

UNIVERSIDAD CENTRAL DE VENEZUELA  
FACULTAD DE CIENCIAS  
ESCUELA DE COMPUTACIÓN  
CENTRO DE COMPUTACIÓN PARALELA Y DISTRIBUIDA

**Pizarra Interactiva de bajo costo  
utilizando WebCams**



Trabajo Especial de Grado  
Presentado ante la Ilustre  
**Universidad Central de Venezuela**

Por los Bachilleres:

**Cabrera C., Dayana M.**

**Romero A., Daniel E.**

Para optar por el título de  
Licenciado en Computación

Tutor:

**Prof. Robinson Rivas**

Caracas, Noviembre 2010



---

## Resumen

En la actualidad existen diversas interfaces que facilitan el uso del computador, entre ellas se encuentran las pizarras interactivas, las cuales son de gran utilidad en ambientes educativos y de conferencias. A pesar de que existe una gran variedad en el mercado, el uso de éstas no se ha masificado debido a su alto costo. Es por ello que en este Trabajo Especial de Grado se presenta un sistema enfocado en ofrecer las características de una pizarra interactiva utilizando webcams, materiales de bajo costo y los conocimientos que ofrece el área de visión por computador. El sistema desarrollado captura las imágenes, les aplica el procesamiento necesario y finalmente realiza la acción correspondiente en el computador.

### **Palabras Clave:**

Pizarra interactiva, visión por computador, procesamiento digital de imágenes, Webcam.



---

## Agradecimientos

A nuestros familiares, que nos apoyaron en todo momento.

A nuestro tutor, el Profesor Robinson Rivas, por aclarar nuestras dudas y estar dispuesto a ayudarnos cada vez que fuese necesario.

Al personal del Centro de Computación Gráfica (CCG), por contar con su ayuda y buena disposición.

A los jurados por toda la disponibilidad y ayuda impartida.



---

## Acta

Quienes suscriben, miembros del jurado designado por el Consejo de Escuela de Computación de la Facultad de Ciencias, para examinar el Trabajo Especial de Grado presentado por los Bachilleres Dayana María Cabrera Castillo, C.I. 18.466.335 y Daniel Eduardo Romero Arias, C.I. 16.903.158, con el título: **“Pizarra Interactiva de bajo costo utilizando WebCams”**, a los fines de optar al título de Licenciado en Computación, dejan constancia de lo siguiente:

Leído como fue, dicho trabajo por cada uno de los miembros del jurado, se fijó el día 1 de Noviembre de 2010, a las 12:00pm para que sus autores lo defendieran en forma pública, lo que se hizo en la Sala Leandro Aristigueta, de la Facultad de Ciencias. Luego de la presentación dieron respuesta a las preguntas formuladas. Finalizada la defensa pública del Trabajo Especial de Grado, el jurado decidió aprobarlo.

En fe de la cual se levanta la siguiente Acta en Caracas el primer día del mes de Noviembre del año 2010, dejando constancia de que actuó como Coordinador del Jurado el Profesor Tutor Robinson Rivas.

Prof. Robinson Rivas  
Tutor

Prof. Rhadamés Carmona  
Jurado

Prof. Ernesto Coto  
Jurado Suplente



---

# Índice general

<b>1. Planteamiento del Problema</b> .....	15
1.1. Objetivo General .....	15
1.2. Objetivos específicos .....	15
<b>2. Visión por Computador</b> .....	17
2.1. Procesamiento Digital de Imágenes .....	17
2.1.1. Imagen Digital .....	18
2.1.2. Operaciones .....	19
2.1.3. Filtrado .....	24
2.2. Sistemas de Visión por Computador .....	29
2.2.1. Captura .....	30
2.2.2. Preprocesamiento .....	32
2.2.3. Segmentación .....	33
2.2.4. Reconocimiento .....	33
2.2.5. Calibración de la cámara .....	33
2.2.6. Visión Estéreo .....	35
2.2.7. Triangulación Geométrica .....	36
2.3. Detección de movimiento .....	37
2.3.1. Flujo óptico .....	38
2.3.2. Análisis del movimiento basado en la correspondencia .....	40
2.3.3. Imagen Diferencial .....	40
2.3.4. Imágenes de diferencias acumuladas .....	41
2.3.5. Sustracción de fondo .....	41

2.4. Interpolación . . . . .	42
2.4.1. Interpolacion Lineal . . . . .	43
<b>3. Pizarras Interactivas . . . . .</b>	<b>45</b>
3.1. Evolución de las Pizarras . . . . .	45
3.2. Pizarra Interactiva . . . . .	47
3.3. Clasificación . . . . .	48
3.3.1. Proyección . . . . .	48
3.3.2. Tecnologías . . . . .	49
3.4. Antecedentes . . . . .	51
3.4.1. LoCoBoard . . . . .	52
3.4.2. WiiBoard . . . . .	56
<b>4. Diseño y desarrollo de la solución propuesta . . . . .</b>	<b>61</b>
4.1. Plataforma de Hardware . . . . .	61
4.1.1. Diseño del Prototipo . . . . .	61
4.2. Plataforma de Software . . . . .	64
4.2.1. Lenguaje de programación . . . . .	65
4.2.2. Librerías de desarrollo . . . . .	65
4.3. Diseño de la aplicación . . . . .	66
4.4. Ejecución de la aplicación . . . . .	67
4.4.1. Calibración de la pizarra . . . . .	67
4.4.2. Captura . . . . .	71
4.4.3. Preprocesamiento . . . . .	71
4.4.4. Segmentación . . . . .	71
4.4.5. Reconocimiento . . . . .	73
4.4.6. Intérprete . . . . .	75
<b>5. Pruebas y resultados . . . . .</b>	<b>77</b>
5.1. Pruebas de desarrollo . . . . .	77
5.1.1. Ángulo de visión de la webcam . . . . .	77
5.1.2. Calibración de la cámara . . . . .	78

	Índice general	11
5.1.3. Técnicas de detección de movimiento		85
5.1.4. Iluminación y contraste		86
5.1.5. Iluminación y contraste en movimiento		90
5.1.6. Detección de color		97
5.2. Pruebas finales		98
5.2.1. Rendimiento del sistema		98
5.2.2. Precisión del sistema		99
<b>Referencias</b>		<b>107</b>



---

## Introducción

Los sistemas interactivos son aquellos que se interrelacionan y dependen de las acciones de un usuario para realizar una tarea. Estos sistemas son ampliamente aplicados a diversas áreas como: robótica, realidad virtual, sistemas evolutivos, simuladores, multimedia, sistemas de comunicación, control distribuido, etc. Y cada día son más la cantidad de dispositivos que nos encontramos en la vida diaria, por ejemplo los sistemas de kioscos interactivos usados para dar información.

En la actualidad la interactividad es una de las áreas de mayor desarrollo a nivel mundial, y cada vez son más los sistemas interactivos que aparecen en el mercado. Investigaciones recientes, han mostrado el impacto que existe en la aplicación de pizarras interactivas en el mundo de la educación, estas pizarras tienen muchas ventajas sobre las pizarras tradicionales, ya que además de todas las posibilidades que tienen al permitir el manejo del computador, poseen funcionalidades adicionales en el software que permiten grabar, imprimir la información, distribuirla a través de una red, entre otras funciones. Sin embargo, una de las principales desventajas que tienen todas las pizarras presentes en el mercado es su alto costo.

En el presente trabajo se muestra el diseño e implementación de un prototipo de pizarra interactiva de bajo costo. Consta de dos cámaras web, que captan una serie de imágenes que son tratadas utilizando técnicas de procesamiento digital de imágenes, para generar datos que son interpretados y traducidos como una acción del computador. En los siguientes capítulos se describen las bases teóricas necesarias para desarrollar el sistema, el

hardware utilizado, descripción del software creado, las pruebas realizadas y los resultados obtenidos.

## Planteamiento del Problema

Las pizarras interactivas que existen en el mercado actualmente, a pesar de ofrecer diversas funcionalidades útiles para la enseñanza y para conferencias, son delicadas y tienen un costo muy elevado. Además muchas de ellas requieren dispositivos como lápices o borradores especiales, que en caso de dañarse son difíciles de remplazar. Otra desventaja es que la mayoría de estas pizarras sólo funcionan bajo Windows.

Por estas razones han surgido alternativas que tratan de solucionar el problema del costo. Sin embargo, estas alternativas siguen teniendo ciertas desventajas, como la necesidad de depender de un lápiz infrarrojo y existe la posibilidad que el usuario obstruya la luz infrarroja al dispositivo de captura. Además sólo ofrecen las funcionalidades básicas de interacción con el computador.

Es por esto que surge la necesidad de crear un sistema que provea mayor funcionalidad que las pizarras de bajo costo existentes y que sea independiente de dispositivos electrónicos para la interacción.

### 1.1. Objetivo General

Crear un sistema de pizarra interactiva de bajo costo que permita el manejo del computador, utilizando webcams, un video beam y una pizarra acrílica.

### 1.2. Objetivos específicos

- Diseñar una interfaz gráfica que permita la configuración del sistema, y acceder a las funcionalidades básicas de la pizarra.

- Permitir al usuario utilizar la pizarra para mover el apuntador del ratón y hacer click.
- Implementar un método para la calibración de las cámaras.
- Detectar mediante técnicas de procesamiento de imágenes la ubicación del objeto utilizado como apuntador y transformar esta ubicación a coordenadas de proyección.
- Elaborar un prototipo que pueda ser utilizado en un ambiente educativo.

## Visión por Computador

La visión por computador es el proceso que permite tomar imágenes del mundo real para extraer de ellas información necesaria para una aplicación específica [1]. Este proceso involucra la **captura** de imágenes mediante dispositivos como cámaras digitales, webcams y escáneres que permiten que sean procesadas; en la etapa de **preprocesamiento** se aplican técnicas para mejorar la imagen, de forma que facilite la extracción de características, que es conocida como **segmentación**; finalmente en la etapa de **reconocimiento** se toman los resultados de las etapas anteriores para realizar el análisis que permita tomar decisiones para obtener el resultado final. Todas estas técnicas van a ser desarrolladas a lo largo de este capítulo.

### 2.1. Procesamiento Digital de Imágenes

El análisis de imágenes digitales es un factor clave para solucionar cualquier problema de tratamiento de imágenes. El análisis de imágenes involucra manipular los datos de esta y determinar exactamente la información requerida para desarrollar un sistema de tratamiento de imágenes. La solución al problema puede requerir el uso de hardware o software existente, o puede requerir el desarrollo de nuevos algoritmos y diseños de sistema.

El proceso de análisis de imagen ayuda a definir los requerimientos para que el sistema sea desarrollado, este análisis es parte de un proceso más grande, iterativo por naturaleza, que puede responder preguntas específicas de la aplicación.

El análisis de imagen es principalmente un proceso de reducción de datos, ya que las imágenes contienen enormes cantidades de datos. A menudo mucha de esta información

no es necesaria para resolver un problema específico, entonces una parte del proceso de análisis de imágenes es determinar exactamente qué información es necesaria. En muchas aplicaciones el factor determinante en la factibilidad del desarrollo del sistema es el resultado del análisis preliminar de la imagen[1].

### 2.1.1. Imagen Digital

El término imagen se refiere a una función bidimensional de intensidad de luz  $f(x,y)$  donde  $x$  e  $y$  representan las coordenadas espaciales y  $f$  es proporcional al brillo (o nivel de gris) de la imagen en ese punto. Cada punto de la imagen es conocido como píxel. El modelo de imagen anteriormente se muestra en la figura 2.1.

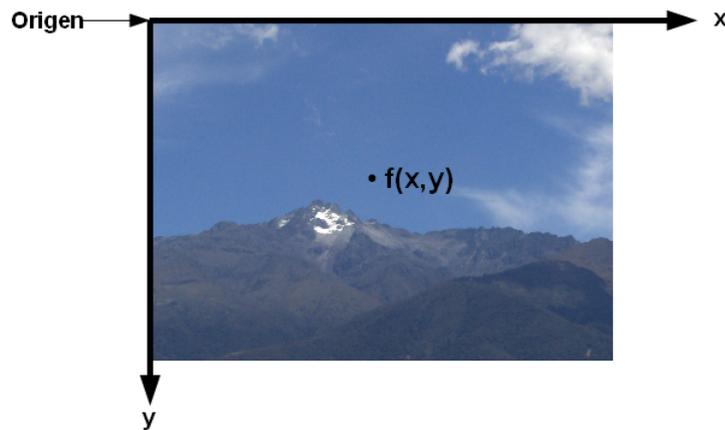


Figura 2.1. Representación de Imagen Digital.

Una imagen  $f(x,y)$  esta formada por dos componentes: una es la cantidad de luz incidente en la escena y la otra es la cantidad de luz reflejada por los objetos. Estos componentes son llamados: iluminación  $i(x,y)$  y reflectancia  $r(x,y)$  [4].

La iluminación  $i(x,y)$  está determinada por las características de la fuente que emite la luz y la reflectancia  $r(x,y)$  por las características del objeto. Por ejemplo se podrían tener los valores de  $i(x,y) = 100$  en una oficina de trabajo,  $i(x,y) = 0,01$  para una noche clara.  $r(x,y) = 0,01$  para el terciopelo negro y  $r(x,y) = 0,93$  para la nieve. Donde

$$0 < i(x, y) < \infty \text{ y } 0 < r(x, y) < 1.$$

Se puede ver el modelo de imagen como una matriz, donde cada fila o columna es un arreglo unidimensional. Éste modelo es utilizado para representar imágenes de escala de grises, pero existen otros tipos de imágenes que requieren modificaciones a este modelo. Típicamente estas son imágenes multibanda, y ellas pueden ser modeladas por diferentes funciones  $x, y$  correspondientes a la información del brillo de cada banda por separado.

Las imágenes digitales pueden ser binarias, de escala de grises y a color [1].

- **Imagen binaria:** La imagen binaria es el tipo más simple de imagen y puede tomar dos valores, típicamente blanco y negro (representados con uno y cero respectivamente).
- **Imagen en escala de gris:** Las imágenes en escala de grises solo contienen información de brillo, ninguna información de color. El número de bits usados por cada píxel determina el número de diferentes niveles de brillo disponibles, comúnmente estas imágenes contienen datos de 8 bits por píxel, lo cual permite tener 256 niveles de brillo.
- **Imagen a color:** Pueden ser modeladas como una imagen monocromática de tres bandas, donde cada banda de datos corresponde a un color diferente. La información almacenada en los datos de la imagen digital es el brillo en cada banda espectral. Comúnmente las imágenes a color son representadas como roja, verde y azul, o imágenes RGB. Usando el estándar monocromático como un modelo la imagen a color debería tener 24 bits por píxeles, 8 bits por cada banda de color (rojo, verde y azul).

### 2.1.2. Operaciones

El procesamiento de imágenes incluye la aplicación de diferentes técnicas y operaciones, éstas operaciones se clasifican dependiendo del tratamiento que se quiera hacer sobre la imagen, las principales técnicas se pueden dividir en: operaciones aritméticas, operaciones lógicas, operaciones de punto y operaciones geométricas, adicionalmente se pueden aplicar filtrado, detección de características y transformaciones sobre la imagen.

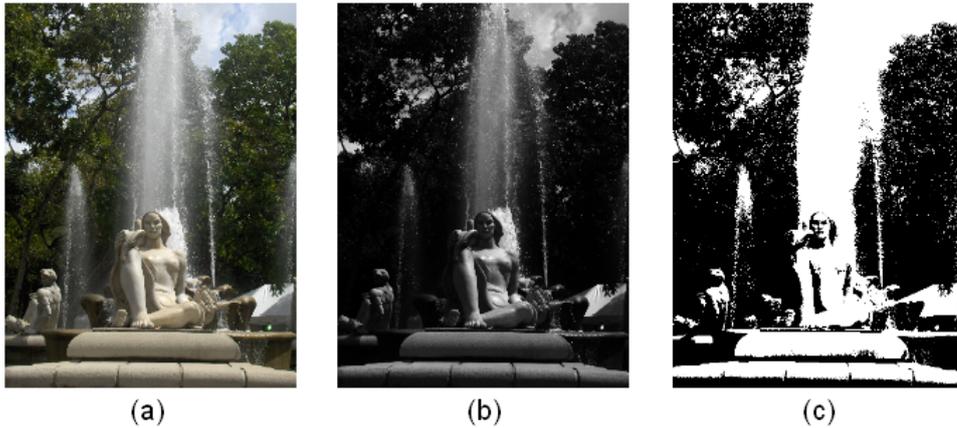


Figura 2.2. Tipos de Imágenes. (a) Imagen a color. (b) En escala de grises. (c) Monocromática

### Operaciones Aritméticas

Las operaciones aritméticas y lógicas entre píxeles se emplean en la mayor parte de las ramas del procesamiento de imágenes. Las operaciones aritméticas entre dos píxeles  $p$  y  $q$  son las siguientes [2]:

$$\text{Suma} = p + q$$

$$\text{Resta} = p - q$$

$$\text{Multiplicación} = \alpha * p$$

$$\text{División} = p/\alpha$$

Estas operaciones se realizan de píxel a píxel, donde  $\alpha$  es una constante que toma valores reales. La suma de imágenes, mostrada en la figura 2.3, se emplea para la eliminación de ruido, tomando el promedio de las imágenes. La resta, mostrada en la figura 2.4, se usa generalmente para eliminar información estática en la detección de movimiento, es utilizado con frecuencia en el procesamiento de imágenes médicas. Con las operaciones de multiplicación y división de imágenes básicamente se busca corregir las variaciones en los niveles de gris, esto ocurre cuando hay fallas en la iluminación o imperfecciones en los sensores que captan la imagen.

### Operaciones Lógicas

Dados los píxeles  $p$ ,  $q$ , se tiene las siguientes operaciones lógicas.



Figura 2.3. Suma de Imágenes.



Figura 2.4. Suma de Imágenes.

$$And = p \text{ AND } q$$

$$Or = p \text{ OR } q$$

$$Complemento = NOT \ q$$

Las operaciones lógicas sólo se pueden aplicar sobre imágenes binarias, mientras que las operaciones aritméticas se aplican a píxeles con varios valores. Las operaciones lógicas son herramientas básicas para el procesamiento de imágenes binarias, donde se emplean en tareas de enmascaramientos, detección de caracteres y análisis de forma [2].

### Operaciones puntuales

Estas operaciones tienden a mejorar el contraste en la imagen, se caracterizan porque los píxeles vecinos no tienen influencia en el resultado del nuevo valor de un píxel. Para explicar algunas de las operaciones de punto que se pueden aplicar a las imágenes digitales es importante conocer el significado de umbral. El umbral es un gráfico estadístico que permite representar la distribución de intensidad de los píxeles de una imagen. En la figura 2.5 se observa el histograma de una imagen en escala de grises.

Para representar el histograma de una imagen a color se suelen representar tres gráficos para representar las intensidades de cada una de las bandas RGB que componen a la

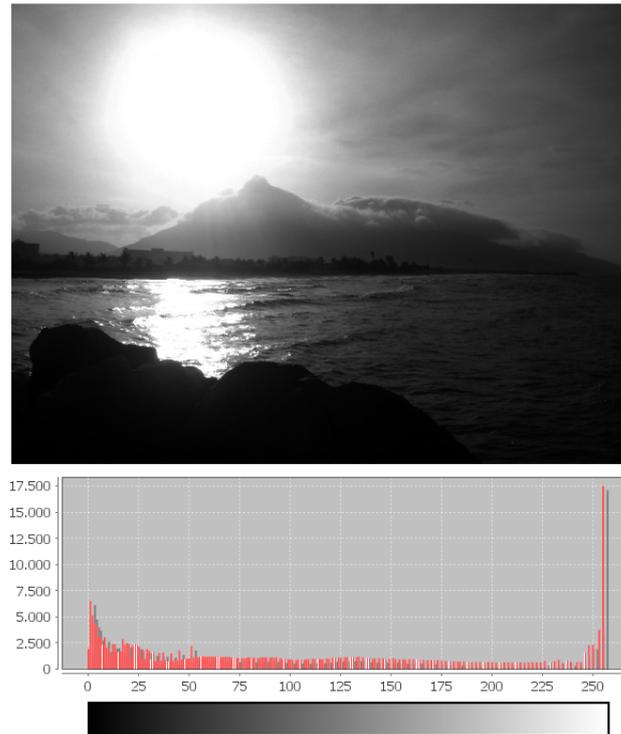


Figura 2.5. Imagen en escala de grises y su histograma.

imagen, un ejemplo se puede observar en la figura 2.6.



Figura 2.6. Imagen a color y los histogramas de sus componentes RGB.

### Ecualización del histograma

La ecualización del histograma es la técnica que reorganiza la distribución de las intensidades en el histograma haciendo que este sea más uniforme. Para aplicar esta técnica se debe obtener histograma original y normalizarlo. Luego se calcula la acumulada del his-

tograma y se obtienen los nuevos niveles de intensidad aplicando la siguiente expresión[9]:

$$S = \text{Int} \left[ \frac{s - s_{\min}}{1 - s_{\min}} \right] 255 + 0,5$$

Donde,  $\text{Int}$  es el entero más cercano,  $S_{\min}$  es el menor valor de la acumulada distinto de cero, y  $255$  es número de niveles de gris menos uno.

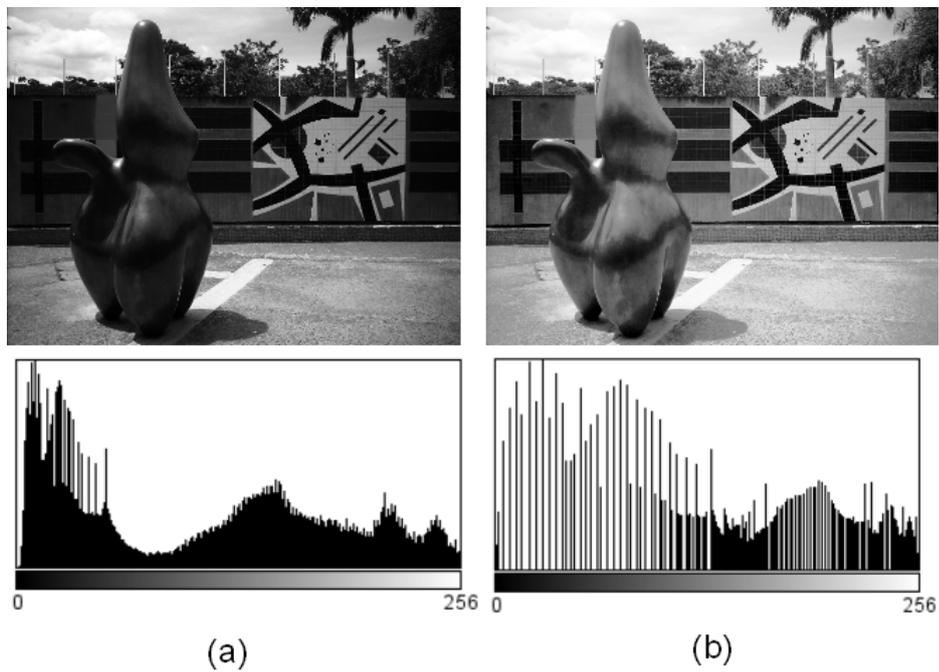


Figura 2.7. Ecuación del histograma. (a) Imagen original. (b) Imagen con el histograma ecualizado.

### Expansión del histograma

Es empleada para aumentar el contraste de una imagen. Una función para expandir los niveles de gris de un histograma se pueden ver de la siguiente manera [9]:

$$g(i, j) = \frac{f(i, j) - f_{\min}}{f_{\max} - f_{\min}} [MAX - MIN] + MIN$$

En este caso  $MAX$  y  $MIN$  representan el máximo y mínimo valor posible de los niveles de gris.

En la figura 2.8 se puede observar un ejemplo de expansión del histograma.

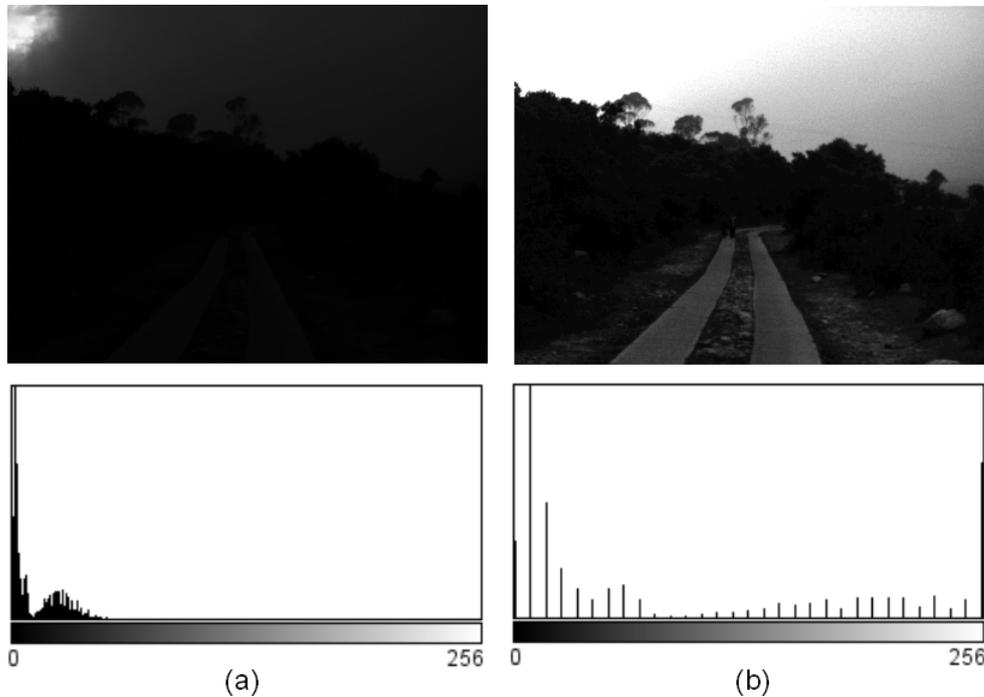


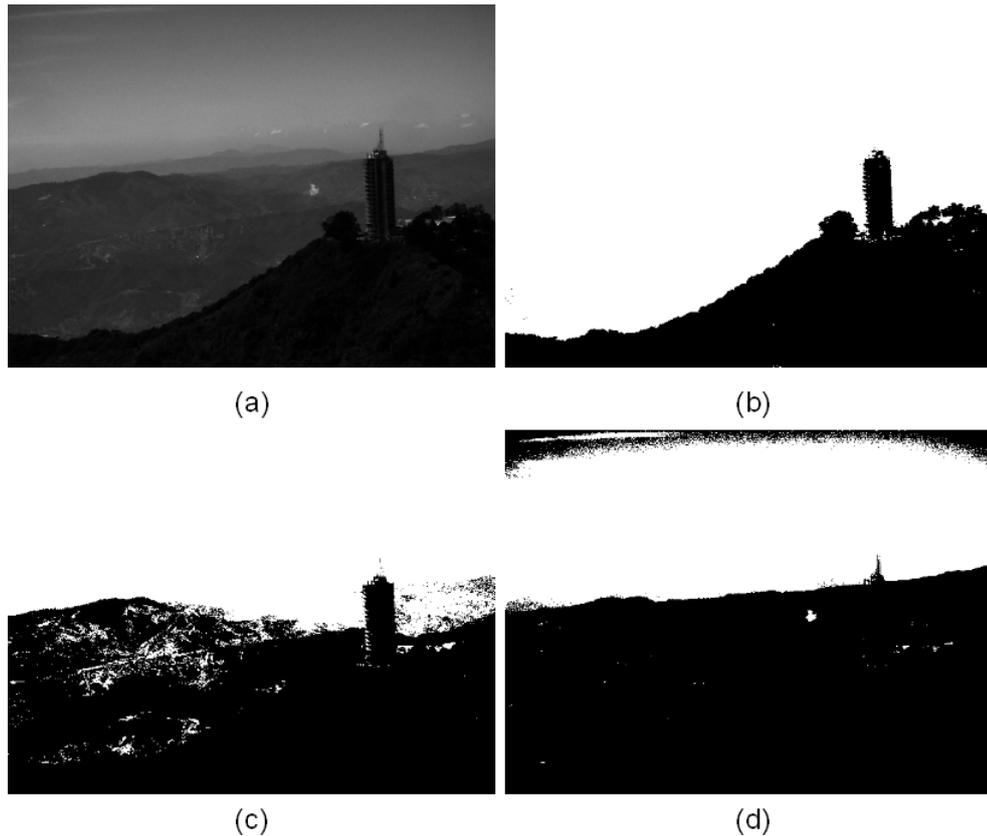
Figura 2.8. Expansión del histograma. (a) Imagen original. (b) Imagen con el histograma expandido.

## Umbralización

Consiste en cambiar los valores de los píxeles de una imagen con respecto a un valor que es llamado umbral. Esta técnica consiste en colocar en cero (o negro) todos los valores inferiores al umbral y en uno (o blanco) los valores mayores al éste. Con esta técnica es posible destacar formas u objetos de una imagen. En la figura 2.9 se puede observar la umbralización con diversos valores de umbral.

### 2.1.3. Filtrado

Según Sbert[4] filtrar una imagen ( $f$ ) consiste en aplicar una transformación para obtener una nueva imagen ( $g$ ) de forma que ciertas características se vean acentuadas o disminuidas.



**Figura 2.9.** Umbralización. (a) Imagen original. (b) Imagen Umbralizada con umbral de 96. (c) Imagen Umbralizada con umbral de 128. (d) Imagen Umbralizada con umbral de 160.

El filtrado permite lograr distintos efectos visuales tales como suavizar una imagen, disminuir la cantidad de ruido y detectar bordes de objetos presentes en la imagen. La convolución es una técnica que consiste en la construcción y aplicación de un operador a la imagen, lo cual producirá una modificación en cada píxel de acuerdo a la información de sus vecinos. Este operador es llamado máscara de convolución y está formado por matrices rectangulares de diferentes tamaños, en la práctica comúnmente se utilizan matrices cuadradas de tamaños  $3 \times 3$ ,  $5 \times 5$ ,  $7 \times 7$  [10].

### Filtros de promedio

Consiste en buscar la media del valor del píxel con respecto a sus vecinos en la máscara de convolución. El valor obtenido será el nuevo valor del píxel. Este filtro se usa generalmente para eliminar el ruido, mientras más grande sea la máscara más píxeles serán tomados en cuenta y más ponderado será el resultado, pero más información podría perderse.

## Filtro Gaussiano

La idea de esta técnica es hacer el efecto de “Campana de Gauss” a la imagen, lo cual significa dar más importancia al píxel central e ir disminuyendo la importancia a medida de que los valores estén más lejos del píxel original. El filtro gaussiano es usado comúnmente con el propósito de suavizado, detalles y ruidos que puedan estar asociados a la imagen.

En la figura 2.10 se pueden observar los resultados de aplicar los filtros del promedio y de Gauss.

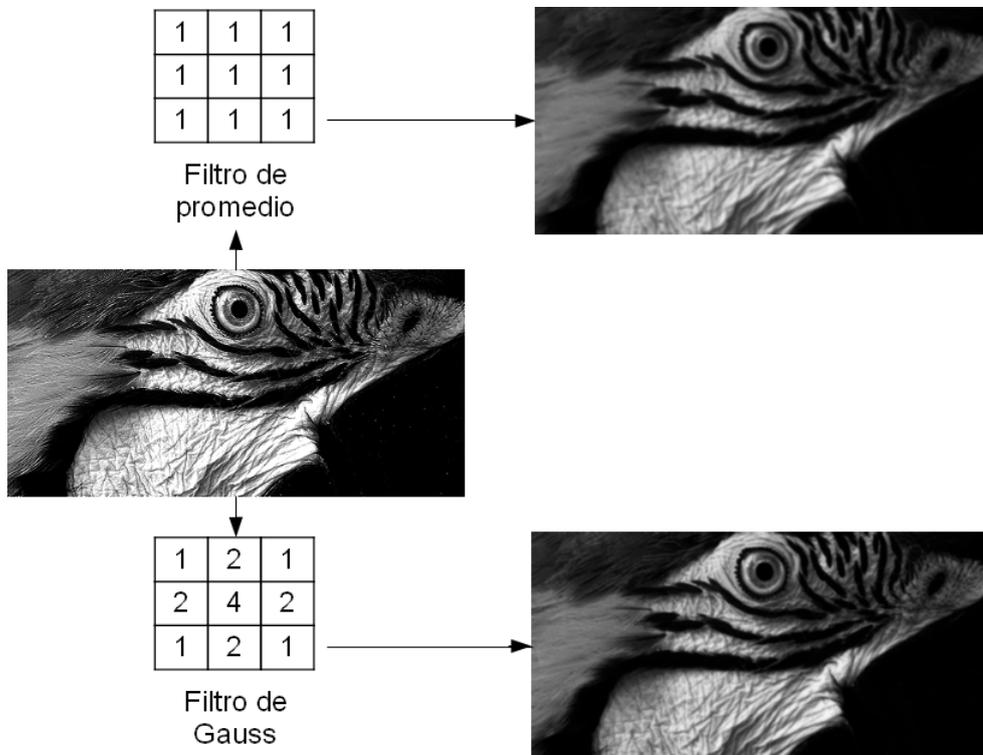


Figura 2.10. Filtros de suavizado: Promedio y Gaussiano

## Filtros de detección de bordes

Los bordes en una imagen se pueden definir como transiciones entre dos regiones de niveles de gris significativamente distintos [11]. Los bordes de los objetos proporcionan información útil sobre los objetos presentes en la imagen y para detectarlos existen diversos filtros. Según Lizarraga, C. y Ortega, L. [10] algunos de estos filtros operan en

una sola dirección (horizontal o vertical) como los de Sobel, Prewitt y Roberts; mientras el de Laplace se caracteriza por ser no direccional e invariante a rotaciones, es decir, el resultado no se ve afectado por la dirección del operador ni por la ubicación de la imagen.

En la figura 2.11 se observa el resultado de aplicar los filtros de Sobel, en la figura 2.12 el filtro de Roberts y en la figura 2.13 el filtro de Laplace.

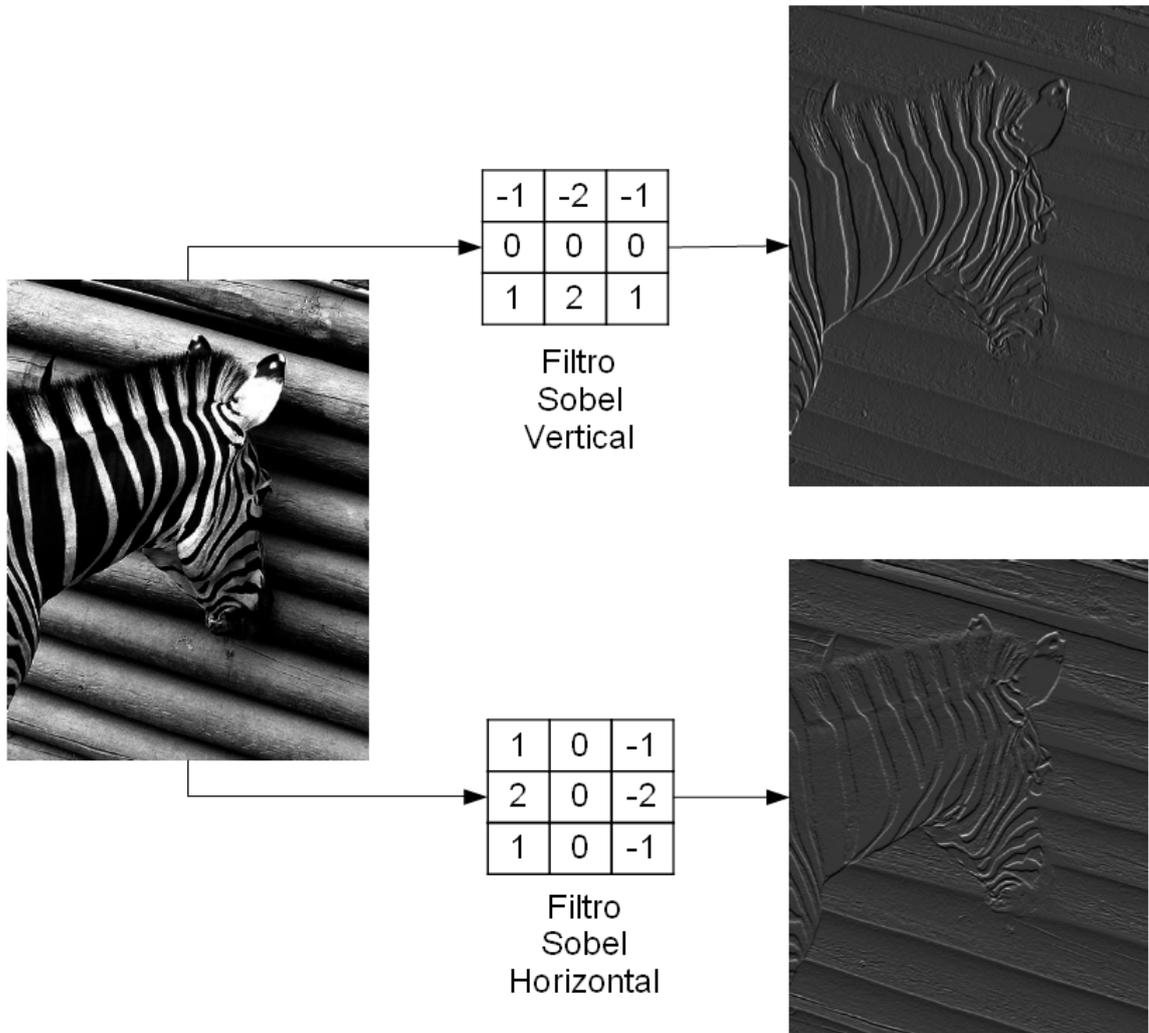


Figura 2.11. Aplicación del filtro de Sobel

### Operaciones morfológicas

Una de las operaciones más utilizadas sobre imágenes binarias son las operaciones morfológicas. Estas operaciones toman como entrada una imagen binaria y la transforma en otra imagen, donde el valor de cada píxel de la imagen resultante dependerá del valor

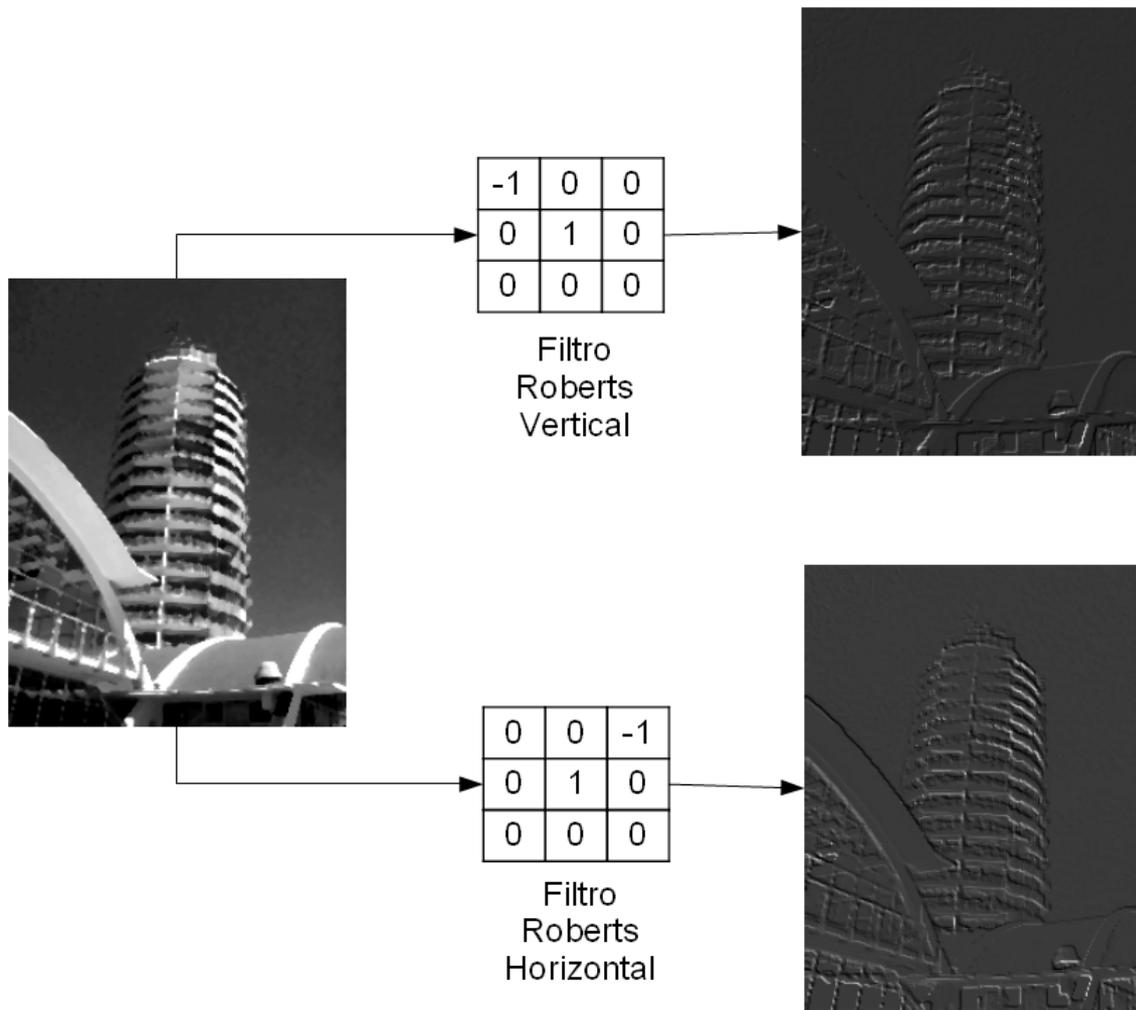


Figura 2.12. Aplicación del filtro de Roberts

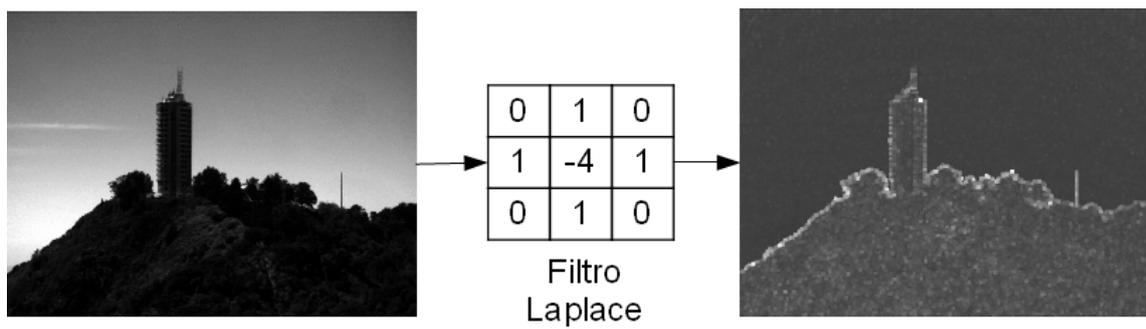


Figura 2.13. Aplicación del filtro de Laplace

correspondiente al píxel de la imagen original y sus vecinos.

Las operaciones morfológicas básicas son la dilatación y la erosión. La operación de dilatación adiciona píxeles en las fronteras de los objetos, mientras la erosión los remueve. En ambas operaciones se utiliza una matriz de convolución que determina cuáles vecinos del elemento central de la matriz serán tomados en cuenta para la determinación del píxel resultado. En este caso la matriz de convolución contiene *unos* y *ceros*, en los lugares que contiene *unos* serán los vecinos de la imagen original con respecto al píxel central, los cuales serán tomados en consideración para determinar el píxel de la imagen resultante, mientras que los lugares que tengan *ceros* no serán tomados en cuenta [17].

En la figura 2.14 se presentan ejemplos de erosión y dilatación dada una imagen original [18].

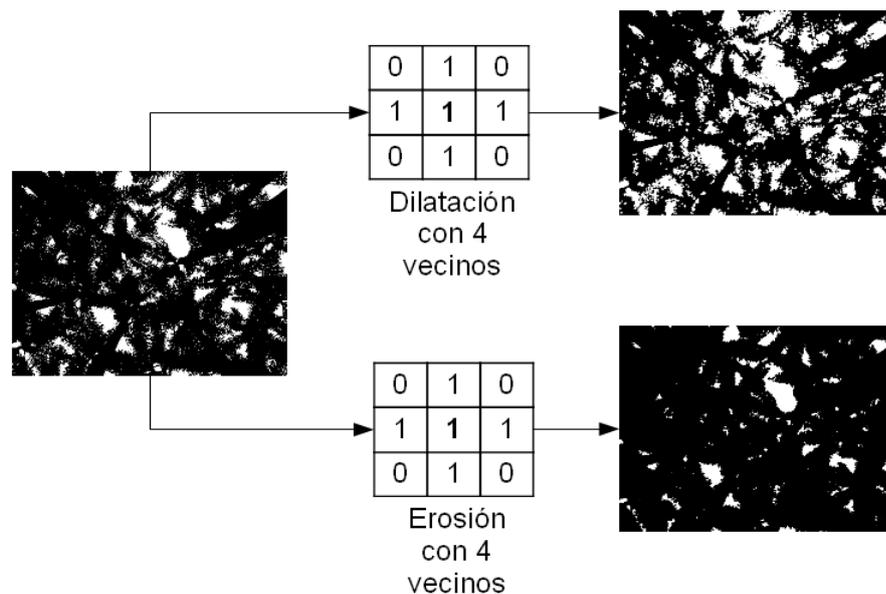


Figura 2.14. Operaciones Morfológicas.

## 2.2. Sistemas de Visión por Computador

Básicamente los sistemas de visión por computador se pueden clasificar en las siguientes etapas [3], como se muestra en la figura 2.15:

- **Captura:** en esta etapa se obtienen las imágenes a ser procesadas.



Figura 2.15. Etapas del proceso de visión por Computador.

- **Preprocesamiento:** consiste en realizar tareas que eliminan características innecesarias de las imágenes y facilitan el procesado posterior de la misma.
- **Segmentación:** consiste en aislar los elementos que existen en la escena, realizando técnicas como detección de bordes y regiones.
- **Reconocimiento:** en esta etapa se extraen las características deseadas de los objetos segmentados anteriormente, luego se utilizan estos datos para realizar el procesado de alto nivel para el cual fue diseñado el sistema de visión por computador.

En las siguientes secciones se describen de forma más detallada cada una de las etapas de un sistema de visión por computador.

### 2.2.1. Captura

Esta etapa conforma el proceso de adquisición de la imagen en un sistema de visión por computador. Consiste en captar la imagen de una escena y a través de un sensor que permite transformar la imagen a formato digital, tal que pueda ser procesada.

#### Dispositivos de adquisición de imágenes digitales

Existen diversos dispositivos utilizados para obtener imágenes en formato digital, de tal forma que puedan ser procesadas. Algunos de estos dispositivos capturan directamente las imágenes y las llevan a formato digital, mientras otros toman fotografías impresas o documentos y los digitalizan. A continuación se presentan los más utilizados: la cámara digital, la webcam y el escáner.

## Cámara digital

Según Hartley, R., Zisserman, A [5], una cámara es un dispositivo que permite hacer una proyección entre el mundo 3D a una imagen 2D. Una cámara digital es una cámara fotográfica con un dispositivo que captura imágenes a través de un sensor de imágenes electrónico. Existen básicamente dos tipos de sensores: Los dispositivos de carga acoplada (*Charge Coupled devices* o CCD) y los sensores basados en la tecnología Metal óxido semiconductor complementado (*Complementary metal oxide semiconductor* o CMOS). En la tabla 2.1 se observan las principales diferencias entre estas tecnologías [6].

CCD	CMOS
Tecnología Madura	Tecnología reciente
Excelente calidad de imagen	Buena calidad de imagen
Excelente inmunidad al ruido	Buena inmunidad al ruido
Acceso secuencial	Acceso aleatorio a píxel
Alto consumo de potencia	Bajo consumo de potencia
Alto coste de producción	Bajo coste de producción
Baja escala de integración	Alta escala de integración

Tabla 2.1. Comparación entre CCD y CMOS

## Webcams

Una webcam básicamente es una cámara digital conectada a una computadora, generalmente utilizan un sensor CMOS. Generalmente son utilizadas para tomar fotos y transferirlas a través de la web en tiempo real para su visualización [14].

Las webcams tienen distintas características con respecto a la tasa de captura. La velocidad de los *frames* indican el número de imágenes que la webcam puede capturar y transferir en un segundo. Para una buena transmisión de video se necesitan al menos 15 *frames* por segundo (fps), aunque lo ideal es 30 fps.

## Escáner

Según Monedero, J. [12] un escáner puede definirse como un instrumento óptico capaz de captar imágenes a través de un sensor que transmite impulsos lumínicos a impulsos electrónicos, y estos a información digital, Se utilizan para convertir fotos, dibujos y documentos de texto a formato digital. Los escáneres pueden clasificarse de la siguiente manera [13]:

- **Escáner de mano:** es un dispositivo de mano que pueden escanear imágenes o documentos pasándolo sobre estos. Sin embargo no son utilizados ampliamente ya que la precisión de los resultados depende de la habilidad del usuario, además si el documento a ser escaneado es más ancho que el escáner la página, ésta debe ser escanada por secciones. Estas secciones son unidas a través de software.
- **Escáner de sobremesa:** son el tipo más común de escáner. Cuando un documento va a ser escaneado es colocado sobre la superficie de vidrio del escáner. La cabeza del escáner y la fuente de luz debajo del vidrio se mueve bajo el documento a una velocidad constante. Estos escáners son muy versátiles ya que pueden escanear objetos de diferentes tamaños.
- **Escáner de alimentación (Tipo fax):** se diferencian de los escáneres de sobremesa ya que los componentes del escáner permanecen estacionarios, y es la página la que se mueve a través del dispositivo.
- **Escáner portátil (lápiz):** son útiles para digitalizar periódicos, libros y otros documentos. Usualmente incluyen un software para reconocimiento óptico de caracteres (OCR) y también para facilitar la conversión a el formato PDF. La mayoría de estos lápices trabajan por palabra u oraciones.

### 2.2.2. Preprocesamiento

Según Umbaugh, S. [1], los algoritmos y operaciones de pre-procesamiento se usan para facilitar la tarea de análisis posterior a la imagen. Estas incluyen operaciones relacionadas a la extracción de regiones de interés, ejecutar operaciones matemáticas básicas en las imágenes, mejoras simples de algunas características de la imagen y reducción de datos en cuanto a resolución y brillo.

Algunas de las operaciones aplicadas en la etapa de preprocesamiento en los sistemas de visión por computador incluyen la aplicación de filtros a la imagen que ayudan a realzar características importantes en la imagen o que eliminen características indeseadas como lo es el ruido. Otras técnicas muy utilizadas en esta etapa son las operaciones aritmético-lógicas, la manipulación del histograma para mejorar el contraste, las transformaciones geométricas, entre otras.

### 2.2.3. Segmentación

Es el proceso mediante el cual una imagen es dividida en grupos disjuntos (es decir, un píxel no puede pertenecer a dos grupos al mismo tiempo) con el propósito de separar las partes de interés de la imagen, usualmente en esta etapa se identifica la existencia del objeto.

Existen diversas técnicas para aplicar segmentación, entre las cuales están las transformaciones morfológicas, la umbralización, separación la imagen en regiones, detección de bordes, entre otros.

### 2.2.4. Reconocimiento

Segun Lizarraga, C., Ortega, L. [10] para la etapa de reconocimiento se toman los resultados obtenidos de todas las etapas anteriores, luego estos son interpretados y procesados para generar la información requerida por la aplicación. Se realiza un procesamiento de alto nivel, encargado de producir el resultado final, comúnmente en esta etapa se realizan tareas como detectar patrones, tomar decisiones, realizar acciones, entre otros.

Dependiendo del objetivo de la aplicación en esta etapa se suelen utilizar técnicas basadas en Inteligencia Artificial. Aunque también se suelen utilizar heurísticas, modelos algorítmicos o matemáticos para resolver el problema en cuestión.

Un ejemplo de las técnicas que se suelen utilizar en la etapa de reconocimiento es la triangulación, la cual es utilizada para encontrar un punto de interés en el espacio, dadas dos imágenes de diferentes cámaras [15]. La idea principal es determinar ese punto del objeto, dado que se conoce la distancia entre las dos cámaras que observan el mismo punto, se puede determinar la intersección de las dos líneas que pasan por el centro de la proyección y la proyección del punto en cada imagen [16]. En la figura 2.16 se puede observar un modelo básico de triangulación.

### 2.2.5. Calibración de la cámara

Como mencionamos anteriormente una cámara se encarga de realizar una proyección entre el mundo real a una imagen, al hacer esto la cámara aplica una transformación que permite relacionar el sistema de coordenadas tridimensional de la escena con el sistema de coordenadas bidimensional de la imagen.

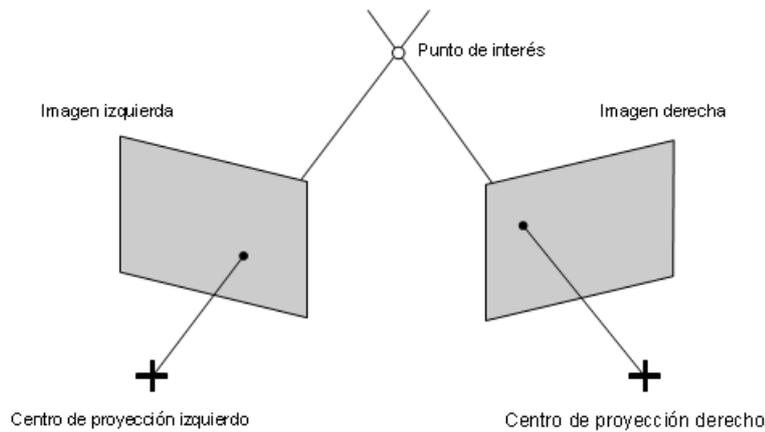


Figura 2.16. Modelo de triangulación.

Para obtener datos precisos de la escena tridimensional utilizando la imagen generada por la cámara es necesario realizar una transformación inversa que permita calcular las coordenadas tridimensionales correspondientes a cada punto de la imagen. Es por esto que es importante conocer con exactitud la transformación que realiza la cámara para poder construir un modelo matemático que realice la operación inversa. La calibración es el proceso que permite adquirir los parámetros de esta transformación [3]. Este proceso suele hacerse como paso previo a la captura de las imágenes en sistemas donde es necesario obtener información de la escena tridimensional.

Existen dos tipos de parámetros de la cámara que pueden ser obtenidos en el proceso de calibración:

- **Parámetros intrínsecos:** representan las propiedades físicas de la cámara y de su óptica. Los más importantes son la *distancia focal*, el *desplazamiento del centro de la imagen* y el *coeficiente de distorsión radial*.
- **Parámetros extrínsecos:** Definen la posición y orientación de la cámara con respecto al sistema de coordenadas de la escena, estos son la *traslación* y la *rotación*.

Las técnicas de calibración se pueden clasificar en parametrizadas e interpoladas:

- **Parametrizadas:** utilizan un modelo matemático para describir el comportamiento físico de la cámara.
- **Interpoladas:** no utilizan ningún modelo matemático, sino que toman cierta cantidad de referencias en la escena y sus correspondientes proyecciones en la imagen para

obtener la posición en la escena de cualquier otro punto tomando como referencia las posiciones conocidas.

La precisión lograda con técnicas parametrizadas es mayor a la lograda con técnicas interpoladas, a no ser que se utilice gran cantidad de puntos de referencia en la escena.

### 2.2.6. Visión Estéreo

Según Molleda, J. [3] las técnicas de visión por estereo tratan de emular el sistema de visión humano, en el que se analizan las diferencias de la proyección de la escena en dos imágenes tomadas desde dos posiciones diferentes. A las diferencias entre ambas imágenes se les denomina disparidad, y su análisis permite obtener la dimensión perdida en la proyección de la escena tridimensional en la imagen bidimensional. En el caso de los seres humanos el cerebro es el encargado de combinar ambas imágenes y de calcular su disparidad.

En el caso de un sistema de visión por computador, el análisis de las imágenes adquiridas simultáneamente por dos sensores distanciados espacialmente debe permitir establecer una correspondencia entre los puntos de ambas imágenes y realizar el cálculo de la profundidad a partir de las diferencias de posición de un punto entre las dos imágenes.

El cálculo de la correspondencia se realiza a partir de algoritmos que tratan de localizar las proyecciones de un punto de la escena en las dos imágenes capturadas, la principal limitación en este cálculo es que un único píxel de una imagen no proporciona información suficiente para identificar el punto correspondiente en la otra. Las soluciones que aplican estos algoritmos son analizar los píxeles de su vecindad por medio de técnicas de segmentación, correlación, detección, emparejamiento de bordes entre otros.

El cálculo de la profundidad se basa en el conocimiento de la geometría del sistema de visión por computador, especialmente en la posición que ocupa cada una de las cámaras en el espacio. En el caso que las dos cámaras estén perfectamente alineadas, para calcular las coordenadas tridimensionales de un punto de la escena se aplican relaciones entre triángulos semejantes obtenidos de la geometría del sistema. En situaciones en que la geometría del sistema no cumpla la configuración ideal se debe modelar la transformación que mezcla los sistemas de coordenadas de ambas cámaras, o aplicar un algoritmo de

rectificación a las dos imágenes que proporcione las mismas imágenes en la configuración ideal.

### 2.2.7. Triangulación Geométrica

Según Munro, P., Gerdelan, A. [15] la técnica para medir la información de profundidad de un punto dada dos imágenes desplazadas es conocida como triangulación. Esta técnica utiliza variables como: el punto central de las dos cámaras ( $O, O'$ ), la distancia focal de las cámaras ( $f$ ), sus ángulos, el plano de las imágenes y el punto de la escena sobre la imagen ( $A, A'$ ). Como se observa en la figura 2.17.

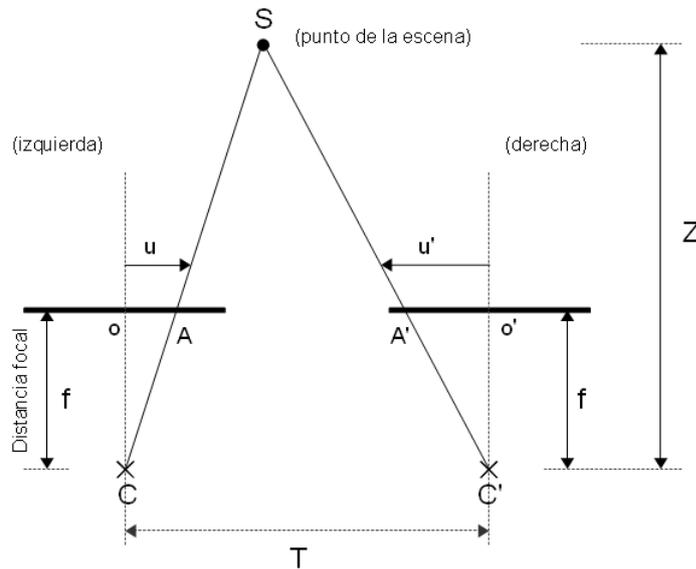


Figura 2.17. Triangulación Geométrica.

Para cualquier punto  $S$  de cualquier objeto en el mundo real,  $A$  y  $A'$  son los puntos en píxeles de representación de  $S$  tomadas por las cámaras ( $C, C'$ ). El valor de  $f$  es la distancia focal de la cámara (distancia entre el lente y el sensor). Cuando las cámaras se encuentran separadas por una distancia  $T$ , la localización de  $A$  y  $A'$  en el eje normal de las cámaras puede diferir. Sin embargo, ésta disparidad puede ser calculada tomando las diferencias entre  $U$  y  $U'$ . Como consecuencia es la disparidad la que permite calcular la distancia  $Z$  del punto  $S$  del objeto. De la relación geométrica de la figura 2.17 se deriva la siguiente fórmula:

$$Z = T \times \frac{f}{U - U'}$$

El ángulo de visión de las cámaras, es posible determinarlo empleando la fórmula: [19]

$$\alpha = 2 \times \arctan\left(\frac{L}{2 \times H}\right)$$

Donde L es la distancia captada por la cámara y H es la distancia entre la cámara y la superficie como se muestra en la figura 2.18.

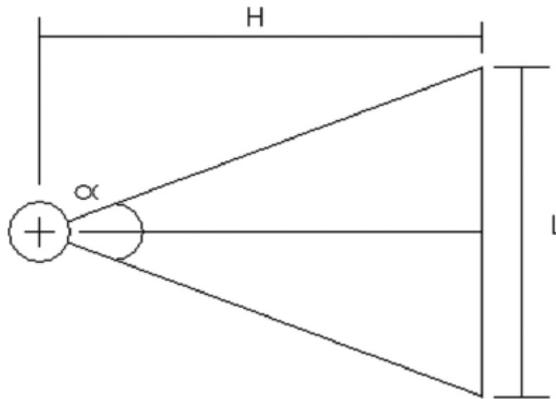


Figura 2.18. Cálculo de ángulo en una cámara.

### 2.3. Detección de movimiento

Según Sosa, J. [6] la detección de movimiento en muchos casos se utiliza para obtener la localización espacial de los objetos en una escena. Esta localización suele estar relacionada con la detección de cambios en la intensidad de los píxeles en una imagen. Sin embargo se puede decir que existen tres grandes grupos de problemas relacionados con el movimiento:

1. **La detección de movimiento:** consiste en registrar cualquier movimiento en la escena.
2. **Detección y localización de los objetos en movimiento:** se refiere a la detección del objeto, conocer la trayectoria del movimiento y predicción de su futura trayectoria.
3. **Obtención de propiedades 3D de los objetos:** estas propiedades se obtienen mediante el uso de un conjunto de proyecciones 2D adquiridas en distintos instantes de tiempo.

El principal problema de un sistema de visión es la cantidad masiva de datos a procesar debido al análisis que se realiza sobre imágenes consecutivas. Por otro lado trabajar con

una secuencia de imágenes hace inevitable introducción una nueva variable tiempo  $t$ . por lo que un sistema de análisis de movimiento, es representado ahora por:  $f(x, y, t)$ , donde  $x$  e  $y$  son las coordenadas espaciales de la imagen en un instante de tiempo  $t$ . El valor de la función  $f(x, y, t)$  representa la intensidad del píxel  $f(x, y)$  en la imagen en el tiempo  $t$ .

Los posibles problemas que pueden ocurrir en un sistema de visión por computador, van desde desde el tipo de modelo considerado para la formación de imagen hasta fenómenos que el ser humano no puede detectar. A continuación se explican algunos de estos:

1. **Modelo de formación de la imagen:** se refiere al instrumento que se utiliza para capturar las imágenes, partiendo desde el modelo usado, hasta el tipo de sensor utilizado por la cámara.
2. **Regiones con poca información:** esto se debe a la poca información del entorno próximo para determinar la existencia de movimiento, esto es conocido como “problema de apertura”.
3. **Cambios de intensidad sin la existencia de movimiento:** suelen ocurrir cambios de intensidad en los sistemas de visión, debido a la variación de iluminación en la escena que pueden ser considerados como movimiento.
4. **Movimientos múltiples:** muchas veces objetos en movimiento sufren rotaciones, expansiones o contracciones. Sin embargo en las aplicaciones se determina el movimiento basado en traslaciones sobre una región determinada, por lo que un solo vector de velocidad se considera para describir el movimiento en esa región.

Con la finalidad de solucionar los problemas antes mencionados se han propuesto múltiples algoritmos de estimación del movimiento. Existen diversas formas para la detección de movimiento para 2D y 3D. La mayoría de las técnicas de detección de movimiento 2D pueden clasificarse en 3 grupos: métodos basados en la obtención del flujo óptico, los métodos basados en la correspondencia y los métodos diferenciales. Esto se muestra en la figura 2.19.

### 2.3.1. Flujo óptico

Según Sosa, J. [6] el flujo óptico puede ser definido como el movimiento aparente de los niveles de intensidad de una imagen. Para determinar este movimiento es necesario obtener

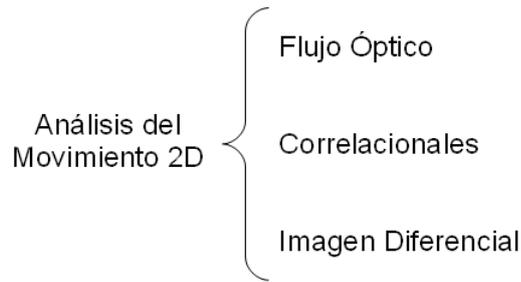


Figura 2.19. Clasificación de las técnicas de detección de movimiento en 2D.

las variaciones entre las intensidades de un píxel para cada imagen en cada instante de tiempo.

La obtención del flujo óptico finaliza con la determinación de la dirección del movimiento y de la velocidad del movimiento de todos los puntos de la imagen. El objetivo inmediato del análisis de imágenes basado en el flujo óptico es determinar el campo del movimiento. El campo de movimiento es el desplazamiento inducido en los píxeles de la imagen por el movimiento relativo de los objetos de la escena, como se aprecia en la figura 2.20.

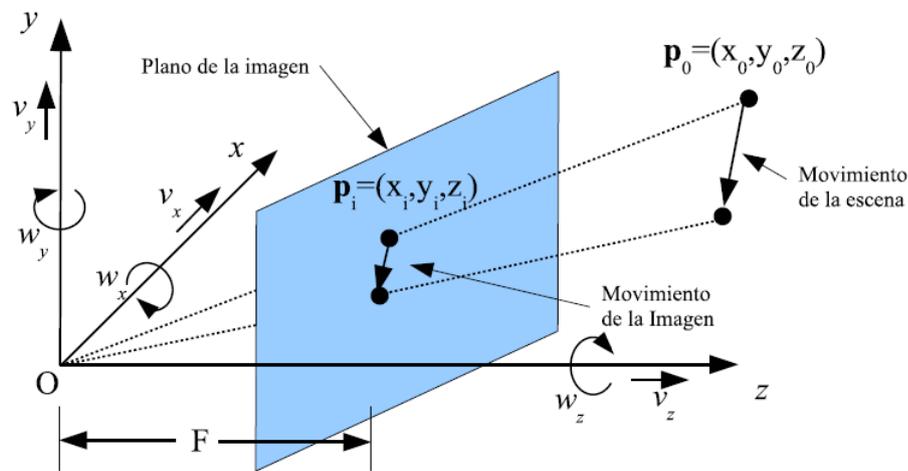


Figura 2.20. Geometría para el análisis del movimiento en el plano de la imagen y el movimiento inducido en un píxel.

El flujo óptico es un excelente método para medir el movimiento en un espacio bidimensional; puede ser usado para detectar objetos en movimiento aún en presencia de una cámara en movimiento. El problema es que su cálculo no es trivial, la mayoría de los métodos de cálculo de flujo óptico son computacionalmente complejos, y no pueden

ser aplicados a una transmisión completa de imágenes de video en tiempo real sin tener un hardware especializado.

### 2.3.2. Análisis del movimiento basado en la correspondencia

Esta técnica se comporta muy bien para grandes desplazamientos entre dos imágenes, algo que no sucede con el método de flujo óptico. Estos métodos se basan en la búsqueda de los vectores de desplazamiento entre las imágenes de referencia y la imagen actual, bajo un criterio de similitud.

### 2.3.3. Imagen Diferencial

Esta técnica es de las primeras en aparecer por su bajo costo computacional. El principio básico de esta técnica es que considera que cualquier movimiento perceptible en la escena se traduce en cambios de las imágenes tomadas, si estos cambios son detectados, se pueden analizar las características de este movimiento.

Estas técnicas se aplican en casos donde se pretende detectar la existencia del movimiento. También se utiliza en combinación con otras técnicas como sustracción de fondo, diferencias acumuladas, entre otros. Esto es útil ya que simplicidad para detectar zonas con movimiento permite concentrar un posterior esfuerzo computacional en el área detectada.

La imagen diferencia  $I_d$ , se puede definir como:

$$I_d(p, t_1, t_2) = I_2(p, t_2) - I_1(p, t_1)$$

Donde  $p=f(x, y)$  es un píxel de la imagen y  $t_1, t_2$  son los instantes de tiempo de dos imágenes consecutivas. Una variante de la ecuación consiste en formar la imagen resultante  $I_{out}$  mediante el siguiente sistema:

$$I_{out} = \begin{cases} I(p, t_2) & \text{si } I_d(p, t_1, t_2) \geq T_d \\ 0 & \text{en otro caso} \end{cases}$$

Donde  $T_d$  es un umbral previamente definido.

Las ventajas que presenta esta técnica son las siguientes:

1. Extremada simplicidad de procesado.

2. Es muy adaptable a medios dinámicos.
3. Puede ser utilizada para detectar de bordes siempre que la imagen diferencia obtenida sea a partir de dos imágenes ligeramente desplazadas.
4. Se presta a una cómoda implementación paralela.
5. Segmentando la imagen diferencial puede estimarse la dirección del movimiento del objeto.

Las desventajas de esta técnica es que la información que proporciona no es muy descriptiva sobre la forma y movimiento de los objetos. Otra desventaja es que carece de robustez frente a cambios de intensidad, por esto el sistema debe estar debidamente iluminado para evitar posibles fallas.

#### 2.3.4. Imágenes de diferencias acumuladas

Esta técnica consiste en tener una acumulación de diferencias de imágenes con el fin de tener una historia de los movimientos para un período de tiempo. Este método fue uno de los primeros métodos de detección en ser implementado. En este método se usan diferencias de imágenes para estimar objetos en movimiento, al incrementar el número de imágenes pueden estimar las regiones de movimiento de uno o más objetos en movimiento, así como el fondo estacionario [8]. La imagen diferencial se representa como:

$$I_t(p, t_n) = Id(p, t_{n-1}, t_n) - Lt(p, t_{n-1}) \text{ para } n \geq 3$$

$$I(p, t_2) = l_d(p, t_1, t_2)$$

Este proceso recursivo permite acumular en  $I_t$  la información del movimiento en el periodo del tiempo seleccionado [6].

#### 2.3.5. Sustracción de fondo

Según Sosa, J [6], este método es considerado como una combinación de dos técnicas íntimamente ligadas. Una que se encarga de obtener el fondo y otra que se encarga de localizar la diferencia entre la imagen de referencia (fondo) y la imagen actual.

A pesar de que existen muchos algoritmos de detección de fondo (*background*), la mayoría de ellos sigue un diagrama de flujo simple como se muestra en la figura 2.21. Los pasos más importantes en un algoritmo de sustracción de *background*, son procesamiento,

modelado del background, detección de objetos de primer plano (*foreground*) y la validación de datos.

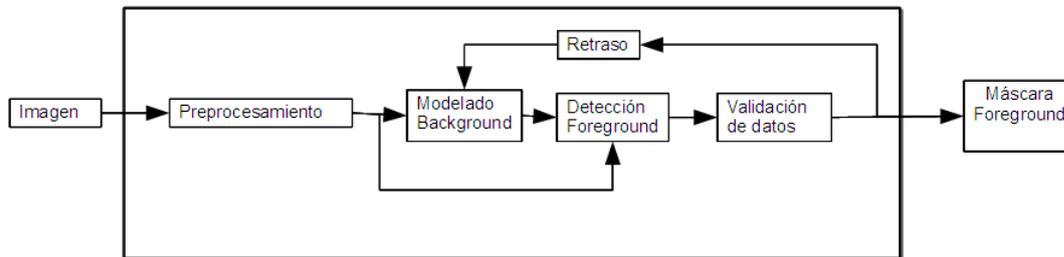


Figura 2.21. Diagrama de Secuencia, de un Algoritmo genérico de sustracción de fondo.

El preprocesamiento consiste en una colección de tareas simples de procesamiento de imágenes para eliminar características no deseadas. El modelado de *background* usa la imagen para calcular y actualizar el modelo, este modelo de background provee una descripción estadística de todo el fondo de la escena. La detección del foreground identifica píxeles en la imagen que no concuerdan con el modelo de background, y generan un candidato a objeto de foreground. Finalmente la validación de datos examina los candidatos y elimina aquellos píxeles que no corresponden a objetos en movimiento[7].

Esta técnica es comúnmente utilizada en sistemas de vigilancia empleando cámaras fijas donde se realiza una sustracción del fondo de la escena. Cuando se registra movimiento, estas imágenes son capturadas y archivadas.

En el siguiente capítulo se estudian diferentes formas donde la visión por computador se ha aplicado al problema de las pizarras electrónicas.

## 2.4. Interpolación

La interpolación es un procedimiento numérico usado para aproximar valores de una función. Es utilizada cuando no disponemos de mayor información y deseamos conocer los valores que toma la función  $f(x)$ . Una manera de resolver este problema es obtener un valor aproximado  $y$  en la función  $f(x)$ , utilizando siguiente procedimiento:

1. Tomamos una función  $h(x)$  conocida, que coincida con  $f(x)$  en dos puntos dados, entonces:  $h(a) = f(a)$  ,  $h(b) = f(b)$  y este cercana al intervalo  $[a, b]$  con  $a < b$ .
2. Siendo  $c$  un valor perteneciente al intervalo  $[a, b]$ , se calcula  $y = h(c)$ .

3. Dada la proximidad de las dos funciones en el intervalo  $[a, b]$ , entonces se establece  $y$  como una aproximación de  $f(c)$ .

Este método de aproximación para obtener un valor desconocido situado entre dos valores conocidos, se conoce como interpolación. Existen muchas funciones para interpolar, y la más conocida por la facilidad de sus cálculos es la función polinómica. Dependiendo del grado de la función polinómica se derivan interpolaciones como: Interpolación Lineal cuando la función polinómica es de primer grado, interpolación parabólica si es de segundo grado, e interpolación cúbica si es de tercer grado, siendo la interpolación lineal la más sencilla por la simplicidad de sus cálculos.

### 2.4.1. Interpolacion Lineal

Es la fórmula más simple de interpolación y consiste en conectar dos puntos conocidos con una línea recta. Como se muestra en la figura 2.22

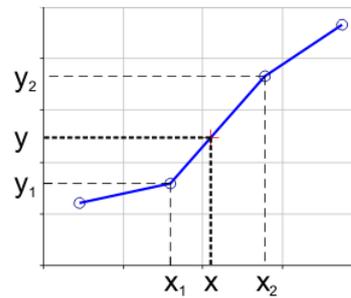


Figura 2.22. InterpolacionLineal

Dados dos puntos  $(x_1, y_1)$  y  $(x_2, y_2)$ , la imagen de un punto  $x$  es:

$$y = y_1 + (x - x_1) \frac{y_2 - y_1}{x_2 - x_1}$$



---

## Pizarras Interactivas

### 3.1. Evolución de las Pizarras

En un principio las clases magistrales se conformaban por una pizarra negra en el cual el profesor o presentador escribía con tiza de ceniza para todos en el salón. Luego vinieron las pizarras blancas de material acrílico en el que se podía escribir con marcador, estas hacen el mismo trabajo que las pizarras negras, sólo que en estas tenían mayor nitidez, adicionalmente se pueden usar como una superficie de proyección para las presentaciones. En la figura 3.1 se muestra un ejemplo de estas pizarras.

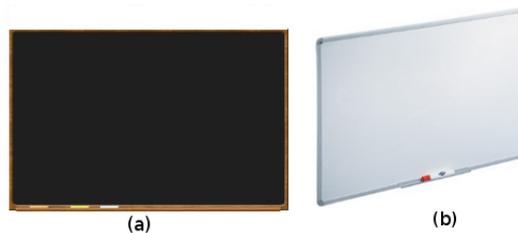


Figura 3.1. (a) Pizarra negra (b) Pizarra acrílica

La desventaja de ambas pizarras, es que se tiene que copiar en papel para guardar la información. Los rotafolios o papelógrafos fue otro progreso, donde el presentador podía escribir y dibujar en largas hojas de papel, por lo que las notas podían mantenerse. Estos rotafolios eran especialmente útiles cuando una lluvia de ideas o *brainstorming* necesitaba ser guardada.

Para evitar escribir sobre papel y la pérdida de atención de los asistentes a una clase o presentación surgieron las *copyboards* (pizarras copiatoras), mostrada en la figura 3.2, estas son pizarras blancas que permiten imprimir todo lo que se ha escrito o dibujado, además guardar, editar y enviar por correo. Algunas *copyboard* funcionan conectándose a

un computador para descargar los datos, mientras otras no necesitan de un computador y permiten guardar la información en una memoria USB o permite la conexión directa de una impresora. Las *copyboards* actuales permiten proyectar sobre ellas y unir lo que está siendo proyectado con lo que ha sido escrito sobre ella, para luego ser distribuido[20].



**Figura 3.2. CopyBoards o Pizarras Copiadoras**

Un **caballete digital** (*digital easel*) es otra alternativa, que permite escribir tus notas dentro de esta especie de pizarra y luego salvarla directamente dentro del computador, para posteriormente enviarlo por correo o imprimirlo, siendo así una herramienta para crear, guardar y distribuir información. Algunas empresas de manufactura permiten el uso de papel si lo desea [21]. Ver figura 3.3.



**Figura 3.3. Caballete Digital**

Posteriormente surgieron las pizarras interactivas que pueden actuar de la misma forma que una copyboard, sin embargo las pizarras interactivas están diseñadas para interactuar con la computadora, y un proyector multimedia o videobeam.

Un **Video Beam** o **proyector multimedia** toma la imagen que se muestra en la pantalla de la computadora y la proyecta sobre una gran pantalla de proyección blanca. Esto permite que todo lo que aparezca en el computador sea visto por todas las personas en la sala. Los proyectores son llamados multimedia porque pueden usarse con una fuente audio/visual tal como grabadoras de video, reproductor de DVD o sistemas satelitales. Una pizarra interactiva puede reemplazar la habitual proyección de

un proyector multimedia, utilizando la imagen de la computadora y proyectándola sobre esta pizarra, con lo cual el sistema se vuelve interactivo. En la figura 3.4 se observa un Video Beam [21].



**Figura 3.4. Video Beam o Proyector Multimedia**

## 3.2. Pizarra Interactiva

Es un panel físico que puede funcionar como una pizarra común, adicionalmente puede servir como una *copyboard* o como una pantalla de proyección donde la imagen de la computadora puede ser controlada tocando o escribiendo sobre la superficie del panel[22].

La configuración de la pizarra interactiva viene dada por la conexión vía cable USB o señal inalámbrica a la computadora, luego se debe cargar el software para utilizarla como pantalla táctil, y conectar el proyector a la computadora para proyectar la imagen sobre la pizarra interactiva.

Las pizarras interactivas completamente funcionales usualmente están compuestas por cuatro componentes: una computadora, un proyector, el panel de despliegue (pantalla de proyección) y el software apropiado. La mayoría de estos software permiten que se guarde todo lo escrito en la pizarra a la computadora. Esto es particularmente útil para agregar anotaciones a las fotografías, o dibujando en documentos existentes, lo cual hace que una pizarra interactiva pueda convertirse en una *copyboard*[21].

Las pizarras interactivas ayudan ampliamente en la enseñanza, entre las ventajas que tienen se pueden mencionar los siguientes [22]:

- ▷ Facilita a los educadores la mejora de la presentación del contenido integrando fácilmente un amplio rango de materiales en una lección, tal como imágenes de internet, un gráfico de una hoja de cálculo o el uso de cualquier aplicación que ayude a explicar algo.

- ▷ Permiten a los educadores crear fácil y rápidamente contenido y adaptarlo a las necesidades de la clase en tiempo real.
- ▷ Permiten a los alumnos participar en la discusión sin preocuparse de tomar notas, ya que el contenido de la clase puede ser distribuido fácilmente de manera digital.

Como se menciona en [22], entre las desventajas de las pizarras interactivas están:

- ▷ Son más costosas que las pizarras convencionales o que la combinación de proyector y pantalla. Su superficie puede ser dañada, necesitando un reemplazo costoso. Las pizarras de proyección frontal pueden ser oscurecidas por uno o más usuarios.
- ▷ Las pizarras de altura fija en algunas ocasiones están colocadas demasiado altas para que los usuarios alcancen la parte de arriba o demasiado bajas para que sea visible para todos los usuarios.
- ▷ Las pizarras móviles (y sus proyectores asociados) son más difíciles de asegurar y necesitan ser realineados (o calibrados) cada vez que son movidos.
- ▷ Si la entrada de múltiples datos es permitida, la entrada puede mezclarse, produciendo un resultado no deseado.

### 3.3. Clasificación

Las pizarras interactivas pueden clasificarse en dos formas básicas, por el tipo de proyección y por la tecnología que utilizan.

#### 3.3.1. Proyección

##### Frontal

En las pizarras de proyección frontal se tiene el proyector frente a la pizarra. Si el presentador se encuentra por delante de la pantalla, este puede causar sombras, además que existe el riesgo de ver la luz del proyector que puede dañar sus ojos. En la imagen 3.5 se muestra una pizarra de proyección frontal con video beam en la parte superior[21].

##### Trasera

En las pizarras de proyección trasera el proyector se sitúa detrás de la pizarra, evitando las sombras que puedan ocurrir, esto da un acabado limpio haciendo parecer como la



**Figura 3.5. Pizarra de Proyección Frontal**

pizarra como un gran monitor. Este tipo de pizarras se puede encontrar sostenida de gabinetes, o sobre la pared. La desventaja es que estos sistemas son mucho más caros que las pizarras de proyección frontal. En la imagen 3.6 se muestra una pizarra de proyección trasera.



**Figura 3.6. Pizarra de Proyección Trasera**

Dentro de esta clasificación se pueden incluir las pantallas de plasma, estos monitores pueden encontrarse generalmente de 42", 50", 60" o 84" medido en diagonal. La funcionalidad de pantalla táctil puede encontrarse para pantallas de 42", 50" y 60" dando la misma funcionalidad que una pizarra interactiva. Estas tienen las ventajas de una pizarra de proyección trasera pero son mucho más costosas que las pizarras de proyección frontal[21].

### **3.3.2. Tecnologías**

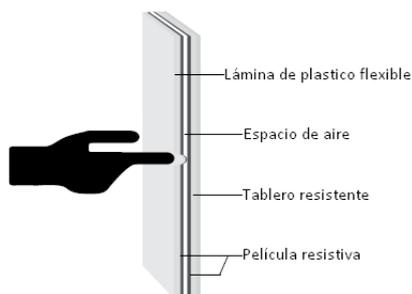
Otra forma de clasificación depende de las tecnologías utilizadas por las pizarras interactivas:

#### **Membrana resistiva**

Estas pizarras tienen una superficie suave y flexible, similar al vinil, que consiste en dos piezas de material resistivo separado por un pequeño espacio que crea una membrana sensible al tacto.

Puede ser utilizada con los dedos o utilizando un marcador especial que puede representar bolígrafos de diferentes colores seleccionados a través del software. El movimiento es seguido detectando la presión del marcador sobre la superficie. Las coordenadas se corresponden al área en el monitor de la computadora.

En la imagen 3.7 se muestra cada una de las partes que compone una pizarra con tecnología resistiva.

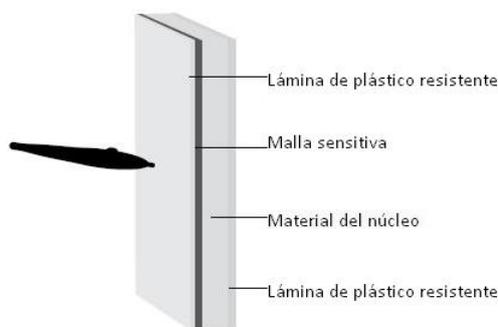


**Figura 3.7. Tecnología resistiva.**

## Electromagnética

Estas pizarras se parecen a las tradicionales en que tienen una superficie dura y pueden ser utilizadas con marcadores normales. Para trabajar interactivamente con ellas se requieren marcadores especiales alimentados con una batería que emite un pequeño campo magnético detectado por la pizarra o por una malla de cables finos integrados bajo la superficie de la pizarra.

En la imagen 3.8 se muestra cada una de las partes que compone una pizarra con tecnología electromagnética.



**Figura 3.8. Tecnología electromagnética.**

### **Cámara HD(High Definition)**

Al igual que las electromagnéticas estas pizarras se pueden utilizar con marcadores normales y tienen una superficie dura, la diferencia es que esta superficie tiene un patrón de puntos casi invisible al ojo, que puede ser leído por un bolígrafo digital que graba e interpreta los trazos escritos a mano [23]. Éste bolígrafo digital posee una micro cámara capaz de captar estos puntos para calcular las coordenadas donde se está escribiendo y enviar esta información al computador inalámbricamente[24].

### **Ultrasonido e infrarrojo**

Algunas pizarras interactivas utilizan un dispositivo en un borde con dos sensores de ultrasonido con una distancia conocida entre ellos y un sensor infrarrojo. Funcionan con un bolígrafo especial que emite estos dos tipos de señal cuando toca la superficie de la pizarra. La señal infrarroja viaja instantáneamente (a la velocidad de la luz) al receptor infrarrojo. La señal infrarroja permite saber al dispositivo que una señal de ultrasonido fue enviada. La señal ultrasónica llega después de la infrarroja (a la velocidad del sonido) a ambos receptores ultrasónicos en el dispositivo. Ahora utilizando la velocidad del sonido y la señal inicial el sistema puede calcular la distancia entre el bolígrafo y los receptores ultrasónicos utilizando trigonometría simple [25].

### **Seguimiento Infrarrojo**

La tecnología de seguimiento infrarrojo se basa en un pequeño dispositivo que puede ser colocado en cualquier superficie y la escanea con una señal infrarroja. Cuando se conecta a la computadora y se calibra de acuerdo a la imagen proyectada, esta imagen se puede utilizar como una superficie interactiva.

Una desventaja de utilizar esta tecnología es que necesita que la superficie este completamente libre para poder escanearla continuamente. Esto puede ser problemático si en algún momento se tiene algún objeto en la pizarra o si el usuario se apoya en ella [26].

## **3.4. Antecedentes**

Existen iniciativas de personas que buscan ofrecer las funcionalidades de las pizarras interactivas presentes en el mercado, utilizando dispositivos de bajo costo. Estas

propuestas utilizan dispositivos que sirven para detectar y emular el movimiento del ratón. Para estas pizarras también se necesita el desarrollo de un software que permita el uso de esos dispositivos y el proceso adecuado para lograr las funcionalidades de las pizarras interactivas, esto puede incluir el desarrollo de algoritmos de procesamiento de imágenes y calcular las coordenadas para el movimiento del ratón, como se explica en las siguientes secciones.

### 3.4.1. LoCoBoard

Es una pizarra interactiva desarrollada y compuesta de código abierto que puede ser instalado en un cualquier computador. Este software utiliza una *webcam* que detecta dispositivos o apuntadores que emitan luz infrarroja reflejada sobre una superficie de proyección. Este sistema fue desarrollado en octubre del año 2009 por Cristophe Soares de la Universidad Fernando Pessoa (Porto, Portugal) como trabajo para obtener el grado de Maestría [27].

#### Arquitectura del Sistema

LoCoBoard emplea un lápiz infrarrojo en su arquitectura para emitir señales que sean capturadas por la *webcam*. Un lápiz infrarrojo es el emisor de luz infrarroja que se utiliza para indicar la acción o movimiento que se quiere realizar sobre la superficie de proyección, de esta forma se logra simular el mouse del computador. Este lápiz puede ser comprado o construido fácilmente con materiales de bajo costo. En la figura 3.9 se muestra un ejemplo de un lápiz infrarrojo.



Figura 3.9. Lápiz infrarrojo, construido con materiales de bajo costo.

La proyección de la pantalla del computador puede hacerse por un video proyector simple. Para aislar la cámara de fuentes de luz indeseadas se utiliza un filtro de luz que

solo deja pasar infrarrojos (puede ser una película fotográfica doblada varias veces). Se utiliza como dispositivo apuntador un lápiz infrarrojo.

El funcionamiento del sistema se inicia con una fase de calibración para establecer la relación entre la resolución de la pizarra y la del computador y poder así mapear los puntos de interés detectados en el movimiento del cursor. Permite manejar el computador realizando desplazamientos con el apuntador infrarrojo sobre la superficie de proyección tal como si fuera un ratón. El sistema detecta una posición e interpreta la acción que el usuario quiere realizar en un determinado punto de proyección. En la figura 3.10 podemos observar la arquitectura de este sistema.

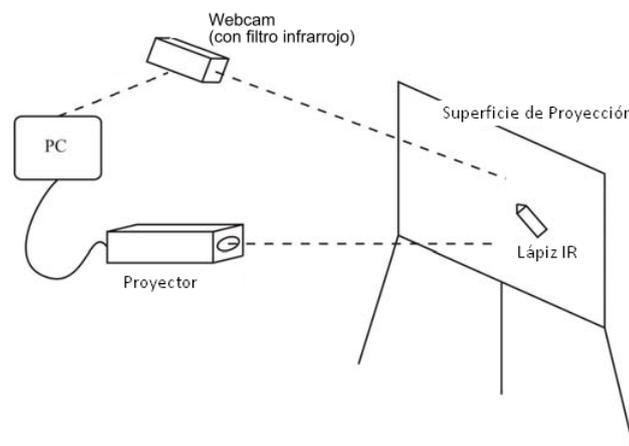


Figura 3.10. Esquema de sistema LocoBoard.

### Etapas del sistema

En la figura 3.11 se observan los cuatro procesos más importantes realizados por el sistema. El primero corresponde a la adquisición de la imagen capturada por la cámara. En el segundo se aplican filtros a la imagen capturada para mejorar la calidad de la misma; opcionalmente en esta etapa se pueden aplicar técnicas de eliminación de fondo (background). Estas permiten obtener un modelo de background que se puede sustraer a cada nueva imagen obteniendo como resultado el foreground de cada imagen. En una tercera fase se utilizan los algoritmos de detección y seguimiento. Al final se reportan las coordenadas del punto encontrado en caso de que exista [27].

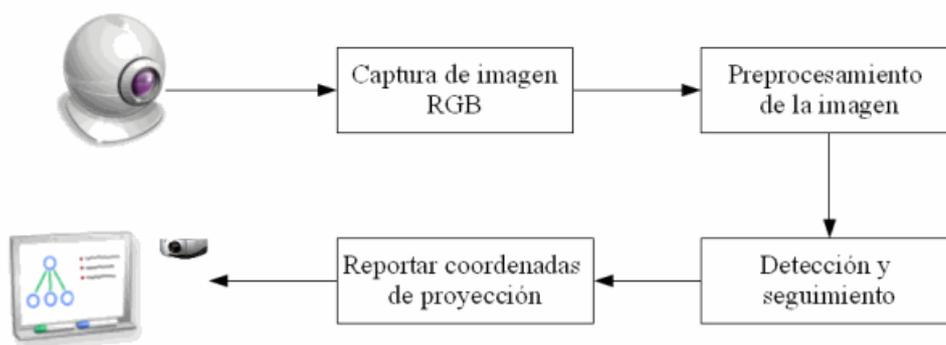


Figura 3.11. Diagrama de Secuencia. Sistema Locoboard

## Funcionamiento del sistema LoCoBoard

### *Captura de la señal de video*

En este paso se establece una relación entre la pizarra y el dispositivo de captura, es decir, la cámara web. Al iniciar el sistema se pueden establecer parámetros para la captura de las imágenes como **frames** por segundo (fps) o la resolución. Luego de establecer estos parámetros la cámara empieza a capturar las imágenes.

### *Preprocesamiento de la imagen*

Después de haber capturado una imagen se efectúa un tratamiento previo a la imagen para disminuir el ruido causado por condiciones del ambiente que no pueden ser controladas. A pesar de usar un filtro de luz IR, muchas veces hay fuentes de ruido presente en el ambiente (incluyendo luz infrarroja) que perturban el buen funcionamiento de los algoritmos de detección. A fin de detectar mejor los puntos de interés evitando confundirlos con los puntos de ruido presente en la sala, el sistema utiliza un modelo de background-foreground en cada una de las imágenes. El modelo de background que utiliza es adaptativo, es decir, a medida que el programa se ejecuta este se va aprendiendo, detectando y actualizando el background. Posteriormente se efectúa una sustracción del background estimado en la imagen capturada.

### *Aplicación de los algoritmos de detección y seguimiento*

Una vez que la imagen ha sido pre-procesada se procede a buscar el punto señalado con el apuntador infrarrojo. Primero se convierte la imagen a escala de grises, en el

procesamiento en tiempo real de cada imagen, esta transformación se vuelve fundamental, ya que el análisis se puede hacer tres veces más rápido, al no tener que verificar las componentes RGB.

Luego se aplica el algoritmo de detección y seguimiento de los puntos generados por el apuntador infrarrojo. Durante el desarrollo del sistema se utilizaron varios algoritmos para este fin. Algunos de estos algoritmos son explicados brevemente a continuación:

- El algoritmo más básico realiza una búsqueda lineal sobre la imagen y retornando la coordenada del centroide del punto encontrado.
- Otro algoritmo incluye una noción de predicción, que permite a la aplicación en cada momento prever donde va a aparecer un punto de interés en un nuevo frame conociendo su ubicación anterior y comenzando a partir de esa ubicación una búsqueda en espiral.

La fase posterior a los algoritmos de detección, reporta en forma de coordenadas cartesianas (x,y) los valores estimados de los puntos de interés. Con base en estos valores es posible interactuar y desencadenar acciones, de forma que el sistema interactúe con el usuario. Esta puede ser vista como una etapa de interpretación lógica. En el ámbito de la pizarra interactiva se definen dos tipos de acciones: click y movimiento/arrastrar.

Antes de empezar a usar el sistema se requiere que se realice la calibración del mismo, esto se hace para establecer una relación entre la proyección y la imagen capturada por la webcam. También permite establecer la relación entre la resolución de la proyección y la resolución de la imagen capturada por la webcam.

Una vez obtenidas las coordenadas de los puntos de interés y transformadas a la resolución del sistema operativo se ejecuta el intérprete del ratón. Este intérprete debe tener la inteligencia para detectar, a partir de las coordenadas de los puntos en formato cartesiano, las acciones que el usuario pretende desencadenar. Estas acciones pueden ser clicks del ratón, operaciones de arrastrar o simples movimientos. Del mismo modo, el intérprete deberá desencadenar las acciones necesarias para mover el apuntador del ratón para las coordenadas esperadas.

### **Materiales necesarios**

- Cámara web.
- Bolígrafo con un LED IR para apuntar a la superficie de proyección.

- Filtro Infrarrojo (se puede utilizar una película fotográfica).
- La aplicación LoCoBoard, que es multiplataforma.

## Experiencias

Actualmente el proyecto desarrollado por Cristophe Soares se encuentra en la página “<http://code.google.com/p/locoboard/>”, desde donde puede ser descargado para ser utilizado y modificado libremente.

Al igual que LoCoBoard existe otro proyecto de código abierto que se basa en el mismo principio de utilizar una webcam y un apuntador infrarrojo para convertir una superficie de proyección en una pizarra interactiva. Este puede ser descargado la página “<http://www.webcam-whiteboard.com/>”. En esta página se pueden encontrar instrucciones para instalar el sistema y ponerlo en funcionamiento.

### 3.4.2. WiiBoard

Es una pizarra interactiva que se compone de dos hardware separados; un lápiz infrarrojo (IR) y un control remoto Wii (Wiimote) [29], adicionalmente utiliza el software Wiiboard [28] que permite calibrar y conectar el control del Wii al computador. Este proyecto fue desarrollado por Johnny Chung Lee de la Universidad de Carnegie Mellon (Pittsburgh, USA) en diciembre del año 2007.

### Funcionamiento del Wiimote

El control de Wii, conocido como “Wiimote”, es el dispositivo de control del sistema de la nueva consola de Nintendo. Sus características más destacables son la capacidad de detección de movimiento en el espacio y la habilidad de apuntar hacia objetos situados en la pantalla. Es un control con detección de movimiento que permite el desarrollo de juegos dinámicos y atractivos. Sus capacidades de puntero y detección de movimiento permiten controlar entornos tridimensionales sin la complicación de los clásicos joysticks y combinaciones de diferentes botones. A nivel técnico, son dos las tecnologías fundamentales del Wiimote: un sensor infrarrojo y un acelerómetro. [30]

### Arquitectura del Sistema

El Wiimote se fija en una posición y se conecta al computador vía *Bluetooth* para seguir la posición de la luz infrarroja y transmitir las coordenadas al computador. Cuando el

lápiz infrarrojo se arrastra sobre la superficie de proyección, esto es considerado como un movimiento, que luego es transmitido al computador como una acción del mouse. De esta forma se logra la sensación de trabajar sobre una pizarra interactiva.

El Wiimote probablemente trabajará mejor cuando esté ubicado a 45° de ángulo de la superficie a ser usada. Esto dará suficiente área de trabajo en la superficie sin obstruir la vista del Wiimote.

La desventaja de este trabajo es la necesidad de depender del control del Wii y tener un lápiz infrarrojo, sin embargo al comparar esta solución con las pizarras interactivas del mercado se tiene entonces la posibilidad de convertir cualquier superficie plana en una pizarra interactiva, portable y de menor costo. En la imagen 3.12 se muestra un esquema del sistema de la pizarra interactiva usando el Wiimote [30].

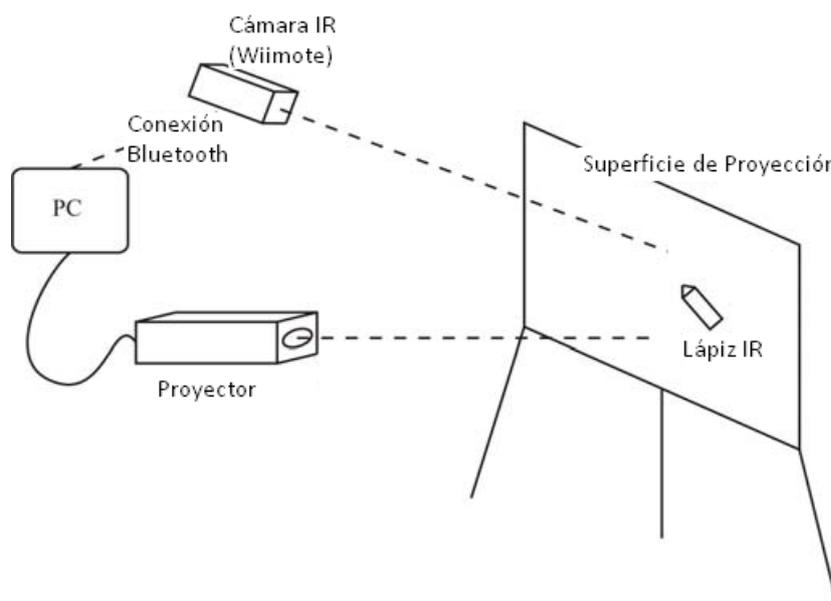


Figura 3.12. Esquema del Sistema Wiiboard.

### Diagrama de secuencia

En la figura 3.13 se puede observar los cuatro procesos importantes realizados por el sistema. El primero corresponde a la captura de la imagen hecha por el Wii. En el segundo se envían las coordenadas de imagen que ya ha calculado el microprocesador del Wii enviando estas coordenadas a través de *Bluetooth* con el computador. En una tercera fase se utilizan los algoritmos que manipulan la librería del Wiimote para recibir estas

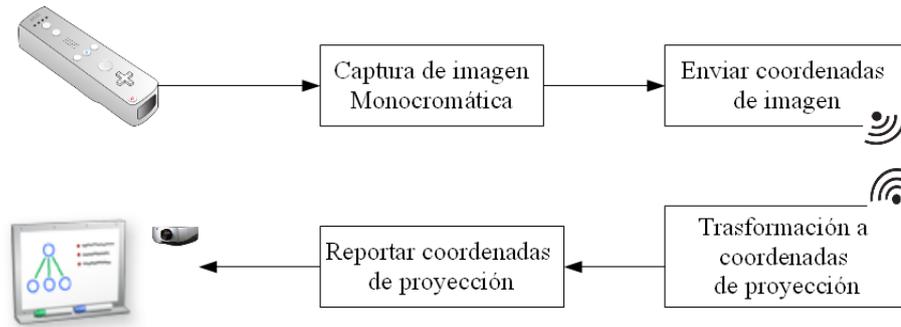


Figura 3.13. Diagrama de secuencia. Sistema Wiiboard

coordenadas y posteriormente transformarlas a las coordenadas de proyección. Al final se reportan las coordenadas de proyección de los nuevos puntos.

### Diseño del software



Figura 3.14. Proceso de Calibración del programa Wiiboard.

Para el Wiiboard, la calibración se realiza con 4 marcas sobre la superficie proyectada, esto se realiza antes de comenzar a trabajar con el sistema, al momento de iniciar el programa, la función de calibración se activa y 4 puntos aparecen en cada una de las esquinas de la imagen a proyectar, estas marcas deben ser señaladas con el lápiz infrarrojo en un orden específico. Una vez la calibración este completa, habrá un número que muestra el porcentaje de seguimiento, este número dice que tan bien ubicado está el Wiimote para rastrear la luz infrarroja del lápiz.

Un porcentaje óptimo debe ser mayor de 50 % con un rango entre 80 % y 90 %. Mientras más alto sea este número, más suave será el trazo de la señal infrarroja. Si el número es realmente bajo se debe ajustar la posición el control Wiimote. Entonces la matriz de transformación se puede determinar utilizando las coordenadas en las marcas de el plano de la imagen proyectada y las coordenadas de las 4 marcas de la imagen captada, que son suministradas por el controlador del Wiimote. Sabiendo que el ángulo de visión de la cámara es de 45°, y la precisión en el seguimiento del lápiz infrarrojo depende de la posición de la cámara contra la superficie de proyección.

La cámara debe estar en una posición que ocupe la mayor área de la imagen proyectada. Mientras más grande sea la superficie utilizada mayor precisión de seguimiento. La relación de superficie útil entre el plano de la imagen de la cámara y la imagen proyectada, puede ser vista como un factor importante durante la primera imagen desplegada al momento de iniciar el programa.

Una funcionalidad importante en este programa es que el algoritmo para suavizado de bordes. Normalmente esto sucede cuando la superficie de proyección, como monitores TFT (*Thin Film Transistor*) y televisores LCD (*Liquid Crystal Display*) es reflectiva, y la luz del lápiz infrarrojo se refleja en la superficie y luego hacia la cámara. Por lo tanto la cámara verá dos puntos, en vez de uno solo. El programa podría escoger la fuente de luz incorrecta, y puede provocar un salto de una fuente de luz a otra fuente de luz que represente el cursor. Además debido a la baja resolución de la cámara comparada con la resolución de la pantalla, ocasiona que una gran área de la superficie de proyección sea asignada a un píxel de la cámara. Por lo tanto, un algoritmo de suavizado de bordes es lo que se utiliza, este encuentra el valor medio de las coordenadas (x,y) de 4 a 20 puntos (configurable) [31]. La cantidad de puntos utilizado en este algoritmo es determinado con un control que está en la configuración del programa Wiiboard que puede apreciarse en la figura 3.15.

### **Materiales necesarios**

- Un Wiimote, mando de la consola Wii de Nintendo
- Un adaptador *bluetooth* USB
- Un puntero que emita luz infrarroja con interruptor

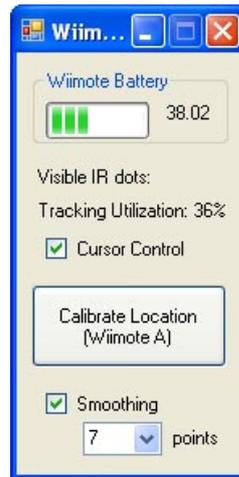


Figura 3.15. Vista del programa Wiiboard.

- El software Wiiboard desarrollado por Johnny Chung Lee, que puede encontrarse en el siguiente enlace: “<http://johnnylee.net/projects/wii/>”.

## Diseño y desarrollo de la solución propuesta

El presente capítulo tiene como objetivo describir el diseño del sistema, la plataforma utilizada, las herramientas y técnicas utilizadas para el desarrollo. Se describe el proceso de desarrollo de la aplicación, así como también las librerías utilizadas. La implementación está basada en las técnicas de procesamiento digital de imágenes, aplicadas a los sistemas de visión por computador.

### 4.1. Plataforma de Hardware

#### 4.1.1. Diseño del Prototipo

El prototipo está compuesto por dos cámaras web, conectadas a un hub USB, y este a un computador por medio de un cable USB; una pizarra acrílica y un video beam para proyectar la imagen sobre ella. Adicionalmente utiliza una lámpara y un borde de color negro en la pizarra para garantizar un ambiente controlado. En la figura 4.1 se muestra la ubicación de estos elementos.

Las cámaras web utilizadas en el prototipo son Microsoft LifeCam VX-7000, las cuales tiene las siguientes características [34]:

- **Resolución:** hasta 1600x1190 píxeles.
- **Transferencia máxima:** 30 imágenes por segundo (fps). Esta transferencia puede variar de acuerdo a la iluminación del entorno, la resolución de captura y otros factores.
- **Sensor:** tecnología CMOS.
- **Conexión:** puerto USB 2.0.
- **Requerimientos mínimos del sistema:** Procesador Intel Pentium 4 2.4 GHz. 256 MB RAM. USB 1.1. Microsoft LifeCam Software.

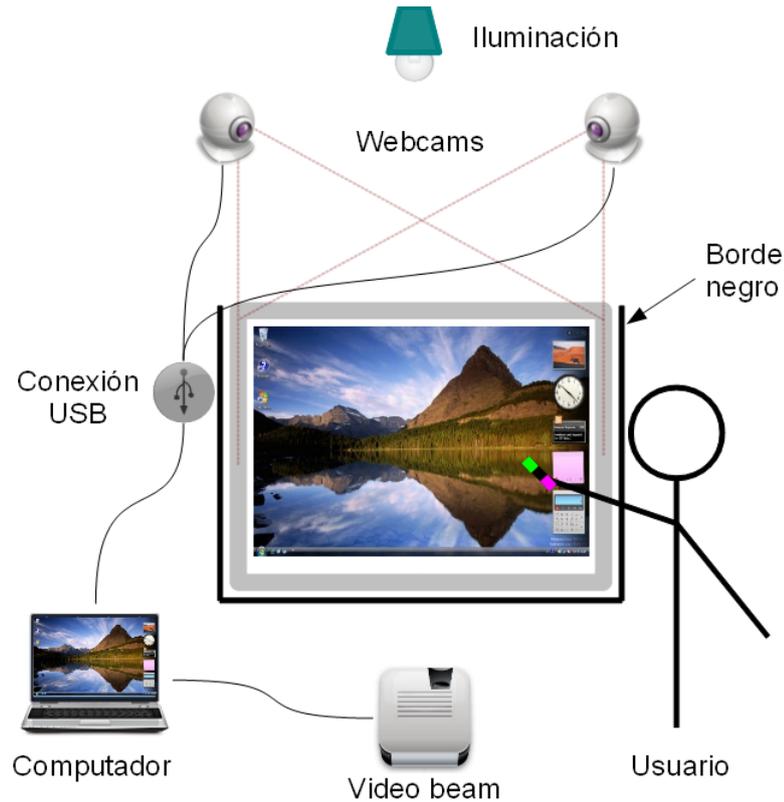


Figura 4.1. Diseño del prototipo.

El costo total aproximado de la pizarra interactiva es de \$215 (sin incluir el costo del video beam) para septiembre de 2010. El prototipo está conformado por dos webcams, una pizarra de 80x60 cms, un bombillo fluorescente de 9w, un hub con extensión USB y los materiales necesarios construir un mueble para fijar los elementos del sistema. La figura 4.2 muestra el prototipo construido utilizando los materiales anteriormente mencionados.

Para realizar la captura de las imágenes se ubican las cámaras en la parte superior de la pizarra, de tal forma que tengan visión sobre toda el área de la pizarra y a su vez no sean afectadas por diferentes fuentes de luz.

Debido a la posición donde se ubican las cámaras solamente es necesario procesar una porción de las imágenes, ya que en el resto de la imagen no se presenta ningún movimiento de interés para el sistema. Un ejemplo de la visión que tiene la cámara y el área que procesa se puede observar en la figura 4.3.

Para determinar la distancia de las cámaras con respecto a la pizarra ( $x$ ) se utilizan las siguientes fórmulas:

$$h^2 = x^2 + w^2 \Rightarrow x = \sqrt{h^2 - w^2}$$



Figura 4.2. Prototipo construido.

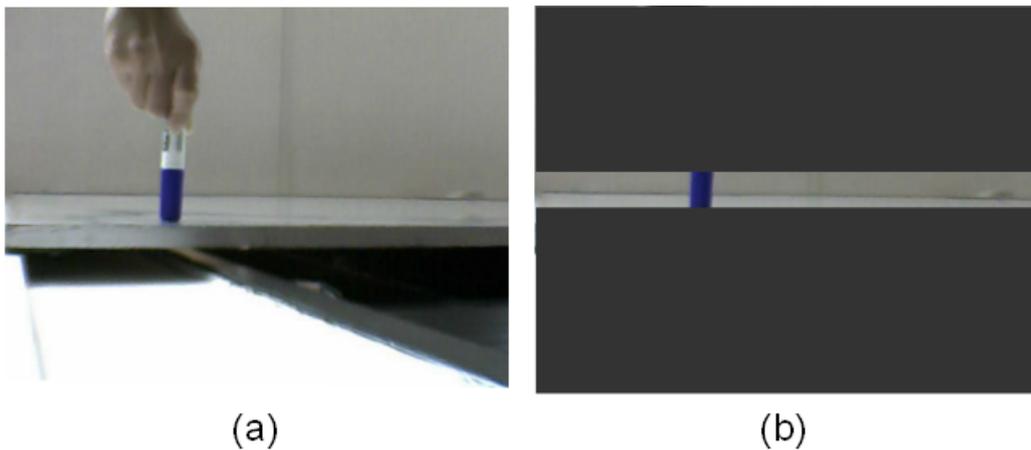


Figura 4.3. Área de la imagen a ser procesada. (a) Imagen original capturada por la cámara. (b) Área de la imagen a ser procesada.

$$\sin(\alpha) = \frac{w}{h} \Rightarrow h = \frac{w}{\sin(\alpha)}$$

Como se muestra en la figura 4.4,  $\alpha$  es el ángulo de visión de las cámaras,  $w$  es el ancho de la pizarra; para la cámara de la izquierda:  $x$  es la distancia con respecto a la esquina superior izquierda de la pizarra y  $h$  es la distancia con respecto a la esquina superior derecha. Análogamente para la cámara de la derecha.

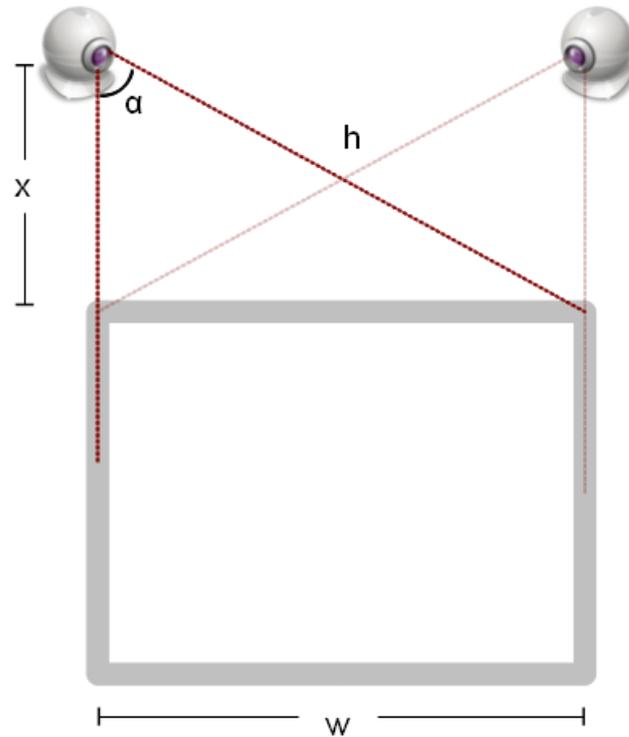


Figura 4.4. Distancia de las cámaras a la pizarra.

En el prototipo construido, se tiene una pizarra de 80x60 centímetros y una webcam con un ángulo de visión de  $58.999475^\circ$ ; para estas medidas las cámaras necesitan estar ubicadas a una distancia  $x$  de 48.06 cms para capturar la pizarra completa.

## 4.2. Plataforma de Software

En las siguientes secciones se describe el lenguaje de programación, las librerías utilizadas, el diseño de la solución, así como también las pruebas realizadas durante el desarrollo del sistema para tomar decisiones sobre cuáles técnicas de procesamiento de imágenes se debían utilizar y cuáles eran las mejores condiciones de ambiente.

### 4.2.1. Lenguaje de programación

El lenguaje de programación utilizado para implementar la aplicación fue C#, que es un lenguaje de programación diseñado para construir aplicaciones que se ejecutan sobre *.NET Framework*, este lenguaje es simple, orientado a objetos y permite desarrollar aplicaciones rápidamente [35]. Además, el uso de C# permite el uso de la librería EmguCV, que facilita el desarrollo de aplicaciones de visión por computador.

La aplicación fue desarrollada utilizando el entorno de desarrollo integrado Microsoft Visual Studio 2008.

### 4.2.2. Librerías de desarrollo

La principal librería utilizada para el desarrollo del sistema fue Emgu CV, que utiliza OpenCV, ambas librerías son explicadas a continuación.

#### OpenCV

OpenCV (*Open source Computer Vision library*) es una librería de código abierto desarrollada por Intel, utilizada para abordar problemas de visión por computador. La librería está escrita en el lenguaje C y C++ y puede ser ejecutada bajo sistemas operativos como Linux, Windows y Mac OS X [32].

Es compatible con IPL (*Intel Processing Library*) de Intel, la cual implementa operaciones de bajo nivel en imágenes digitales, como binarización y filtrado; y de alto nivel como calibración, detección de características, análisis de forma, análisis de movimiento, reconstrucción 3D, segmentación de objetos y reconocimiento entre otros. OpenCV contiene acerca de 500 funciones que abarcan muchas áreas en visión, como control de calidad, imágenes médicas, seguridad, interfaz de usuario, calibración de cámara, visión estéreo y robótica[10].

La librería OpenCV fue diseñada para obtener la mayor eficiencia computacional, con un enfoque en el desarrollo de aplicaciones en tiempo real y con múltiples procesadores.

#### Emgu CV

Emgu CV es un contenedor multiplataforma de la librería OpenCV desarrollada bajo .Net, que permite llamar a las funciones de OpenCV desde cualquier aplicación .Net, lo

que lo hace compatible con lenguajes como C#, Visual Basic, Visual y C++. Este wrapper puede ser compilado de forma Mono (Monodevelop) permitiendo ejecutarse en cualquier plataforma que soporte Mono, incluyendo Linux/Solaris y Mac OS X [33].

### 4.3. Diseño de la aplicación

Para realizar el diseño de nuestra solución se tomaron las etapas de un sistema de visión por computador explicadas en la sección 2.2, y se adaptaron a los requerimientos de nuestro sistema, el resultado lo podemos observar en la figura 4.5. A continuación se describen las etapas diseñadas:

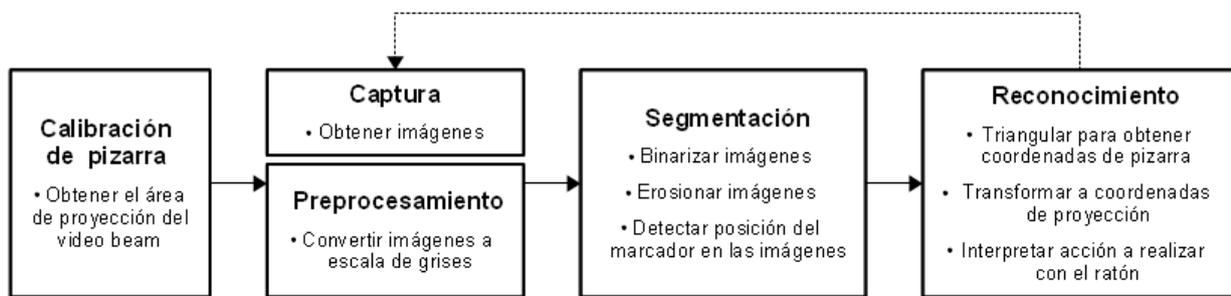


Figura 4.5. Etapas del sistema de visión por computador.

- **Calibración de pizarra:** se realiza para obtener el área de la pizarra sobre la que se proyecta la imagen del computador.
- **Captura:** en esta etapa se obtienen las imágenes a ser procesadas.
- **Preprocesamiento:** consiste en transformar las imágenes a escala de grises para realizar el procesado posterior de las mismas.
- **Segmentación:** consiste en aislar los elementos de interés que existen en las imágenes, en el caso de la pizarra interactiva nos interesa detectar el apuntador. Para ello se pueden aplicar técnicas de detección de movimiento, como imagen diferencial o sustracción de fondo. En el caso de tener condiciones controladas se puede hacer simplemente umbralizando la imagen. También se puede aplicar erosión para eliminar el ruido. Luego de detectar el apuntador es importante extraer el centro del mismo para hacer la triangulación en la etapa siguiente.

- **Reconocimiento:** en esta etapa se toma el centro del apuntador obtenida en la etapa anterior para realizar la triangulación. Al ejecutar la triangulación se calculan las coordenadas de pizarra donde se encuentra el apuntador. Las coordenadas de pizarra son transformadas en coordenadas de proyección (utilizando los datos obtenidos durante la calibración de la pizarra). Finalmente estos datos son interpretados para ejecutar la acción correspondiente con el apuntador.

En la siguiente sección se explican las pruebas que se aplicaron para determinar cuáles eran las mejores técnicas y condiciones a ser utilizadas en este sistema.

## 4.4. Ejecución de la aplicación

A continuación se explican detalladamente las etapas de ejecución del sistema.

### 4.4.1. Calibración de la pizarra

Este módulo se encarga de detectar el área de la pizarra sobre la cual el video beam proyecta la imagen del computador. Es la etapa que se ejecuta al iniciar la aplicación, sin embargo puede ser realizada nuevamente en caso de que sea necesario reajustar el área de proyección. En las siguientes secciones se explican todas las etapas del módulo de calibración de la pizarra.

#### Detección de cámara derecha

Para calcular correctamente las coordenadas de la pizarra donde se encuentra el apuntador es necesario determinar cual es la cámara izquierda y cuál es la derecha. Para ello se colocaron marcas de color (magenta y verde) en lugares extremos de la pizarra, de tal forma que una cámara capte sólo el color verde y la otra el color magenta. De esta forma se puede distinguir cual cámara es la derecha y cual es la izquierda. En la figura 4.7 se puede observar la imagen capturada por la cámara derecha.

#### Inicializar área a procesar de la imagen (rejilla)

Al iniciar la calibración se establece la rejilla de la siguiente manera:

```
rect = new Rectangle(x, y, anchoImagen, altoRejilla);
```

Donde *rect* es un objeto *Rectangle* que se le establece a cada imagen después de ser capturada para que realice el procesamiento de la misma solamente sobre esa área; *x* e *y* son enteros que representan las coordenadas de inicio del área en la imagen; *anchoImagen* es el ancho del área a ser procesada, que en este caso corresponde al ancho de la imagen capturada; y *altoRejilla* es el alto del área de la imagen a ser procesada.

Los parámetros iniciales de la rejilla son previamente cargados de un archivo de configuración que posee los valores por defecto. Estos valores pueden ser cambiados utilizando una aplicación como se puede apreciar en la figura: 4.6 (la figura 4.7 muestra la imagen original de la cámara derecha).

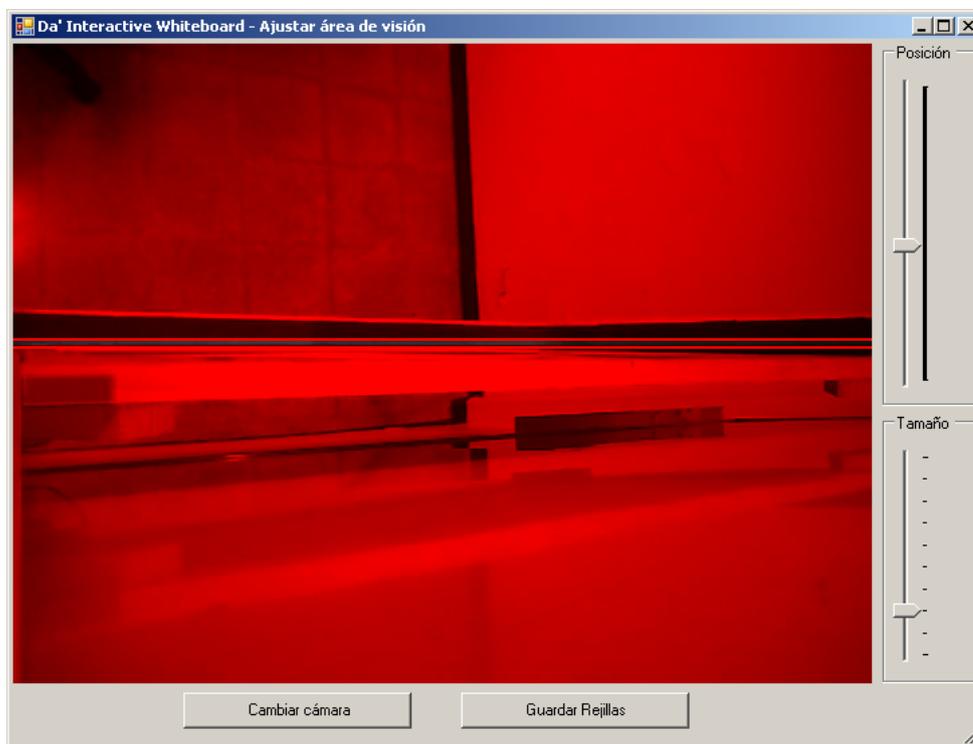


Figura 4.6. Ventana de configuración del sistema (cámara derecha).

### Obtención de las esquinas de proyección

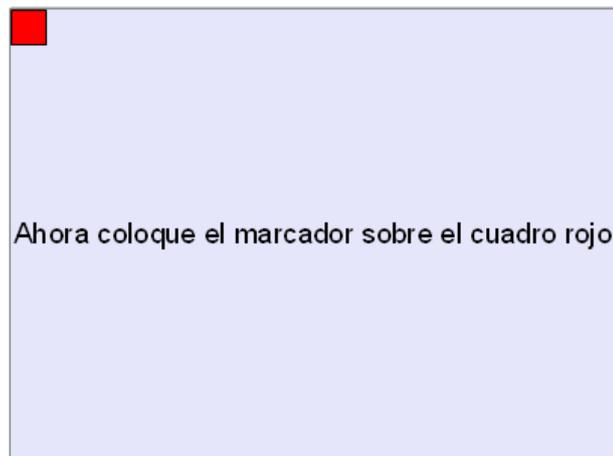
Es importante determinar las esquinas de proyección de la imagen del computador, ya que normalmente éstas no coinciden con las esquinas de la pizarra. Además, al realizar la triangulación se obtienen las coordenadas de la pizarra en centímetros y es necesario llevar esas coordenadas a coordenadas de proyección en píxeles, que dependen de la resolución



**Figura 4.7.** Imagen desde la cámara derecha.

a la cual está proyectando el computador. En esta etapa se toman los datos necesarios para realizar esta transformación.

Para conocer el área de la pizarra sobre la que se está proyectando la imagen del computador se muestra una ventana en pantalla completa dibujando un cuadro rojo en cada esquina de la proyección, esta ventana se puede observar en la 4.8.



**Figura 4.8.** Ventana de calibración de sistema.

El cuadro rojo inicialmente aparece en la esquina superior izquierda de proyección, luego de que el sistema detecta el apuntador, se muestra en las otra esquina hasta que se ha mostrado en todas.

Cada una de las coordenadas calculadas cuando el usuario sitúa el apuntador sobre los cuadros rojos sirven para inicializar una matriz que se utiliza para realizar la transformación de coordenadas de pizarra a coordenadas de proyección. Este proceso se hace automáticamente a través de una función que ofrece OpenCV como se muestra a continuación:

```
PointF[] srcCoord = new PointF[4], dstCoord = new PointF[4];
Matrix<Double> matTrf = new Matrix<Double>(3, 3);
...
```

```
//Se inicializan los arreglos de puntos
...
    CvInvoke.cvGetPerspectiveTransform(srcQuad, dstQuad, matTrf);
```

Donde *srcCoord* es un arreglo de *PointF* que contiene las coordenadas de pizarra obtenidas cuando el usuario coloca el apuntador sobre los cuadros rojos; *dstCoord* es un arreglo de *PointF* que contiene las coordenadas de las esquinas de proyección donde fueron dibujados los cuadros rojos en pantalla; *matTrf* es la matriz de transformación; *CvInvoke* permite invocar funciones de la librería *OpenCV* desde *Emgu CV* y *cvGetPerspectiveTransform* inicializa la matriz de transformación *matTrf*, dadas las coordenadas origen *srcCoord* (coordenadas de pizarra) y las coordenadas destino *dstCoord* (coordenadas de proyección).

Después de este proceso se cierra la ventana de calibración y se inicia la barra de herramientas que será descrita en la siguiente sección.

### Barra de herramientas

La barra de herramientas mostrada en la figura 4.9 le permite al usuario realizar diversas acciones son explicadas con mayor detalle a continuación.



Figura 4.9. Barra de herramientas.

- **Click derecho:**

Este botón emula la función del click derecho sobre el ratón. Una vez pulsado el botón en el software, podemos realizar click derecho en cualquier zona de la pizarra.

**Herramienta de dibujo:** Esta función despliega la herramienta básica de dibujo que posee el sistema operativo, en este caso inicia el Paint en el caso de Windows.

- **Calibración de pizarra:** Esta función permite calibrar el sistema nuevamente.

### 4.4.2. Captura

Por medio funciones que posee Emgu CV, se establecen los parámetros para la captura de la imagen, en este caso la resolución de la imagen a capturar (640x480 pixeles):

```
Capture cam; // objeto de tipo Capture.
cam.SetCaptureProperty(Emgu.CV.CvEnum.CAP_PROP.CV_CAP_PROP_FRAME_WIDTH, 640.0);
cam.SetCaptureProperty(Emgu.CV.CvEnum.CAP_PROP.CV_CAP_PROP_FRAME_HEIGHT, 480.0);
```

Posteriormente por medio del método *QueryFrame* se obtienen las imágenes que luego serán procesadas.

```
imgColor = cam.QueryFrame(); // captura de la imagen de la cámara.
```

### 4.4.3. Preprocesamiento

Las imágenes son obtenidas a color y luego estas son transformadas a escala de grises por medio del siguiente método *Convert<Gray, Byte>()*:

```
Image<Gray, Byte> imgGris; // imagen gris.
imgGris = imgColor.Convert<Gray, Byte>(); // convierte la imagen a escala de grises.
```

### 4.4.4. Segmentación

#### Binarización de imágenes

Las imágenes son binarizadas bajo un umbral previamente establecido:

```
Image<Gray, Byte> imgUmbralizada;

Gray blanco= new Gray(255); // intensidad máxima.
Gray umbral= new Gray(100); // umbral

imgUmbralizada = imgGris.ThresholdBinary(umbral, blanco);
```

El método *ThresholdBinary* umbraliza la imagen de acuerdo a un umbral y un valor máximo, en este caso el umbral es 100 y el valor máximo es 255. Para todos los valores que estén por encima del umbral tomarán el valor máximo y por debajo de este tomarán el valor 0 por defecto.

#### Erosión de imágenes

Luego de aplicar umbralización, se aplica erosión a las imágenes para disminuir el ruido que puede estar presente en la imagen.

```
imgUmbralizada._Erode(1);
```

El método `_Erode` aplica erosión sobre la imagen que la invoca, con un elemento estructurante de 3x3, el parámetro 1 indica la cantidad de veces que va a realizar la erosión.

## Detectar posición del apuntador en las imágenes

Para determinar la posición del apuntador se utiliza un algoritmo de búsqueda que recorre la imagen umbralizada y detecta su ubicación  $(x, y)$  en píxeles. Los algoritmos que explicaremos en la siguiente sección, realizan una búsqueda recorriendo la imagen umbralizada para determinar el centro del apuntador. La figura 4.10 muestra un ejemplo del algoritmo de búsqueda detectando el centro del apuntador.



Figura 4.10. Detección del apuntador en el área de proyección.

- **Algoritmo de búsqueda 1:**

Este algoritmo realiza una búsqueda lineal, partiendo desde la fila central de la imagen y luego recorriendo las filas superiores e inferiores hasta conseguir un píxel de color blanco, siendo este nuestro punto de interés. Dado este punto, se realiza una búsqueda alrededor del píxel para calcular el centro del apuntador.

- **Algoritmo de búsqueda 2:**

Este algoritmo realiza también una búsqueda lineal como el algoritmo anterior, pero saltando las filas de la imagen. Si es encontrado un punto blanco en la imagen, entonces se realiza una búsqueda alrededor para conseguir el centro del apuntador.

- **Algoritmo de búsqueda 3:**

Este algoritmo realiza una búsqueda como el algoritmo anterior, pero saltando filas y columnas de la imagen. Si es encontrado un punto blanco en la imagen, entonces se realiza una búsqueda alrededor para conseguir el centro del apuntador.

- **Algoritmo de búsqueda 4:**

Este algoritmo realiza la búsqueda empezando por la fila inferior de la imagen, hasta encontrar un píxel blanco o hasta recorrer toda la imagen. Si es encontrado un punto

blanco en la imagen, entonces se realiza una búsqueda alrededor de este para conseguir el centro del apuntador.

Como este algoritmo inicia la búsqueda por la parte inferior de la imagen los resultados tienden a ser mas precisos, ya que la parte inferior de la imagen corresponde a la parte mas cercana a la pizarra.

- **Algoritmo de promedio:**

Realiza una búsqueda de píxeles blancos en un área alrededor del pixel encontrado por los algoritmos de búsqueda. Luego calcula un promedio de las coordenadas encontradas para encontrar el centro del marcador.

- **Algoritmo de búsqueda en espiral:**

Es un algoritmo que se ejecuta solamente cuando el apuntador fue conseguido en la iteración anterior. Se efectúa una búsqueda en espiral alrededor de la coordenada donde fue encontrado el marcador en la iteración anterior, de esta manera aumenta la probabilidad de conseguir el marcador rápidamente.

#### 4.4.5. Reconocimiento

##### Triangulación para obtener las coordenadas de pizarra

Después de conseguir la posición del apuntador en las imágenes se aplica el método de triangulación para detectar la posición  $(x, y)$  dentro de las dimensiones de la pizarra (80x60 cms.). En la figura 4.11 se muestra la triangulación aplicada en el sistema, donde  $P$  es el apuntador,  $A$  y  $B$  son las cámaras y  $L$  es la distancia entre las cámaras. En la figura 4.12 se muestra cómo se aplica esta triangulación sobre la pizarra para detectar el marcador, donde:

$$D' = D - 48,06cms$$

$$\alpha' = 90 - \alpha$$

$$\beta' = 90 - \beta$$

$$X = D / \tan(\beta')$$

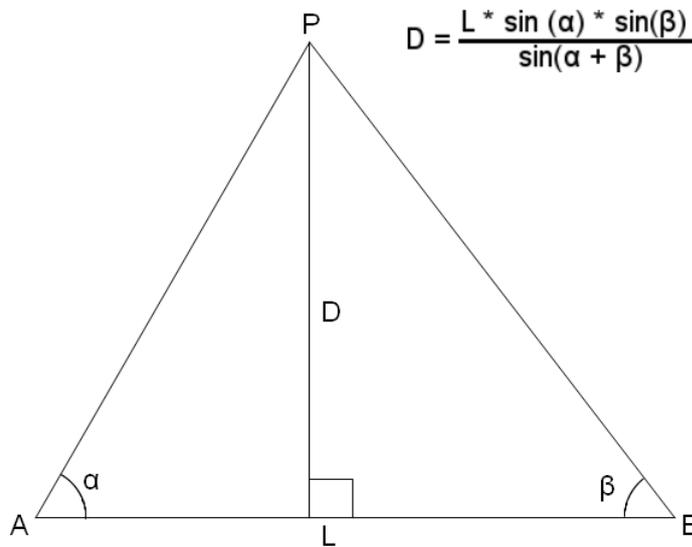


Figura 4.11. Forma de triangulación aplicada.

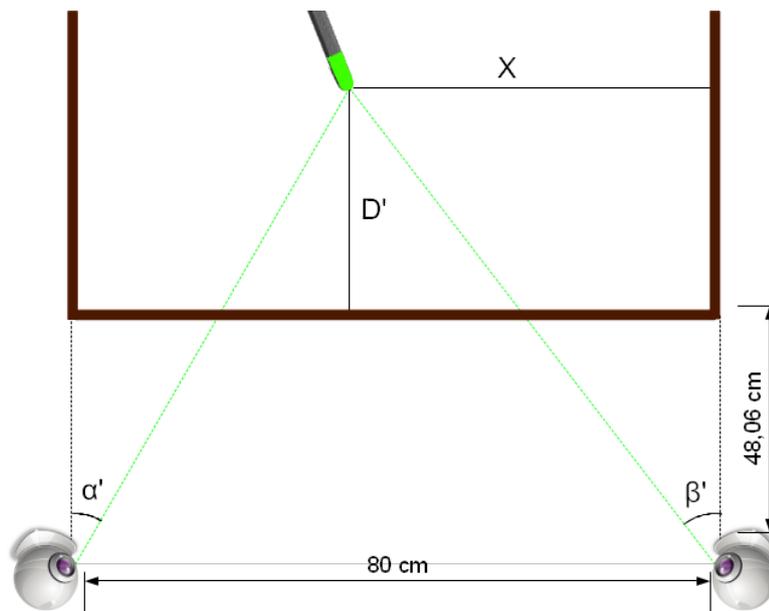


Figura 4.12. Triangulación aplicada a la pizarra.

### Transformar coordenadas de proyección

Luego de obtener las coordenadas  $(x, y)$  de la pizarra, donde se encuentra ubicado el apuntador, estas se transforman a las coordenadas de proyección. Esto fue logrado utilizando la siguiente función de OpenCV, que utiliza la matriz de transformación obtenida en la etapa de calibración:

```
CvInvoke.cvPerspectiveTransform(srcPoint, dstPoint, matTrf);
```

*srcPoint*, contiene las coordenadas  $(x, y)$  de la pizarra. *dstPoint*, es donde quedará almacenada las coordenadas de proyección. *matTrf*, es la matriz de transformación obtenida durante el proceso de calibración.

#### 4.4.6. Intérprete

El intérprete es el módulo que se encarga de descifrar cada una de las acciones que se quieren realizar sobre la pizarra. Son estas acciones las que nos permiten simular el ratón del computador, permitiéndonos hacer click, arrastra el mouse y hacer doble click.

Para facilitar la interpretación de los eventos se utiliza un apuntador que posee dos colores que pueden ser reconocidos por el sistema, el color magenta para hacer click y el color verde para doble click. El apuntador utilizado es mostrado en la figura 4.13.



Figura 4.13. Apuntador.

Es natural cuando manipulamos sistemas interactivos, tener la sensación de hacer click cuando finalmente logramos tocar la pantalla. Para simular esta sensación en nuestro prototipo, es necesario descartar las imágenes sobre las cuales el apuntador aún no ha tocado aún la superficie de la pizarra. Esto se logra descartando los tres primeros frames cuando el apuntador entra en la rejilla. Un ejemplo de los primeros frames cuando el apuntador entra en el área a procesar por el sistema se puede observar en la figura 4.14.

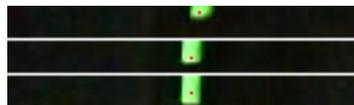


Figura 4.14. Imágenes capturadas cuando el apuntador entra en la rejilla.



## Pruebas y resultados

En las siguientes secciones se muestran las pruebas realizadas durante el desarrollo del sistema y al finalizar el mismo.

### 5.1. Pruebas de desarrollo

En las siguientes secciones se describen las pruebas y resultados de obtener información sobre el ángulo de visión de las cámaras y la calibración de las mismas. Además se describen las pruebas realizadas para definir las técnicas de procesamiento de imágenes y las condiciones de ambiente a utilizar en el sistema final.

#### 5.1.1. Ángulo de visión de la webcam

Esta prueba se realizó con el objetivo de encontrar el ángulo de visión horizontal de la cámara.

Se realizaron varias mediciones como lo indica la figura 5.1 colocando la cámara sobre una línea recta  $H$ , de tal forma que la recta fuese lo último que viera la cámara en su campo de visión por la derecha, luego se marcaban dos puntos  $A$  y  $B$ , tal que esos puntos fuesen lo último que viera la cámara en su campo de visión por la izquierda. Se trazaba una recta  $Ca$  entre los puntos  $A$  y  $B$ , de esta forma la intersección entre  $Ca$  y  $H$  coincidía con el punto focal de la cámara. Por último se trazaba una línea  $Co$ , perpendicular a  $Ca$ , para formar un triángulo rectángulo. Así por trigonometría simple se puede obtener el valor del ángulo de visión de la cámara.

Para realizar esta prueba se realizaron 15 medidas, se calculó el promedio y este fue usado en el sistema, estos datos se muestran en la tabla 5.1.

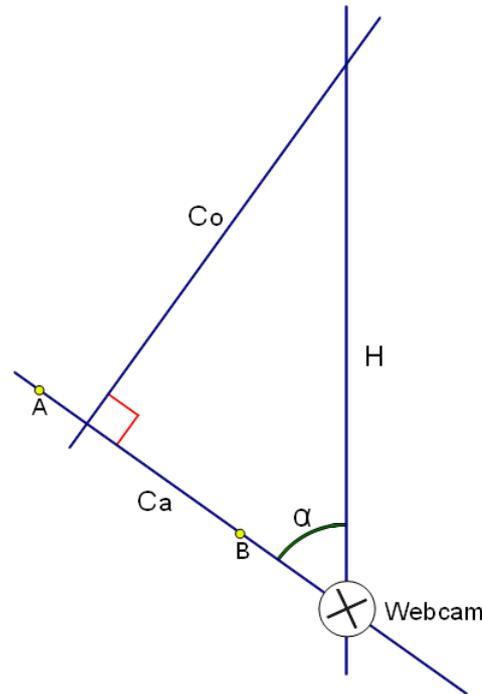


Figura 5.1. Método utilizado para obtener el ángulo de las cámaras.

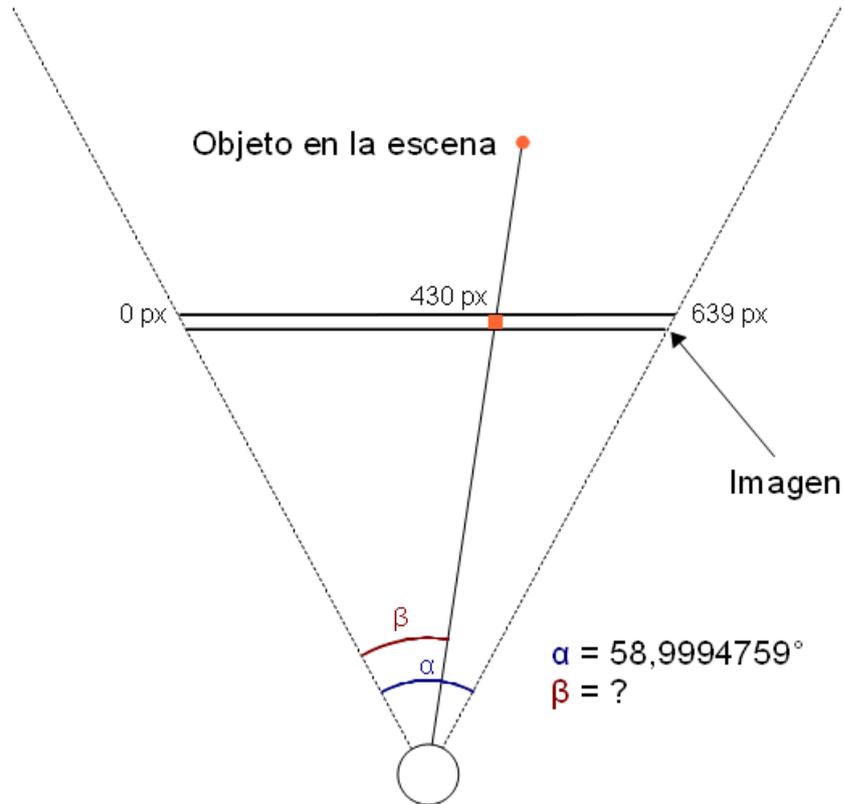
Ángulos
59,14579549
59,35423948
59,07160191
59,14570000
58,78150000
58,91830000
58,73140000
58,87190140
59,02148885
58,81744854
58,88690642
58,83673355
58,90519453
59,17198407
59,33194420
Promedio: 58,99947590

Tabla 5.1. Ángulos medidos.

### 5.1.2. Calibración de la cámara

Se realizaron varias pruebas para establecer la mejor forma de calibrar la cámara con la finalidad de obtener una correspondencia entre los píxeles en el eje horizontal de la imagen y los ángulos de la escena como se muestra en la figura 5.2.

Para probar la calibración de la cámara se simuló la geometría del sistema dibujando sobre papel un rectángulo que representa la pizarra. Luego se se dibujó sobre este una



**Figura 5.2. Correspondencia entre píxel obtenido y el ángulo de la escena.**

En la imagen se observa que un objeto presente en la escena se corresponde con el píxel 430 en el eje X de la imagen, y se desea conocer el ángulo correspondiente a ese píxel.

cuadrícula y se posicionaron las cámaras a la distancia que se utiliza en el sistema real como se muestra en la figura 5.3.

Para elaborar los casos de prueba que se explican a continuación se colocaron objetos cada 10 cms sobre la pizarra utilizando la cuadrícula que simula la pizarra. Estos objetos fueron capturados con las dos webcams en una resolución de 640x480 píxeles. De las imágenes obtenidas se ubicaron los píxeles donde se encontraba el objeto y se guardaron todos estos datos en la tabla 5.2.

En cada caso se realizaba una calibración de la cámara utilizando una técnica de interpolación. Luego se aplicaba la triangulación para cada uno de los objetos y se calculaba la coordenada en la que se encontraba ubicado. Como se conocía previamente la coordenada de la pizarra donde se encontraba el objeto se pudo hacer un cálculo del error para cada una de las calibraciones realizadas.

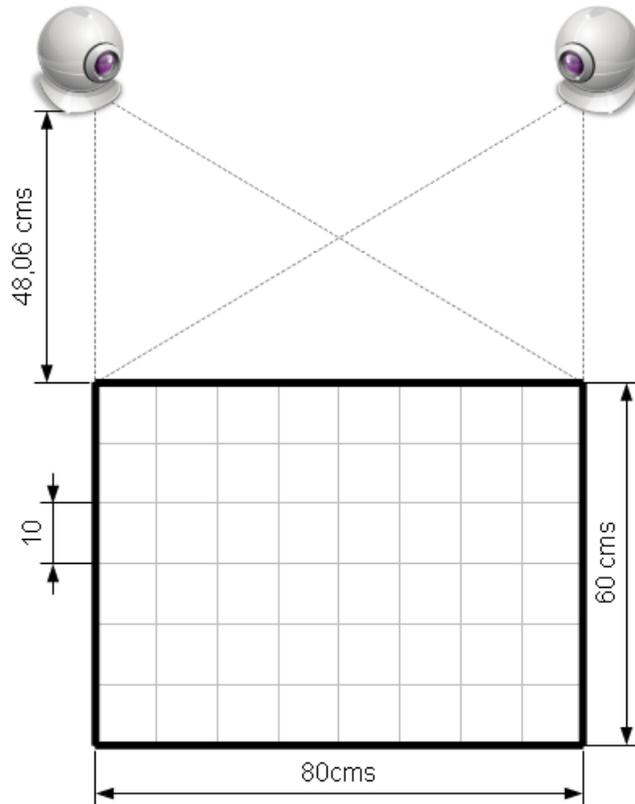


Figura 5.3. Prototipo que simula la geometría del sistema.

### *Caso base*

El caso base consistió en asociar el píxel 0 con el ángulo  $0^\circ$ , y el píxel 639 con el ángulo  $58.9994759^\circ$ , y realizar una interpolación lineal para obtener el resto de los ángulos. La diferencia entre los resultados obtenidos y los resultados esperados se pueden observar en la figura 5.4.

### *Caso 1*

Para este caso se calibró la cámara colocando objetos cada 10 cms en el eje  $y=0$  de la pizarra, se calcularon los ángulos para estos puntos y se capturaron los objetos con la webcam para obtener los píxeles donde estaban ubicados. Luego se llenó una tabla de píxeles con sus respectivos ángulos y se obtuvieron los valores de los ángulos faltantes mediante interpolación lineal. En la figura 5.5 se pueden observar los resultados obtenidos con esta calibración.

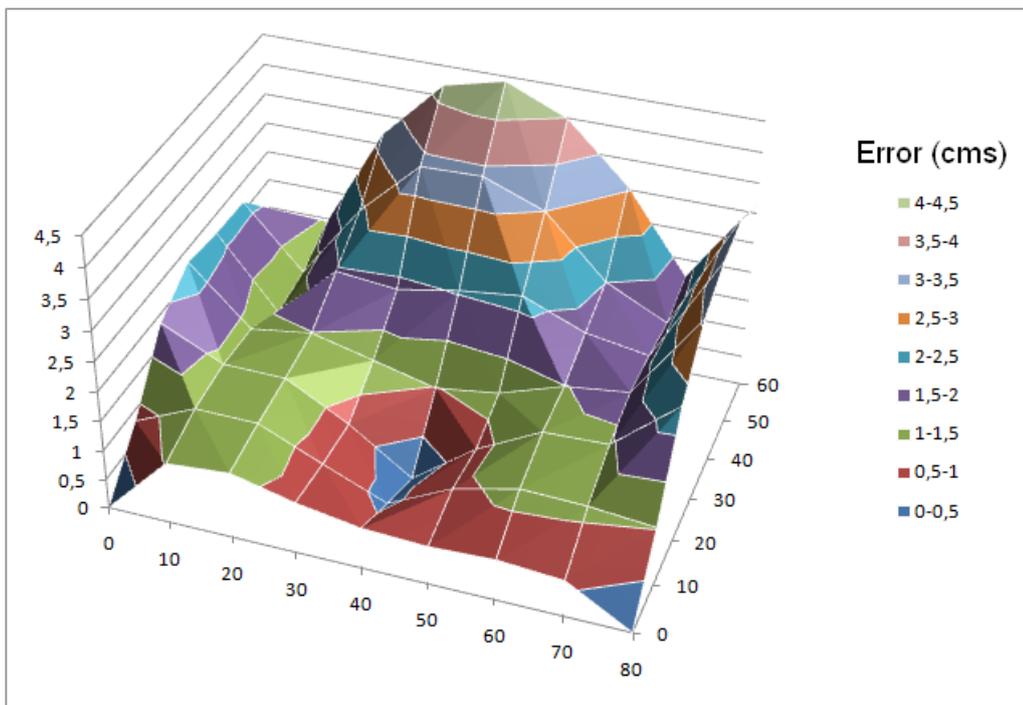


Figura 5.4. Medida del error de calibración. Caso base.

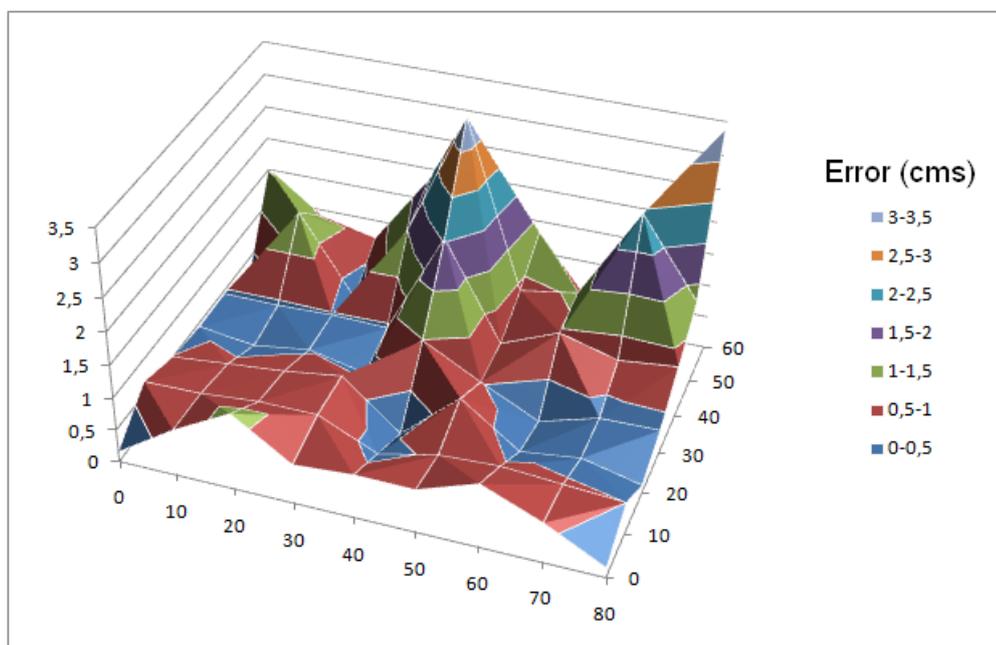


Figura 5.5. Medida del error de calibración. Caso 1.

X	Y	Píxel Derecho	Píxel Izquierdo	X	Y	Píxel Derecho	Píxel Izquierdo
0	0	639	639	50	30	289	236
10	0	499	600	60	30	239	167
20	0	383	549	70	30	195	87
30	0	289	492	80	30	156	0
40	0	211	425	0	40	639	450
50	0	145	346	10	40	562	412
60	0	88	251	20	40	489	369
70	0	39	135	30	40	428	323
80	0	0	0	40	40	371	272
0	10	639	581	50	40	319	215
10	10	521	538	60	40	272	150
20	10	422	490	70	40	231	78
30	10	338	435	80	40	193	0
40	10	266	372	0	50	639	418
50	10	205	300	10	50	569	381
60	10	149	215	20	50	504	341
70	10	101	115	30	50	446	299
80	10	59	0	40	50	393	250
0	20	639	529	50	50	345	196
10	20	538	488	60	50	302	136
20	20	450	442	70	50	262	71
30	20	375	389	80	50	225	0
40	20	309	330	0	60	639	392
50	20	251	264	10	60	576	356
60	20	199	188	20	60	517	318
70	20	152	99	30	60	462	276
80	20	112	0	40	60	413	231
0	30	639	486	50	60	368	181
10	30	551	446	60	60	326	126
20	30	472	402	70	60	289	67
30	30	403	352	80	60	252	0
40	30	343	298				

**Tabla 5.2. Coordenadas de pizarra y píxeles capturados por las cámaras**

X e Y son las coordenadas de la pizarra en centímetros. Píxel derecho y Píxel Izquierdo corresponden a las coordenadas X de una imagen sobre la cual se proyecta un objeto que se encuentra sobre la coordenada de la pizarra.

### *Caso 2*

Para este caso la calibración se realizó dibujando sobre papel una línea recta  $AB$  de 40 cms. en el punto  $B$  se trazó una perpendicular a  $AB$ , se le hicieron marcas cada  $2.5^\circ$  con respecto a  $AB$  y se trazó otra línea recta desde  $A$  hasta  $C$  a  $24.499^\circ$  de  $AB$  (La mitad del ángulo de visión de la cámara) como se muestra en la figura 5.6.

Luego se colocó la cámara en el extremo  $A$  de tal forma que el píxel central coincidiera con un objeto presente en el punto  $B$  y que borde izquierdo de la imagen capturada coincidiera con un objeto colocado sobre el punto  $C$ . Después de que la cámara quedaba bien posicionada se puso el objeto en cada una de las marcas (cada  $2.5^\circ$ ), se tomaron los píxeles correspondientes a cada grado, se obtuvo una tabla de píxeles con sus respectivos

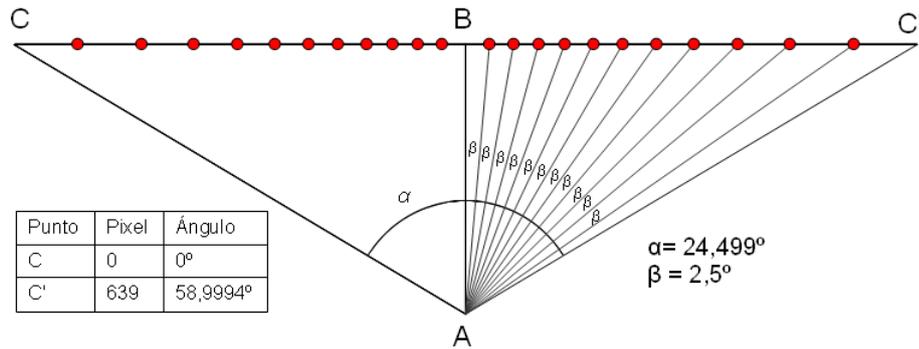


Figura 5.6. Prototipo que simula la geometría del sistema.

ángulos y se interpolaron linealmente los valores faltantes. Los resultados obtenidos se muestra en la figura 5.7.

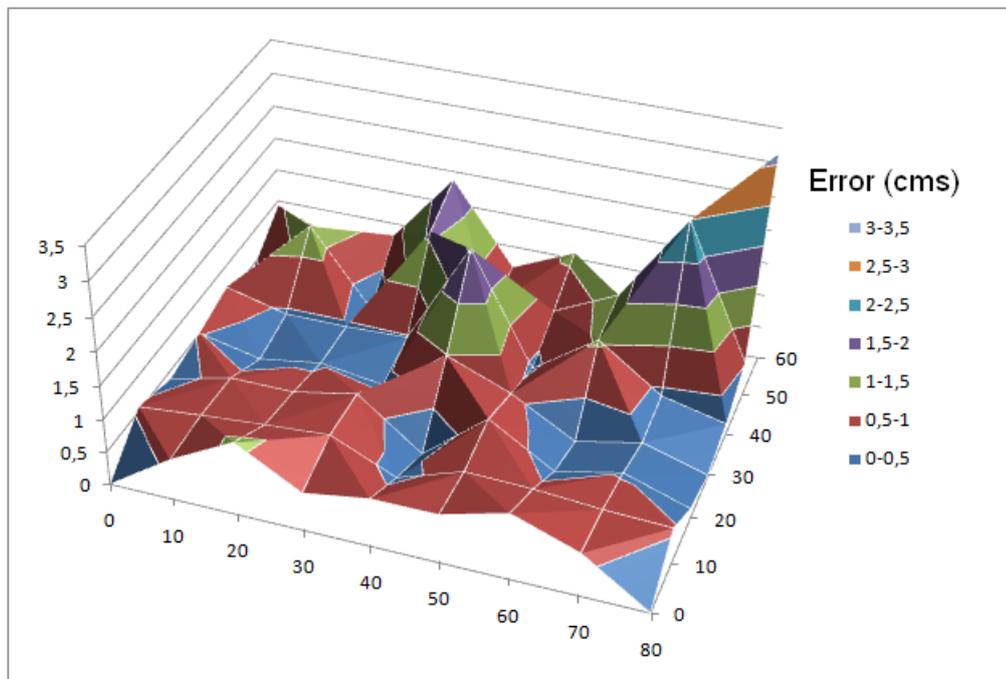


Figura 5.7. Medida del error de calibración. Caso 2.

### Caso 3

Al observar que los resultados obtenidos con la técnica de calibración realizada anteriormente eran buenos se repitió esta calibración colocando la cámara y los objetos que tomamos como referencia de una forma más precisa para ver si se obtenían mejores resultados, al obtener la tabla se interpolaron los datos linealmente y se puede observar el error en la figura 5.8.

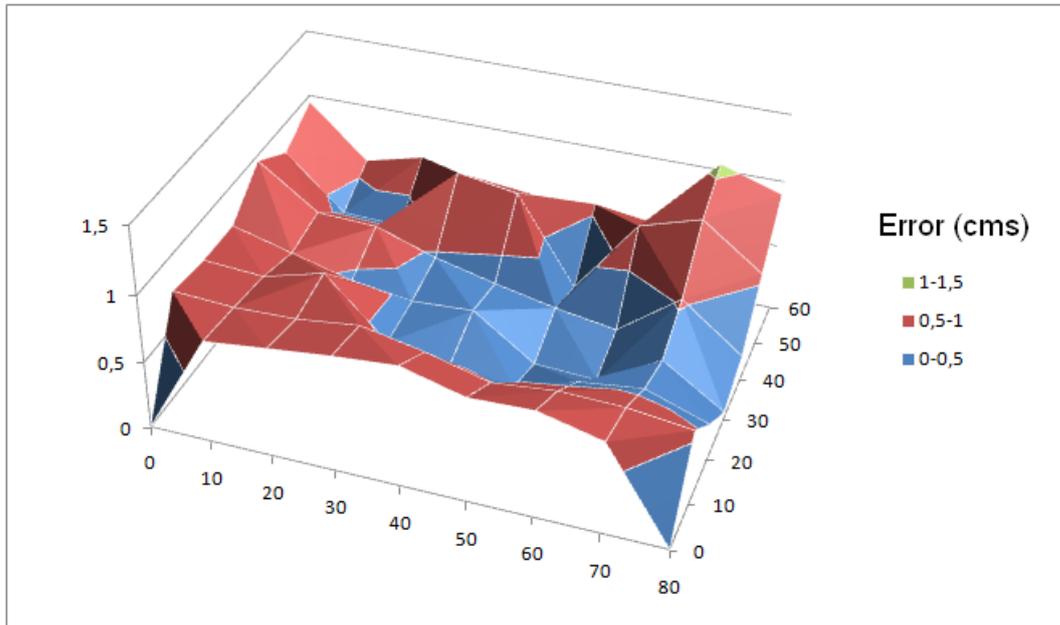


Figura 5.8. Medida del error de calibración. Caso 3 con interpolación lineal.

Luego se interpolaron los mismos datos por spline cúbico, estos resultados se pueden observar en la figura 5.9.

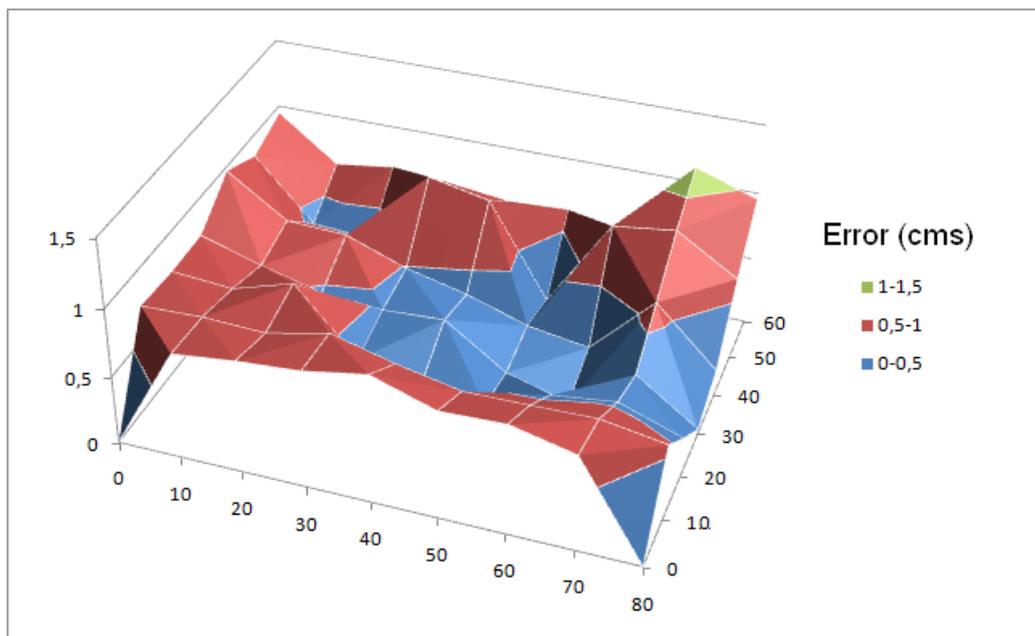
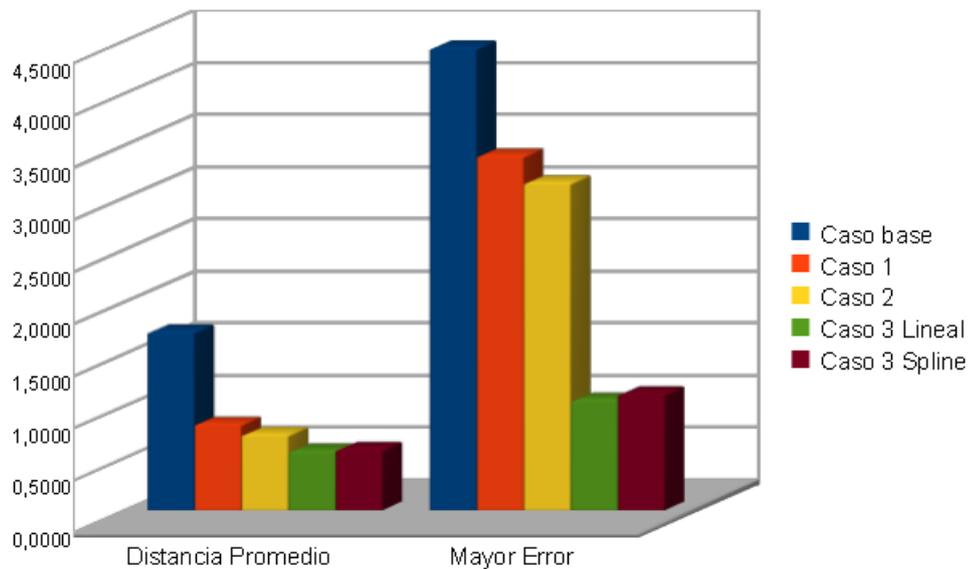


Figura 5.9. Medida del error de calibración. Caso 3 con interpolación por spline cúbico.

### *Resultados de la calibración*

Después de realizar todas las pruebas de calibración explicadas y graficadas anteriormente se decidió tomar los resultados de la calibración del caso 3 con interpolación lineal para utilizarlos en el sistema, ya que esta fue la que obtuvo los mejores resultados. Los comparación de las pruebas de calibración se pueden observar en la figura 5.10.



**Figura 5.10. Medida del error de calibración en todos los casos.**

**Distancia Promedio** corresponde a la distancia promedio (en centímetros) entre las coordenadas obtenidas a través de la triangulación con cada calibración y las coordenadas reales. **Mayor Error** es la mayor distancia obtenida entre las coordenadas obtenidas y las coordenadas reales.

#### **5.1.3. Técnicas de detección de movimiento**

Esta prueba fue realizada con la intención de elegir cual técnica de detección de movimiento resultaba ser mas adecuada para el sistema. Para ello se utilizaron las técnicas de imagen diferencial y sustracción de fondo utilizando un objeto negro sobre un fondo blanco.

Como observamos en la figura 5.11 los mejores resultados se obtuvieron al utilizar sustracción de fondo, ya que al restar imágenes consecutivas la imagen resultante parece tener 2 objetos.

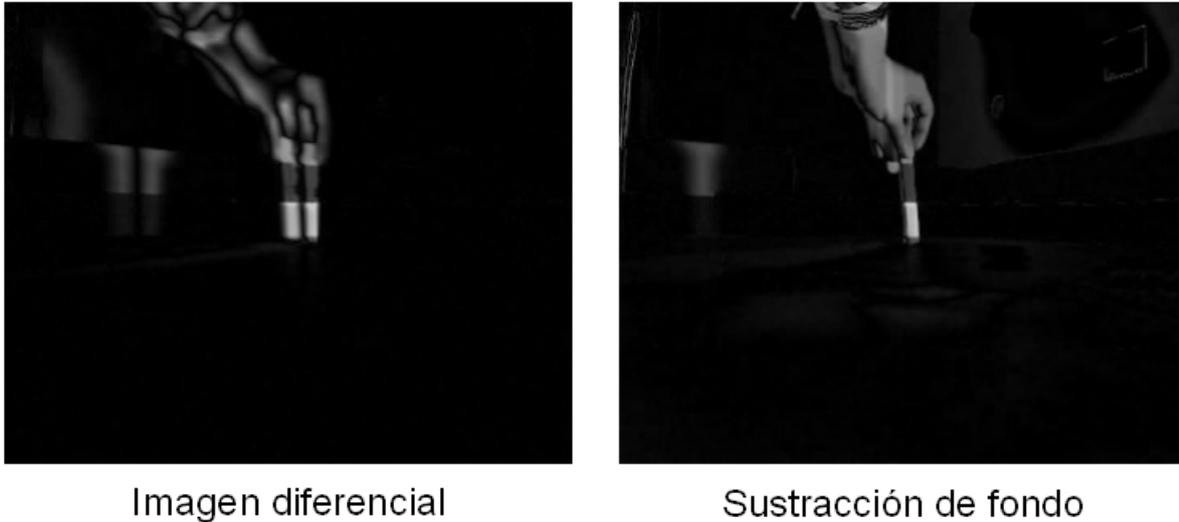


Figura 5.11. Comparación de técnicas de detección de movimiento.

#### 5.1.4. Iluminación y contraste

Se realizaron distintas pruebas de iluminación donde se variaron las condiciones del sistema hasta asegurar un ambiente controlado. Para realizar las pruebas se colocaron fondos de colores alrededor de la pizarra, que fuesen contrastantes con el objeto de interés para facilitar su búsqueda. Además se probaron con diferentes fuentes de iluminación.

#### Caso experimental usando fondo blanco

En estas pruebas se usó fondo blanco como color de contraste, con distintas condiciones de iluminación utilizando luz artificial en la parte superior de la pizarra. Para el tratamiento de las imágenes se aplicaron técnicas de procesamiento de imagen como: sustracción del fondo y umbralización.

En la figura 5.12, se muestra una secuencia de imágenes obtenida desde una cámara, donde no se utilizó ninguna fuente de iluminación adicional y se colocó un objeto de interés de color negro. Se puede observar existencia de ruido en un umbral bajo y al aumentar el umbral se elimina parcialmente, pero el objeto de interés deja de percibirse.

En las figuras 5.13 y 5.14 se colocaron 3 objetos de interés en diferentes áreas, variando la intensidad de la iluminación para observar el comportamiento del umbral.

Como se puede observar en la figura 5.13, donde se utilizaron fuentes de iluminación de alta intensidad, la existencia de sombra incluso utilizando umbrales altos (ver imagen 5.13: umbral de 50), mientras que en la figura 5.14, en la que se utilizaron bombillos de poca

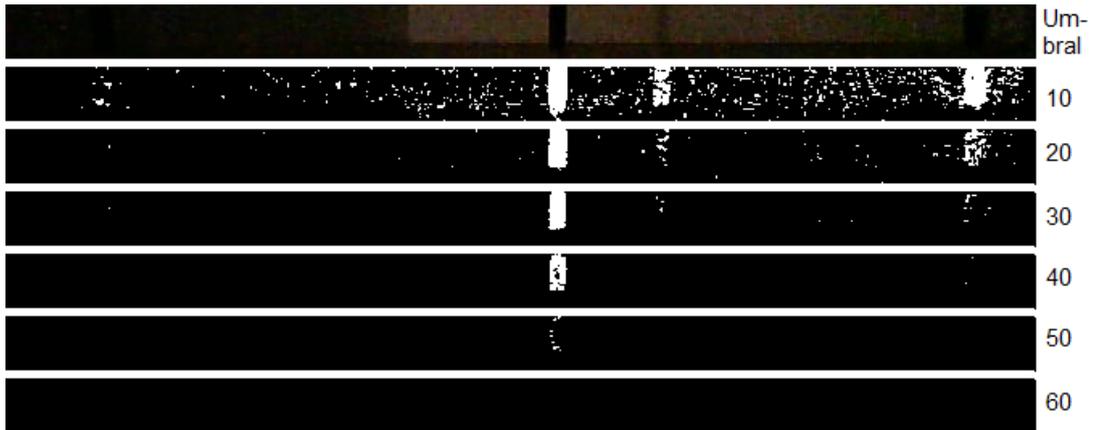


Figura 5.12. Fondo blanco sin fuente de iluminación artificial. Operaciones aplicadas: Sustracción de fondo y umbralización(umbral:10-60)

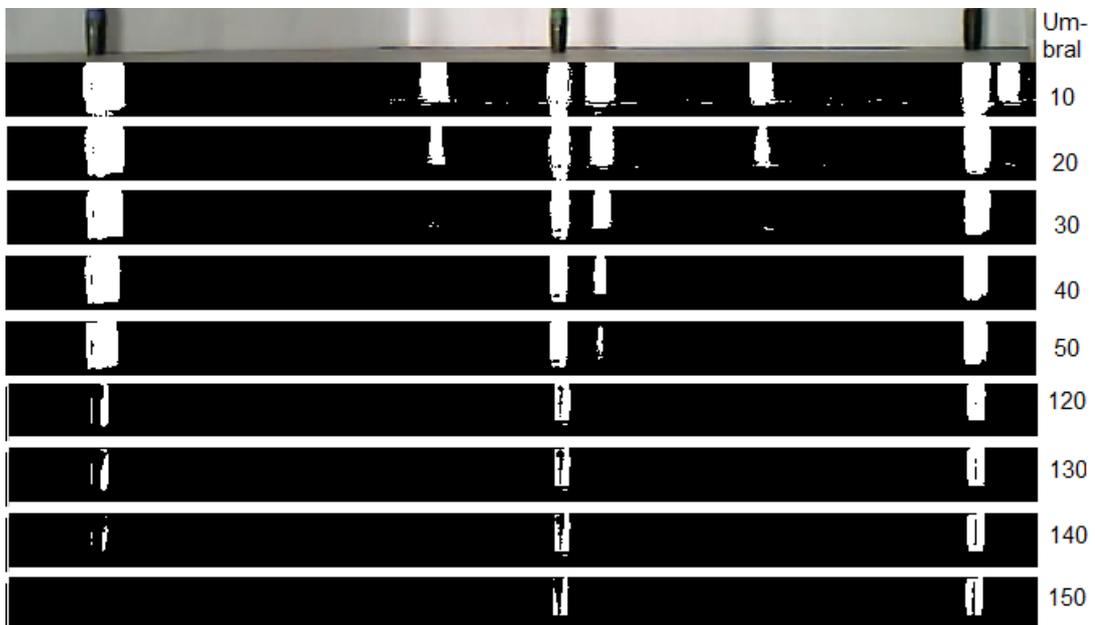


Figura 5.13. Fondo blanco, fuentes de iluminación de alta intensidad. Operaciones aplicadas: sustracción de fondo y umbralización (umbral:10-150).

intensidad, la sombra deja de percibirse en un umbral menor (ver figura 5.14: umbral de 40), otro beneficio de utilizar una baja intensidad es que disminuye el reflejo de luz sobre el objeto de interés, aumentando la facilidad en la localización. Bajo las condiciones de fondo blanco se determinó que usar fuentes de iluminación con baja intensidad permiten obtener un mejor resultado.

### Caso experimental usando fondo verde

Para este caso experimental se usó un fondo verde de alto contraste, y un objeto de interés de color negro. Adicionalmente se aplicaron técnicas de umbralización.

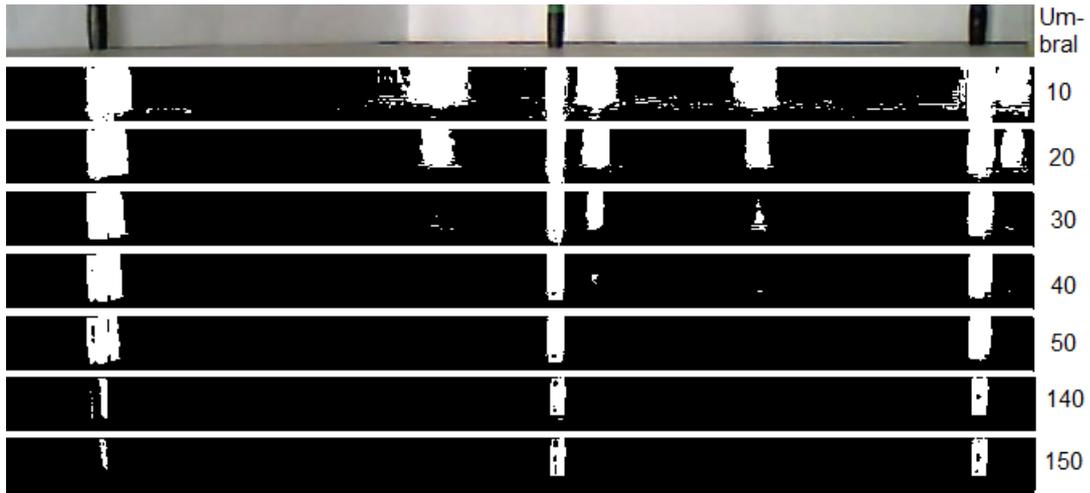


Figura 5.14. Fondo blanco, fuentes de iluminación de baja intensidad. Operaciones aplicadas: sustracción de fondo y umbralización (umbral:10-150).

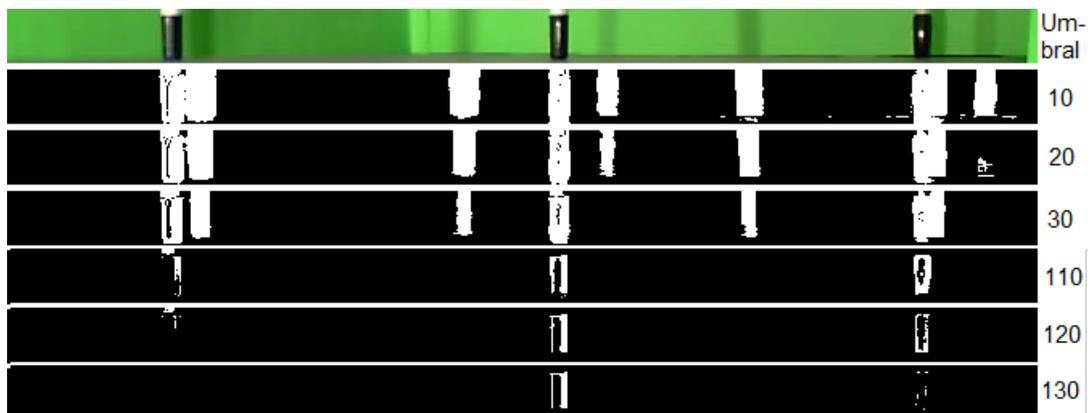


Figura 5.15. Fondo verde, fuentes de iluminación de alta intensidad. Operaciones aplicadas: sustracción de fondo y umbralización (umbral:10-130).



Figura 5.16. Fondo verde, fuentes de iluminación de baja intensidad. Operaciones aplicadas: sustracción de fondo y umbralización (umbral:10-90).

Las figuras 5.15 y 5.16 muestran 3 objetos de interés en diferentes áreas, se puede notar una leve mejora con respecto al fondo blanco, como se puede observar en la figura 5.16: umbral 10, 20 y 30) donde las sombras eran más tenues.

### Caso experimental usando fondo negro

Para estas pruebas se utilizó un fondo de color negro, y un objeto de interés de alto contraste. Adicionalmente se aplicaron técnicas de umbralización y erosión.



Figura 5.17. Fondo negro, fuentes de iluminación de alta intensidad. Operaciones aplicadas: sustracción de fondo y umbralización (umbral:10-100).



Figura 5.18. Fondo negro, fuentes de iluminación de baja intensidad. Operaciones aplicadas: sustracción de fondo y umbralización (umbral:10-120).

Los resultados obtenidos fueron los mejores en base a los dos experimentos anteriores, el problema de las sombras quedó resuelto debido que el color negro tiene la propiedad de absorber la luz anulando las sombras generadas por el apuntador en cualquier sección de la pizarra. La figura 5.17 posee una fuente de iluminación de alta intensidad, se le aplicaron

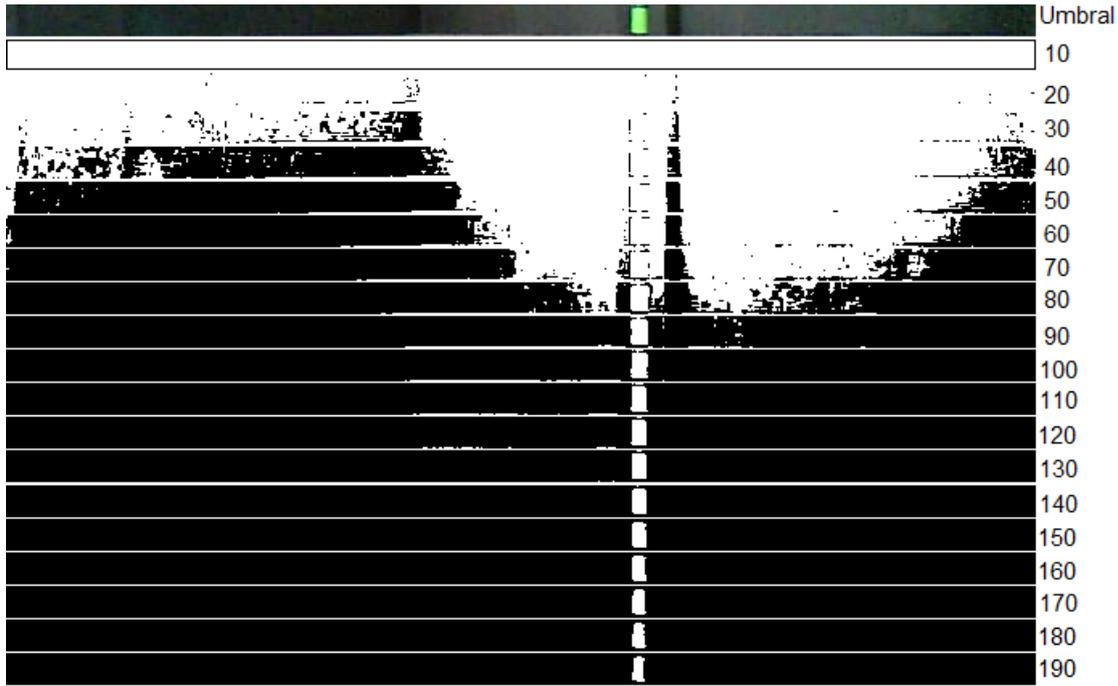


Figura 5.19. Fondo negro, fuentes de iluminación de baja intensidad. Operaciones aplicadas: umbralización (umbral:10-100).

técnicas de sustracción de fondo y umbralización, mientras que las figuras 5.18 y 5.19 usan una fuente de iluminación baja. Ambos resultados fueron similares, dado que la sombra no se refleja en el fondo, por lo que cualquiera de las opciones es viable. Sin embargo para un mejor rendimiento en el sistema se usará la opción que no posee sustracción de fondo ya que obtiene un resultado similar y se ahorra tiempo de cómputo.

### 5.1.5. Iluminación y contraste en movimiento

Las pruebas de movimiento permiten determinar el ajuste necesario para la ubicación del objeto de interés cuando este se encuentra en movimiento. Estos ajustes pueden ser de umbral, erosión, iluminación o área de procesamiento de las cámaras.

En esta sección se muestran una secuencia de imágenes donde se encuentra el objeto de interés en movimiento sobre distintos sectores de la pizarra, y con un algoritmo de detección se determina su posición en la imagen. Estas imágenes muestran el objeto de interés deslizándose sobre la pizarra, en un principio el desplazamiento es a una velocidad moderada que luego se va incrementando. Los valores de umbralización fueron considerados los que mejor se adaptaban a las condiciones del entorno.

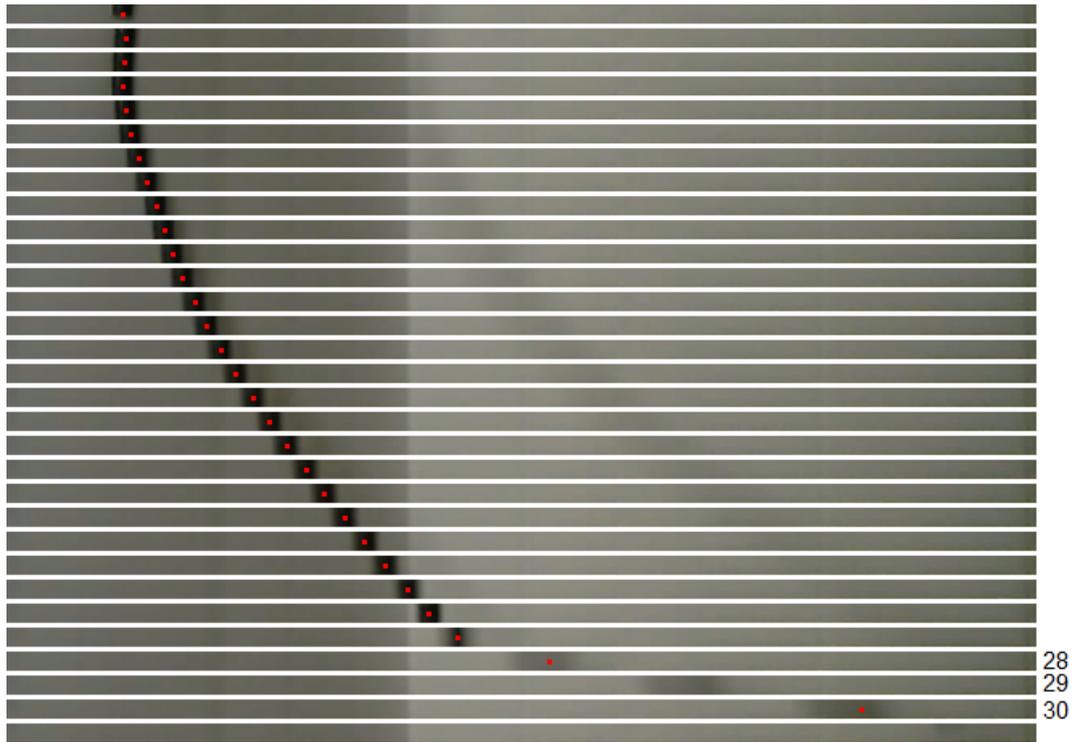


Figura 5.20. Movimiento del apuntador sobre fondo blanco en la parte superior de la pizarra. Imagen a color.

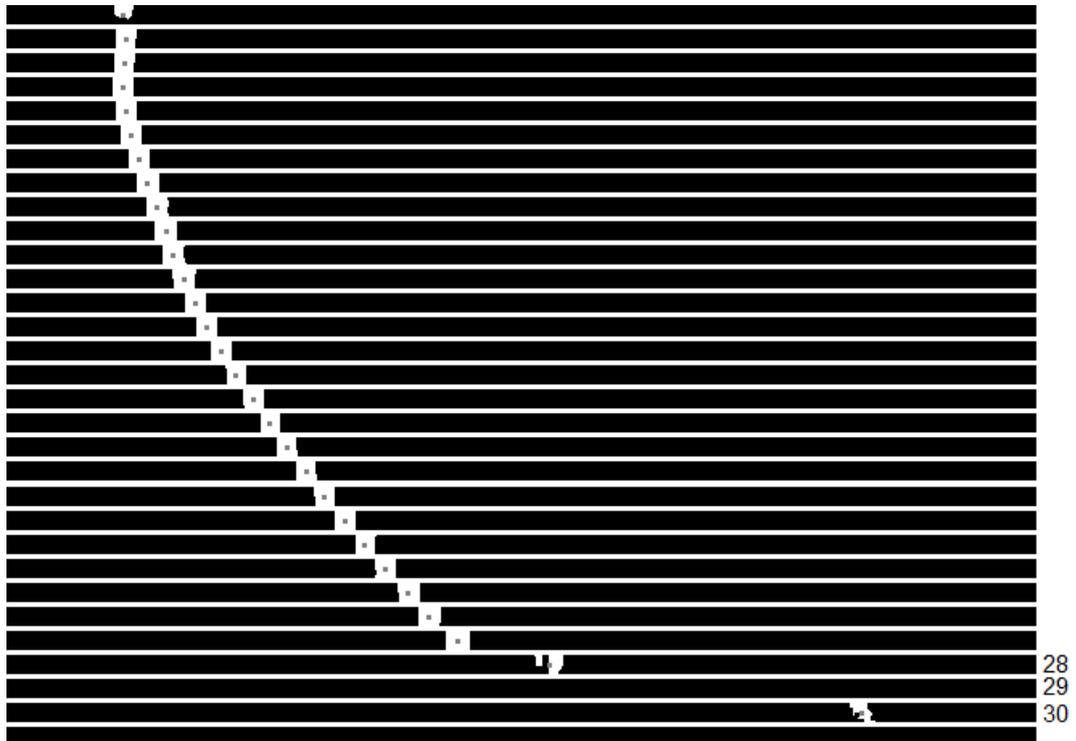


Figura 5.21. Movimiento del apuntador sobre fondo blanco en la parte superior de la pizarra. Imagen umbralizada: umbral 35, erosión 1.

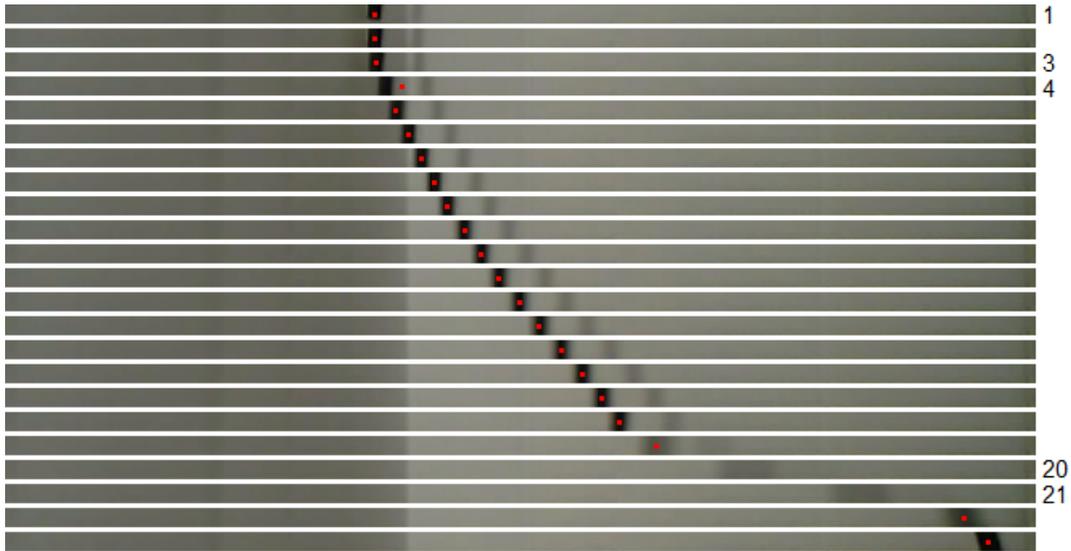


Figura 5.22. Movimiento del apuntador sobre fondo blanco en la parte inferior de la pizarra. Imagen a color.

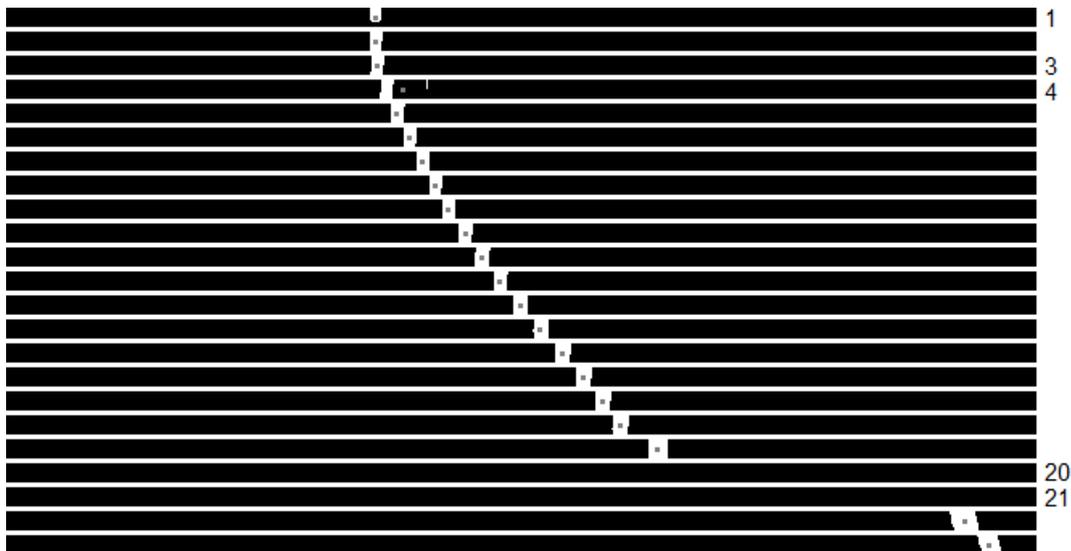


Figura 5.23. Movimiento del apuntador sobre fondo blanco en la parte inferior de la pizarra. Imagen umbralizada: umbral 35, erosión 1.

### Movimiento sobre fondo blanco

La figura 5.20 es la imagen original y la figura 5.21 es la imagen umbralizada. Ambas son el resultado de desplazar el objeto sobre la parte central de la pizarra, las figuras 5.22 y 5.23 son el resultado de desplazar el objeto cerca del fondo blanco. Para las figuras anteriores (5.22 y 5.23) en los frames 20 y 21, el objeto de interés se desplazó a una velocidad que no pudo ser detectada por las cámaras. También se puede observar que en

la figura 5.23 (frame 4), el algoritmo detectó la sombra proyectada y promedió la ubicación entre el objeto de interés y la sombra.

### Movimiento sobre fondo verde

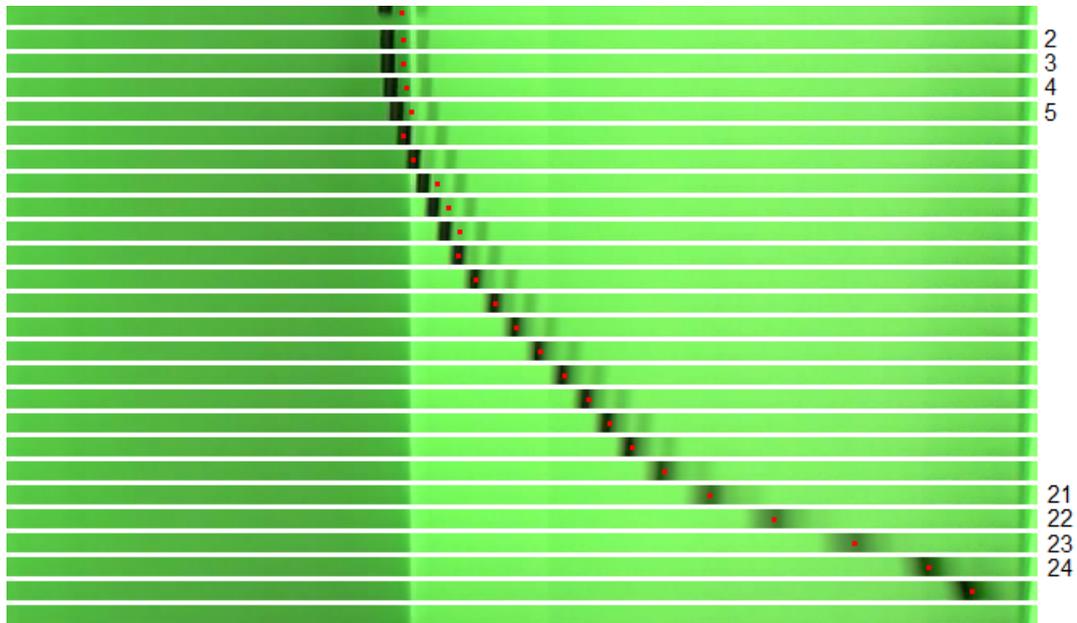


Figura 5.24. Movimiento del apuntador sobre fondo verde: imagen a color.

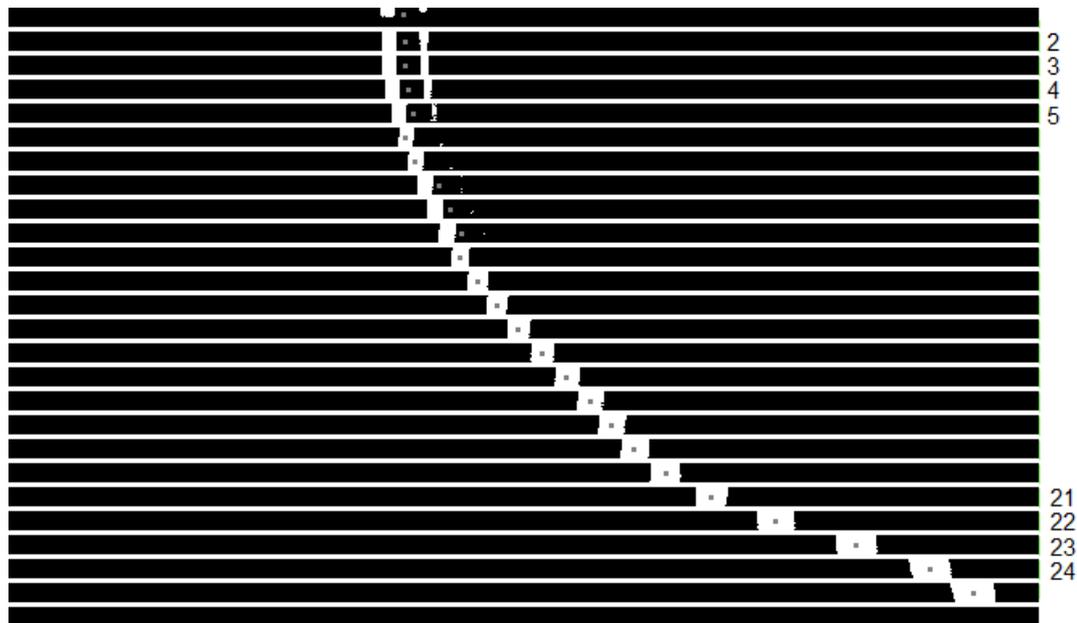
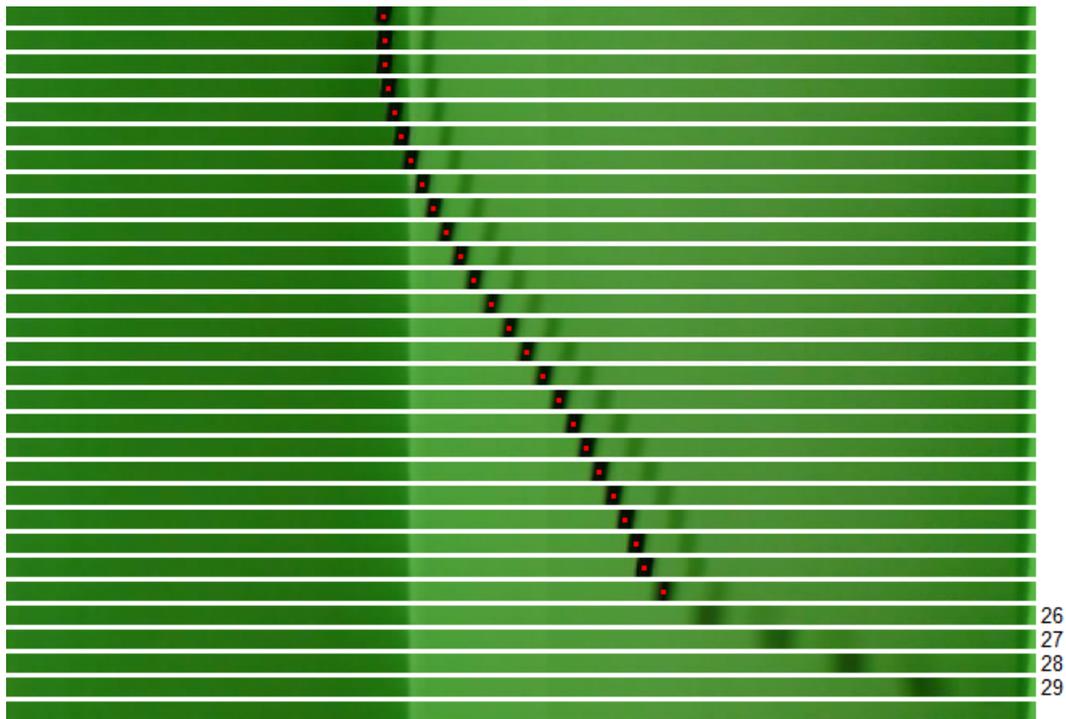


Figura 5.25. Movimiento del apuntador sobre fondo verde: Imagen umbralizada (Umbral 40), sin erosión.

El conjunto de figuras 5.24 y 5.25 son el resultado de desplazar el objeto sobre el fondo verde, si observamos estas imágenes en los frames: 2, 3, 4 y 5, se obtienen los mismos problemas que al utilizar fondo blanco, dado que en algunos sectores de la pizarra el algoritmo de detección promedia con la sombra. Sin embargo se observaron algunas mejoras con respecto a la velocidad de desplazamiento debido que sí se detectaba el objeto (ver frames 21, 22, 23 y 24).



**Figura 5.26.** Movimiento del apuntador sobre fondo verde en la parte inferior de la pizarra. Imagen a color.

En las figuras 5.26 y 5.27 las sombras del objeto de interés no afectan la ubicación del apuntador, sin embargo al moverse rápidamente sobre la pizarra la erosión lo elimina de la imagen como se observa en los frames 26, 27 y 28.

Como se observó en las pruebas anteriores, no se obtuvieron buenos resultados, dado que en un caso no se eliminaba la sombra y en el otro el apuntador no se podía detectar al desplazarlo sobre la pizarra.

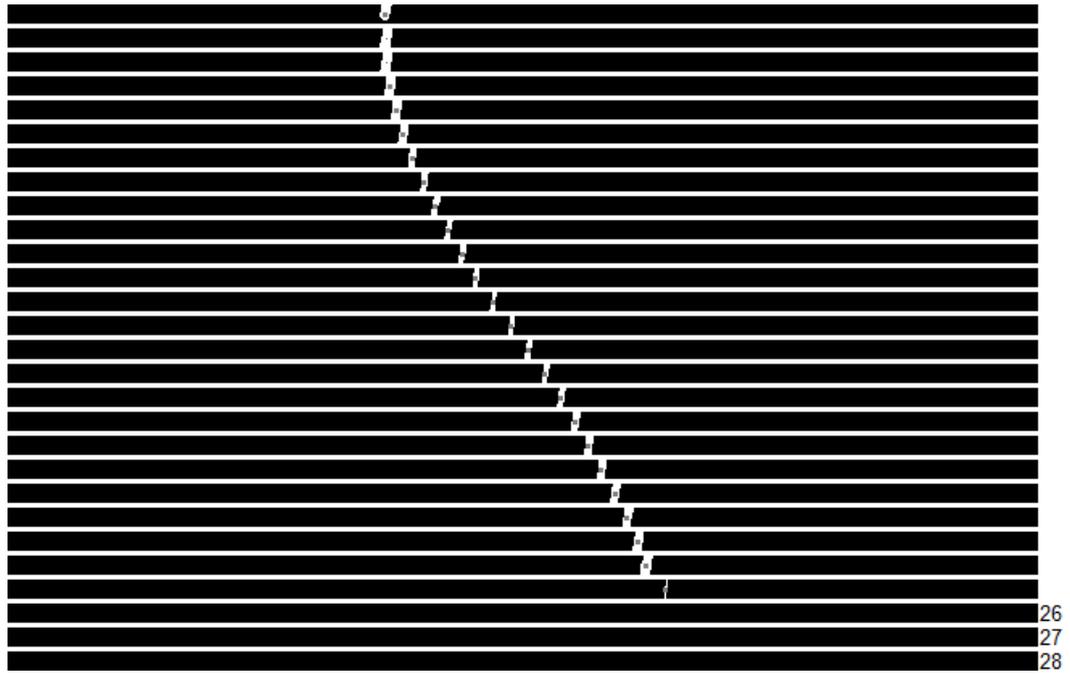


Figura 5.27. Movimiento del apuntador sobre fondo verde en la parte inferior de la pizarra. Imagen umbralizada: umbral 27, erosión 1.

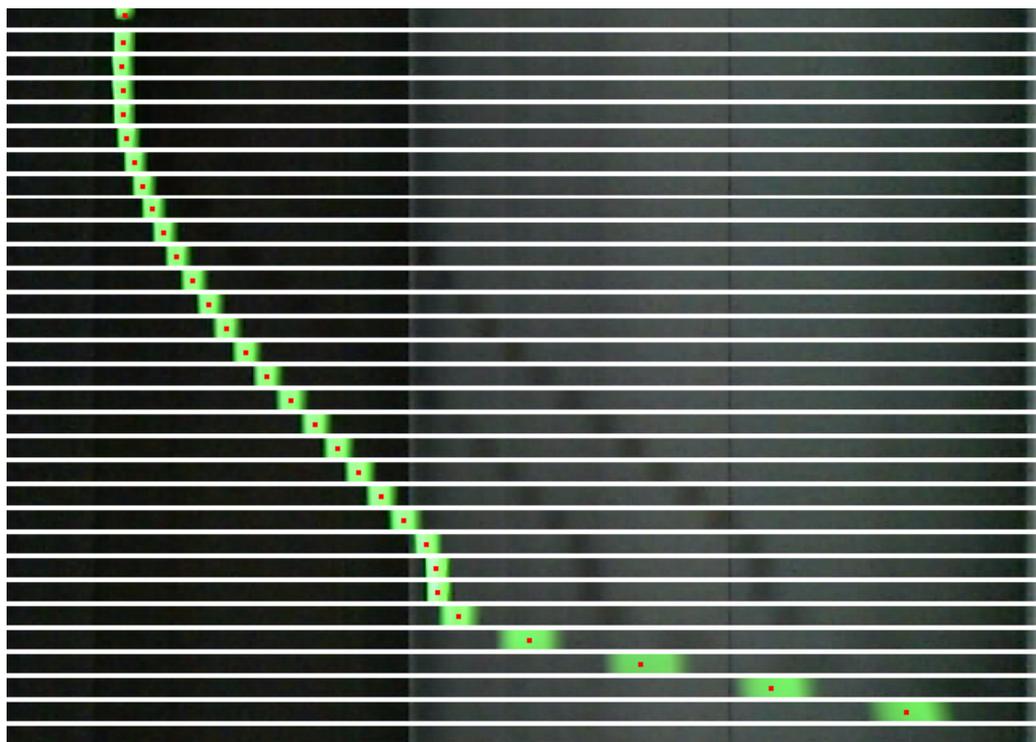


Figura 5.28. Movimiento del apuntador sobre fondo negro en la parte superior de la pizarra. Imagen a color.

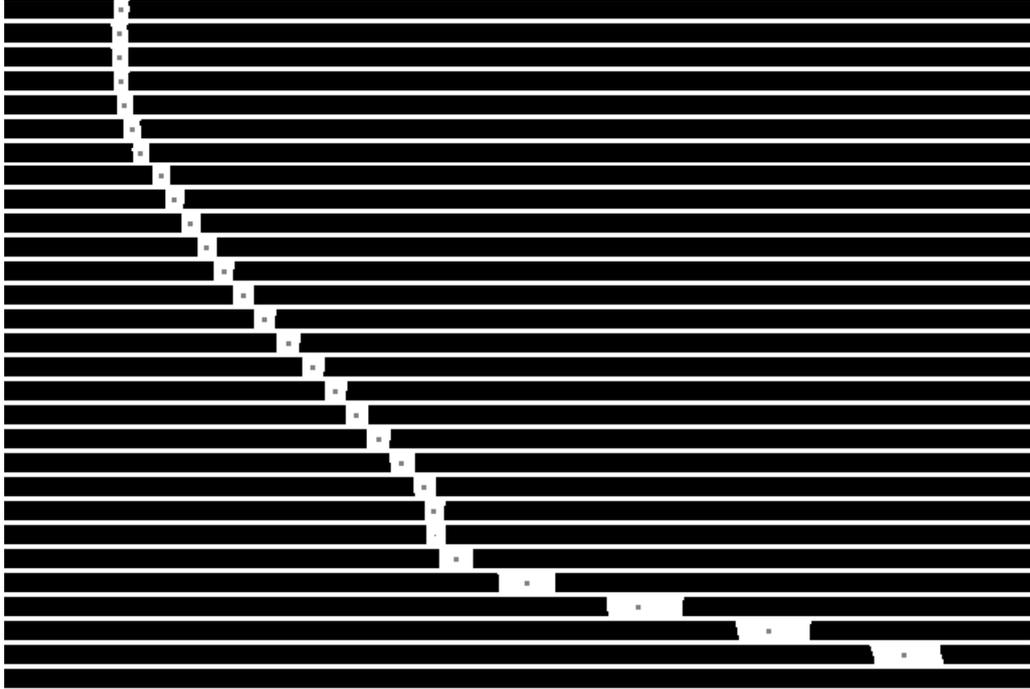


Figura 5.29. Movimiento del apuntador sobre fondo negro en la parte superior de la pizarra. Imagen umbralizada: umbral 100, erosión 1.

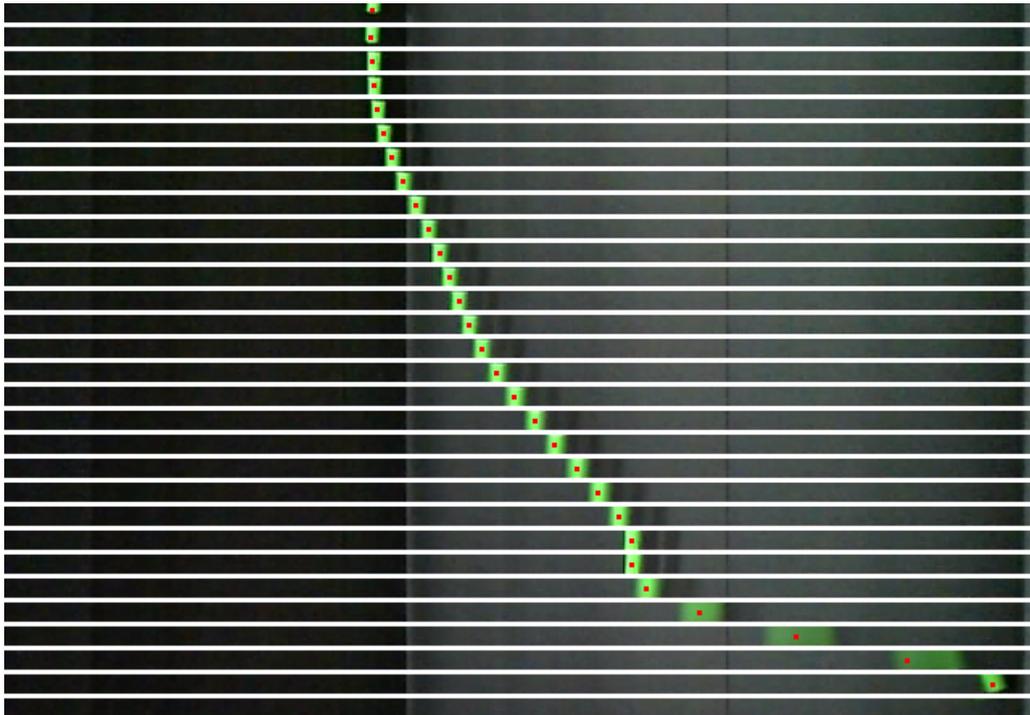


Figura 5.30. Movimiento del apuntador sobre fondo negro en la parte inferior de la pizarra. Imagen a color.

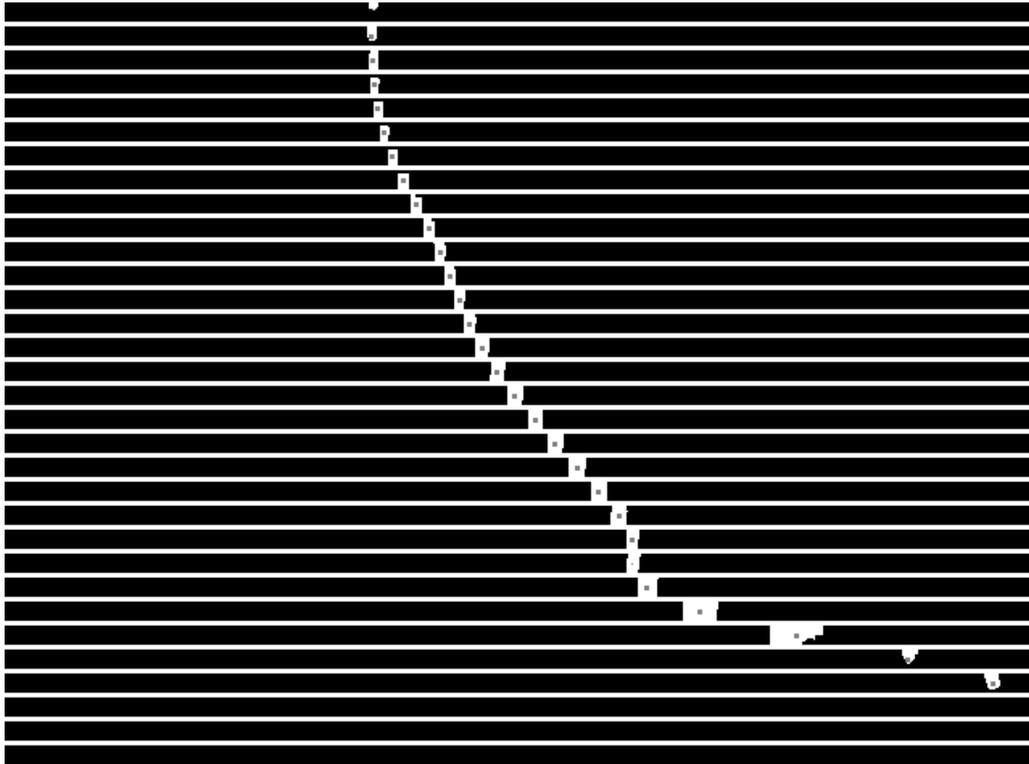


Figura 5.31. Movimiento del apuntador sobre fondo negro en la parte inferior de la pizarra. Imagen umbralizada: Umbral 100, erosión 1.

### Movimiento sobre fondo negro

En las figuras 5.30 y 5.31 sobre fondo negro, se observa que la sombra del objeto de interés no es detectada debido al color del fondo. Adicionalmente, el objeto no desaparece a pesar de la velocidad de desplazamiento sobre la pizarra.

Los resultados obtenidos al utilizar fondo negro fueron los mejores al momento de realizar procesamiento de imágenes. Por lo que se utilizaron estas condiciones en el sistema final.

#### 5.1.6. Detección de color

Estas pruebas se realizaron con la finalidad de determinar los colores que ofrecen mejor contraste con el fondo negro a utilizarse en el sistema. Además estos colores deben ser diferenciables entre sí, y así poder utilizarlos para hacer distintas acciones como click y doble click.

Como se observa en la figura 5.33, se colocaron objetos de interés de diferentes materiales y colores de alto contraste con el fondo, los colores utilizados fueron: cian, amarillo, magenta, verde y blanco.



Figura 5.32. Imagen de los colores utilizados.

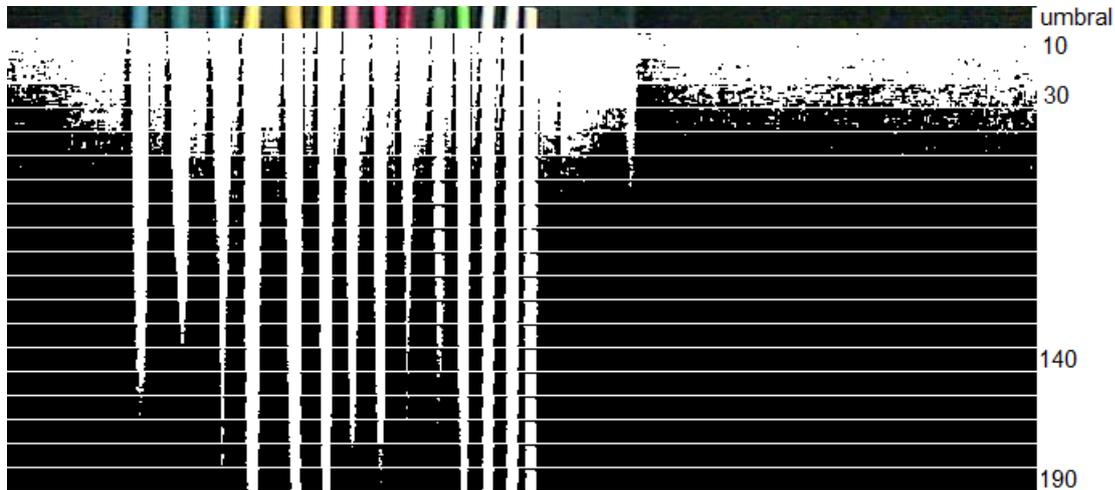


Figura 5.33. Umbralización aplicada a distintos colores (umbral de 10-190).

Los colores que se mantienen al utilizar el umbral más grande (umbral de 190) fueron los colores: amarillos, verde fosforescente, y blancos. Dada esta propiedad, estos tienen mayor posibilidad de mantener su intensidad aún cuando se muevan sobre la pizarra. Sin embargo los colores amarillos, verdes y blancos utilizados tienen intensidades parecidas, lo que hace difícil acotar sus bandas RGB para hacer la distinción entre estos colores.

Después de los colores anteriormente mencionados, el color que se destacaba mejor al utilizar umbrales grandes era el magenta. Además, al ser el color complementario del verde, se puede distinguir fácilmente del mismo. Por esta razón se tomaron los colores magenta y verde para hacer las acciones de click y doble click respectivamente.

## 5.2. Pruebas finales

Las pruebas que se presentan a continuación fueron realizadas al terminar el desarrollo del sistema.

### 5.2.1. Rendimiento del sistema

Al desarrollar el sistema se hicieron varios algoritmos de búsqueda para detectar el marcador en las imágenes capturadas por las webcams, estos algoritmos aparecen explicados en la sección 4.4.4.

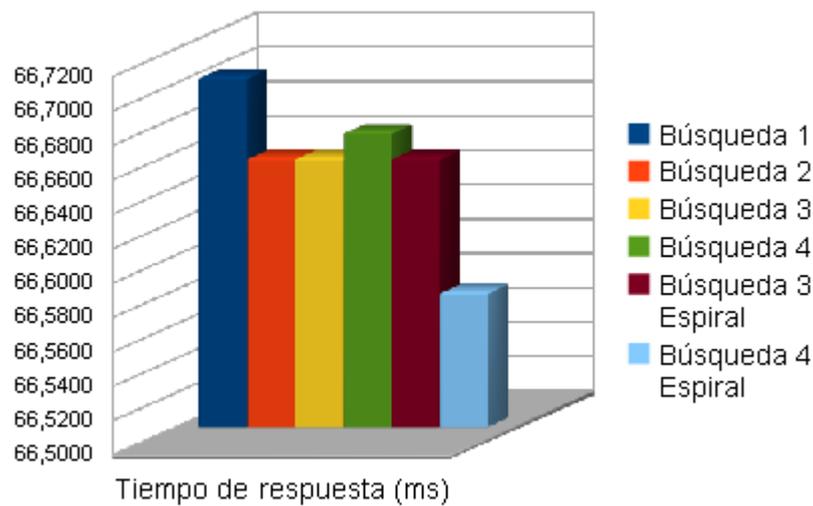
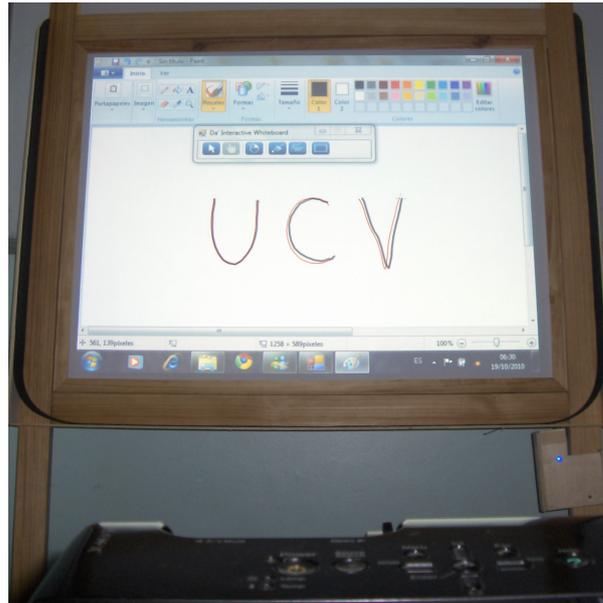


Figura 5.34. Tiempo de respuesta del sistema utilizando diferentes algoritmos de búsqueda.

Para comparar el rendimiento de cada uno de estos algoritmos se realizaron mediciones del tiempo de respuesta del sistema. Como se puede observar en la figura 5.34 la diferencia entre los tiempos de respuesta utilizando cada uno de los algoritmos no es significativa, sin embargo se pudo notar una leve mejora utilizando el algoritmo 4 con búsqueda en espiral, por esto es algoritmo es utilizado en el sistema final. El mejor tiempo de respuesta de sistema obtenido fue 66,5780 ms. Esto es equivalente a detectar el marcador 15 veces por segundo.

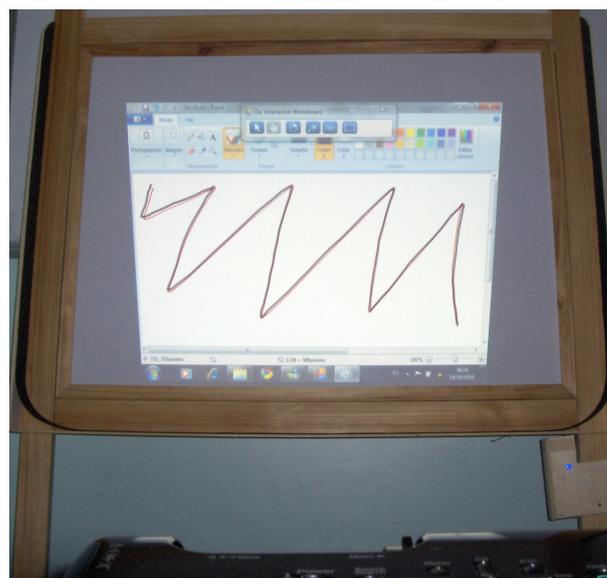
### 5.2.2. Precisión del sistema

Para comprobar la precisión del sistema al detectar la ubicación del marcador se realizaron trazos utilizando un marcador acrílico rojo sobre la pizarra mientras se utilizaba la herramienta de dibujo del sistema operativo (*Paint Brush*), de esta manera se pueden comparar los trazos reales hechos por el marcador sobre la pizarra y los trazos hechos por software. En las figuras 5.35 y 5.36 se pueden observar cómo se realizaron las pruebas, adicionalmente, de la figura 5.37 a la 5.42 se muestran con detalle los resultados obtenidos.



**Figura 5.35. Pruebas de precisión.**

La imagen fue tomada desde una cámara situada detrás de la proyección del videobeam. Los trazos rojos se realizaron usando marcador acrílico sobre la pizarra, y los trazos negros son el resultado sobre la herramienta de dibujo.



**Figura 5.36. Pruebas de precisión (Proyección en forma de trapecio).**

La imagen fue tomada desde una cámara situada detrás de la proyección del videobeam, está proyección fue calibrada con la proyección en forma de trapecio generada por el videobeam.

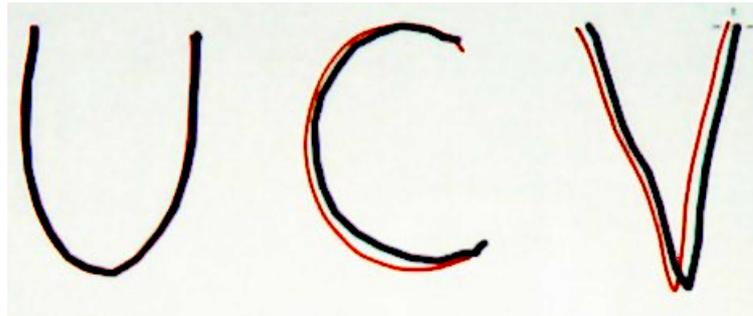


Figura 5.37. Pruebas de precisión (UCV).

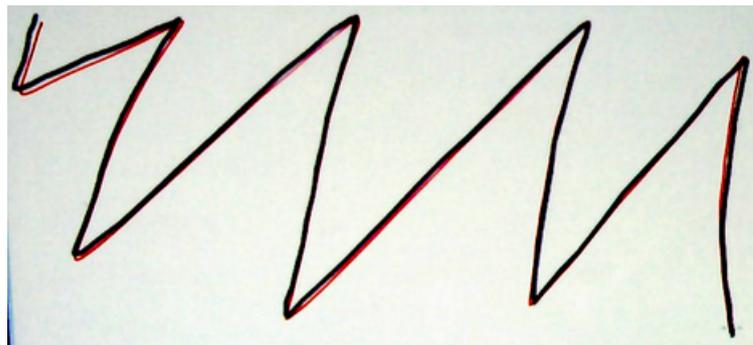


Figura 5.38. Pruebas de precisión (Zig-zag).

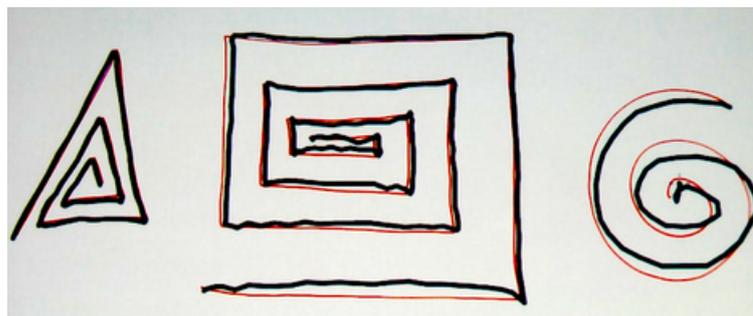


Figura 5.39. Pruebas de precisión (Figuras).

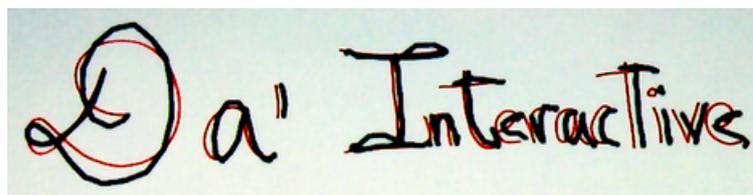


Figura 5.40. Pruebas de precisión (DaInteractive).

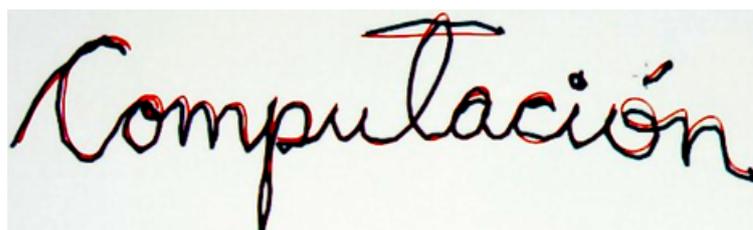


Figura 5.41. Pruebas de precisión(Computación).

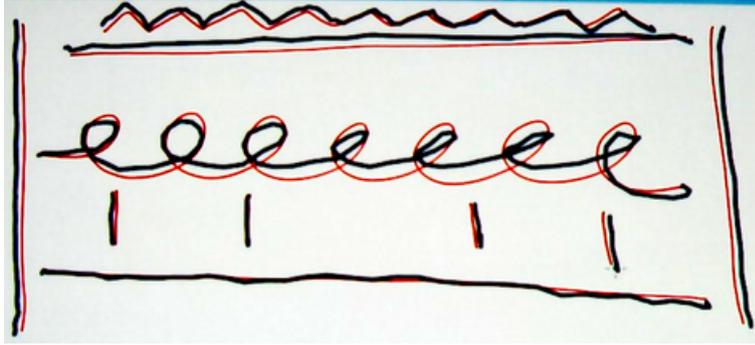


Figura 5.42. Pruebas de precisión (Figuras varias).

---

## Conclusiones

La visión por computador ha servido como base para el desarrollo de nuevas tecnologías, entre ellas se encuentran las pizarras interactivas. Estas ofrecen grandes posibilidades que las han llevado a ser utilizadas en ambientes educativos.

Nuestro proyecto permite utilizar una pizarra acrílica, dos webcams y un video beam como pizarra interactiva. Además se construyó una interfaz que permite la calibración del sistema, y provee al usuario las funcionalidades de hacer click derecho y hacer anotaciones sobre la imagen en pantalla. Se utiliza un apuntador de dos colores, que permite realizar acciones como click, doble click y mover al apuntador.

Se utilizaron técnicas de procesamiento digital de imágenes para detectar la ubicación del apuntador en las imágenes capturadas por las webcams. También se implementaron varios métodos de calibración de las cámaras con la finalidad de conocer las coordenadas del apuntador sobre la pizarra.

Con la realización de este prototipo se obtendrán muchos beneficios, ya que permite incluir una herramienta para la enseñanza que puede ser útil en diversos niveles. Por ejemplo, en la educación primaria puede ser utilizada para incluir juegos educativos que aumenten la participación y atención de los niños en clase; en la educación secundaria permitiría mostrar contenido interactivo para lograr un mayor dinamismo en las clases; y en general puede ser utilizada para enseñar el manejo del computador o para mostrar el funcionamiento de algún software.



---

## Trabajos Futuros

El trabajo elaborado presenta un prototipo, en el que se pueden mejorar ciertas características, como las siguientes:

- Utilizar un método alternativo de calibración de las cámaras para lograr una mayor precisión.
- Adaptar al sistema para ser utilizado en múltiples sistemas operativos. Para esto se puede compilar el sistema utilizando el proyecto *Mono*[36], que permite compilar y ejecutar programas desarrollados en *.Net Framework* bajo Linux o Mac OS.
- Mejorar la interacción con el usuario, para que pueda realizar acciones como doble click de una forma más intuitiva.
- Crear un prototipo donde se pueda utilizar proyección trasera para evitar que el usuario genere sombras sobre la proyección del video beam y así facilitar el uso del sistema.



---

## Referencias

1. Umbaugh, S. (2005, Enero). *Computer imaging: digital image analysis and processing*. por CRC Press, pp.3-8,15-19,43-54,67-69. Citado en página(s). 17, 18, 19, 32
2. González, R., Wood, R. (1996, Enero). *Tratamiento digital de imágenes*. por Addison-Wesley, pp.51-53. Citado en página(s). 20, 21
3. Molleda, J. (2008, Julio). *Técnicas de visión por computador para la reconstrucción en tiempo real de la forma 3D de productos laminados*. Departamento de Informática - Universidad de Oviedo, pp.13-39. Citado en página(s). 29, 34, 35
4. Sbert, C. (2008,2009). *Procesamiento Digital de la Señal*. Curso - Universidad de les Illes Balears. Citado en página(s). 18, 24
5. Hartley, R., Zisserman, A. (2005, Enero). *Multiple view geometry in computer vision*. por Cambridge University Press, pp.1-13,25-31. Citado en página(s). 31
6. Sosa, J. (2007, Junio). *Sistema de visión basado en procesamiento guiado por cambios y lógica reconfigurable para el análisis de movimiento de alta velocidad*. Universidad de Valencia, pp.13-39,44-49. Citado en página(s). 31, 37, 38, 41
7. Cheung, S., Kamath, C. (2007, Junio). *Robust techniques for background subtraction in urban traffic video*. pp.1-2. Citado en página(s). 42
8. Sarkar, S., Majchrzak, D., Korimilli, K. (2002, Febrero). *Perceptual Organization Based Computational Model for Robust Segmentation of Moving Objects*. pp.1-3. Citado en página(s). 41
9. Alvarez, R. (2004, Septiembre). *Procesado de la Imagen*. Extraído el 23 de Mayo de 2010 desde <http://sabia.tic.udc.es/gc/Contenidos%20adicionales/trabajos/Imagenyvideo/Procesado%20de%20imagen/> Citado en página(s). 23
10. Lizarraga, C., Ortega, L. (2008, Octubre). *Visión Artificial para detección y ubicación espacial de objetos en una escena*. Escuela de Computación - Universidad Central de Venezuela, pp.32-34,45. Citado en página(s). 25, 26, 33, 65
11. Satorres, S. (2006) *Detección de bordes en una imagen*. Curso - Universidad de Jaén. Citado en página(s). 26
12. Monedero, J. (2008, Septiembre). *Teoría de la Imagen Digital. Conceptos Básicos*. Curso ETSAB, pp.25-29. Citado en página(s). 31
13. National Centre of Technology in Education. (2008, Noviembre). *NCTE Advice Sheet 11*. NCTE, pp.1-2. Citado en página(s). 31
14. Layton, J. *Procesado de la Imagen*. Extraído el 20 de Mayo de 2010 desde <http://www.howstuffworks.com/webcam.htm> Citado en página(s). 31
15. Munro, P., Gerdelan, A. (2006). *Stereo Vision - Computer Depth Perception*. por Massey University, Albany New Zealand, pp.7-8. Citado en página(s). 33, 36
16. Valdemar, E., Zaldivar, D., Rojas, R. (2007). *Intelligent active vision systems for robots*. por Cuvillier Verlag, pp.133-134. Citado en página(s). 33
17. National Instruments. (2005, Enero). *NI Vision.OOO*, pp.104-105. Citado en página(s) 29
18. Valdemar, E., Zaldivar, D.(). *Visión por computador utilizando MatLAB y el Toolbox de Procesamiento Digital de Imágenes*, pp.22-23. Citado en página(s) 29
19. Castillo, R., Díaz, I., Huertas, M. (2006, Julio). *Localización espacial de un punto en XYZ mediante visión artificial*. Universidad Militar Nueva Granada 7(3), pp.2-3. Citado en página(s) 37
20. PLUS Vision Corp., of America. (2010, Enero). *Copyboards - PLUS Vision Corp. of America Version 1.0*. Extraído el 25 de diciembre de 2009 desde [http://www.plus-america.com/list\\_copyboards.html](http://www.plus-america.com/list_copyboards.html) Citado en página(s). 46
21. Wedgwood IT Group. (2008). *Whiteboard guide*. Citado en página(s) 46, 47, 48, 49
22. Brown, S.,(). *Interactive Whiteboards in Education*, pp.1-2. Citado en página(s) 47, 48

23. PolyVision.(2009, Diciembre). *About Anoto*. Extraído el 21 de diciembre de 2009 desde <http://www.polyvision.com/ProductSolutions/Interactivewhiteboards/enointeractivewhiteboard/tabid/157/Default.aspx> . Citado en página(s). 51
24. AV news. (2009, Febrero). *About Anoto*. Extraído el 15 de febrero de 2010 desde [http://www.avnews.co.uk/avn\\_main\\_contents/avn\\_home\\_page\\_feb09/avn0209\\_productupdate\\_iwb\\_polyv.htm](http://www.avnews.co.uk/avn_main_contents/avn_home_page_feb09/avn0209_productupdate_iwb_polyv.htm) . Citado en página(s). 51
25. Articles Base. (2009, Abril). *Looking for the Best Interactive White Board*. Extraído el 14 de febrero de 2010 desde <http://www.articlesbase.com/college-and-university-articles/looking-for-the-best-interactive-white-board-851887.html> . Citado en página(s). 51
26. Betcher, C., Lee, M. (2009). *The Interactive Whiteboard revolution* . Australia: ACER Press 51
27. Soares, C.(2009). *Quadro Interactivo de Baixo Custo Recurriendo a Algoritmos de Visao por Computador*. Porto: Facultad de Ciencia e Tecnología - Universidade Fernando Pessoa. Citado en página(s) 52, 53
28. Chung, J. (2007). *Johnny Chung Lee - Projects - Wii*. Extraído el 29 de noviembre de 2009 desde <http://johnnylee.net/projects/wii/>. Citado en página(s) 56
29. Nintendo. (2006). *Wii - página oficial*. Extraído el 29 de noviembre de 2009 desde <http://latam.wii.com/>. Citado en página(s) 56
30. Chung, J. (2008,Julio). *Hacking the Nintendo Wii Remote*. *IEEE Pervasive Computing* 7(3), pp.39-45. Citado en página(s) 56, 57
31. Todorovi, D., Prokic, A., Đorđević, G. (2009, Junio). *Interactive Multimedia Touch Sensitive System*. pp.38-40. Citado en página(s) 59
32. Bradski, G., Kaebler, A. (2008, Septiembre). *Learning OpenCV*. por O'Reilly, pp.1. Citado en página(s) 65
33. EmguCV, Wiki. (Última actualización: 15 de Abril del 2010). *Emgu CV* . Extraído el 07 de Junio de 2010 desde [http://www.emgu.com/wiki/index.php/Main\\_Page](http://www.emgu.com/wiki/index.php/Main_Page) . Citado en página(s) 66
34. Microsoft Corporation. (2007). *Microsoft LifeCam VX-7000 Technical Data Sheet*. Extraído el 11 de octubre de 2010 desde [http://download.microsoft.com/download/9/1/f/91f19ea7-6de0-4e03-b00a-56bf61b61b5e/TDS\\_LifeCamVX-7000\\_0710A.pdf](http://download.microsoft.com/download/9/1/f/91f19ea7-6de0-4e03-b00a-56bf61b61b5e/TDS_LifeCamVX-7000_0710A.pdf). Citado en página(s) 61
35. Microsoft Corporation. *Visual C#*. Extraído el 11 de octubre de 2010 desde [http://msdn.microsoft.com/en-us/library/kx37x362\(v=VS.90\).aspx](http://msdn.microsoft.com/en-us/library/kx37x362(v=VS.90).aspx). Citado en página(s) 65
36. Mono Project. *Mono*. Extraído el 20 de octubre de 2010 desde [http://www.mono-project.com/Main\\_Page](http://www.mono-project.com/Main_Page). Citado en página(s) 105