia

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

En la tabla 1 se muestran los tiempos de adquisición, procesamiento y el tiempo total de latencia el cual es la suma de los dos anteriores, estos datos están dados en segundos

Tabla 1. Tiempos de latencia para los algoritmos implementados

Método	Correlación		Gradientes	
	xyY	HLS	xyY	HLS
Adquisición	0,0735	0,0735	0,0735	0,0735
Procesamiento	0.8108	0.3018	0.7742	0.3000
Total	0,8821	0,3748	0,8515	0,3712

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

Sistema Operativo: Microsoft Windows XP Profesional versión 2002.

Procesador: Pentium IV de 1.5 GHz.

Memoria RAM: 256 MB

6 Conclusiones

El resultado final 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.

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 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.

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 unas futuras investigaciones sobre el flujo óptico [12] con el cual se pueden realizar mediciones mas precisas sobre el movimiento tales como velocidad y aceleración.

Las pruebas realizadas a la aplicación desarrollada 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.

Referencias

- [1] CHAPRA, Steven y CANALE, Raymond. Métodos Numéricos para ingenieros. México: McGraw-Hill interamericana editores S.A .1999
- [2] GONZÁLEZ, Rafael y WOODS, Richard. Tratamiento digital de imágenes. Estados Unidos de América: Editorial Addison-Wesley iberoamericana.1996
- [3] JAIN, Ramesh; RANGACHAR, Kasturi y BRIAN, Shunck. Machine Vision. Singapore: McGraw-Hill Book Co.,1995.
- [4] KRUGLINSKI, David J; SHEPHERD, George y WINGO, Scot. Programación Avanzada Con Visual C++. España: McGRAW HILL / INTERAMERICANA DE ESPAÑA. Primera edición en español, 1999.
- [5] 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/>>

- [6] PAJARES, Gonzalo y DE LA CRUZ, Jesús. *Visión Por Computador: Imágenes Digitales Y Aplicaciones*. México: Alfaomega, 2002.
- [7] RUSS John. *The Image Processing Handbook*. Boca Raton, FL. CRC PRESS, IEEE PRESS. 2002.