



**FACULTAD DE INGENIERÍA Y ARQUITECTURA
ESCUELA PROFESIONAL DE INGENIERÍA ELECTRÓNICA**

**DISEÑO DE PIZARRA VIRTUAL INTERACTIVA MEDIANTE
DETECCIÓN DE LUMINANCIA BASADO EN TÉCNICAS DE
PROCESAMIENTO DIGITAL DE IMÁGENES**

**PRESENTADA POR
JUAN JOSÉ CÁRDENAS AVILÉS**

**TESIS PARA OPTAR EL TÍTULO PROFESIONAL DE
INGENIERO ELECTRÓNICO**

LIMA – PERÚ

2015



**Reconocimiento - No comercial - Compartir igual
CC BY-NC-SA**

El autor permite transformar (traducir, adaptar o compilar) a partir de esta obra con fines no comerciales, siempre y cuando se reconozca la autoría y las nuevas creaciones estén bajo una licencia con los mismos términos.

<http://creativecommons.org/licenses/by-nc-sa/4.0/>



ESCUELA PROFESIONAL DE INGENIERÍA ELECTRÓNICA

**DISEÑO DE PIZARRA VIRTUAL INTERACTIVA MEDIANTE
DETECCIÓN DE LUMINANCIA BASADO EN TÉCNICAS DE
PROCESAMIENTO DIGITAL DE IMÁGENES**

TESIS

**PARA OPTAR EL TÍTULO PROFESIONAL DE INGENIERO
ELECTRÓNICO**

PRESENTADO POR

CÁRDENAS AVILÉS, JUAN JOSÉ

LIMA – PERÚ

2015

ÍNDICE

	Página
RESUMEN	iv
ABSTRACT	v
INTRODUCCIÓN	vi
CAPÍTULO I. MARCO TEÓRICO	1
1.1. Antecedentes	1
1.2. Bases teóricas	5
1.3 .Definición de términos básicos	10
CAPÍTULO II. METODOLOGÍA	11
2.1. Materiales	11
2.2. Metodología	13
CAPÍTULO III. DESARROLLO DEL PROYECTO	24
3.1. Configuración de Logitech Quickcam	26
3.2. Calibración del sistema	27
3.3. Algoritmo de sustracción de fondo para detección de puntero led	29
3.4. Algoritmo de evaluación de áreas para eliminación de interferencias	31
3.5. Algoritmo de evaluación de centroide para cálculo de posición	33

3.6. Algoritmo de control de funciones del mouse	35
3.7. Construcción de puntero led	37
CAPÍTULO IV. PRUEBAS Y RESULTADOS	39
4.1. Proceso de calibración de sistema	40
4.2. Proceso de detección de puntero	38
4.3. Pruebas de distancia de funcionamiento	45
4.4. Pruebas de tiempo de ejecución	47
4.5. Pruebas de retardo de programa	48
4.6. Pruebas de funcionamiento de prototipo	49
4.7. Sustento desde el punto de vista comercial	50
CAPÍTULO V. DISCUSIÓN Y APLICACIONES	52
CONCLUSIONES	54
RECOMENDACIONES	56
FUENTES DE INFORMACIÓN	58
ANEXOS	60

RESUMEN

El presente proyecto tiene como objetivo el diseño de una pizarra virtual interactiva con funcionamiento similar al de pizarras interactivas que se encuentran actualmente en el mercado, de manera que se convierta en una solución de bajo costo y una herramienta más accesible para el desarrollo de las actividades educativas y/o empresariales para instituciones de bajos recursos. El término virtual se debe a que no requiere de una superficie electrónica especializada para determinar la posición del puntero. Su funcionamiento y proceso de interacción con el usuario está basado en técnicas de procesamiento digital de imágenes, obteniendo información producida por un puntero emisor de luz, en tiempo real, mediante una webcam, que permite utilizar la computadora mediante el control del mouse.

En la investigación realizada, se utilizaron técnicas de procesamiento digital de imágenes para realizar la detección de luminancia, además de funciones importadas de lenguaje C para que operen como funciones propias de Matlab® y controle los eventos propios del mouse.

ABSTRACT

This project have as objective the design of a virtual interactive whiteboard with similar operation than commercial interactive whiteboards to become a low-cost and a more accessible tool for development of educational or business activities for low resources institutions. The virtual term is because that does not require a specialized electronic surface to determinate the pointer position. Its operation and interaction process is based on techniques of digital images processing, obtaining information from a transmitter pointer light captured by a webcam on real time through a webcam, permitting use the computer through the mouse control.

In the research, processing techniques for digital image luminance detection were used, in addition to imported C language functions to operate as own Matlab functions and get control of the mouse events.

INTRODUCCIÓN

El objetivo general del proyecto es diseñar un sistema que permita brindar interacción a distancia entre la computadora y el usuario a través de control de mouse mediante el uso de un puntero, basado en técnicas de procesamiento digital de imágenes para poder utilizar cualquier superficie como una pizarra digital interactiva.

Los objetivos específicos están definidos como:

- Modelamiento de un algoritmo de detección de luminancia mediante umbralización global para detección de puntero.
- Modelamiento de un algoritmo de evaluación de áreas de objetos para eliminación de interferencias.
- Modelamiento de algoritmo para corrección de distorsión proyectiva mediante homografías y/o técnicas de transformaciones proyectivas.
- Modelamiento de algoritmo de evaluación de centroide de objeto para cálculo de coordenadas.
- Modelamiento de función de evaluación de eventos de Windows para control de funciones de mouse.

La tesis está conformada por cinco capítulos. En el primero, se explican los antecedentes y el estado del arte del proyecto resultado de la investigación realizada así como el marco teórico que sirve como base para el desarrollo del proyecto. En el segundo, se detallan los métodos utilizados para el modelamiento del sistema, indicando los fundamentos matemáticos que se aplican en cada etapa que comprende el proceso. En el tercero, se describen los procesos y esquemas implementados de acuerdo a los ejes de modelamiento definidos para el desarrollo del proyecto. En el cuarto, se muestran los resultados obtenidos durante las pruebas de funcionamiento realizadas. En el quinto, se comentan las diferentes situaciones presentadas en la tesis, mostrando las aplicaciones del sistema.

CAPÍTULO I

MARCO TEÓRICO

1.1. Antecedentes

La pizarra interactiva es una tecnología diseñada para proporcionar al usuario herramientas que faciliten y estimulen el entorno interactivo, colaborativo y multimedia de una clase. Estos dispositivos ofrecen la capacidad de interactuar sobre una imagen proyectada a través de un proyector multimedia. Dicha interactividad se da gracias a la capacidad de la pizarra de detectar las acciones que se realicen sobre ella con un 'bolígrafo digital', comunicando el resultado de estas acciones a una computadora conectada a la pizarra. Según el artículo "*The benefits of an interactive whiteboard*", realizado por la *British Educational Communications and Technology Agency BECTA* (2006), la pizarra interactiva es una herramienta aplicable a todas las etapas educativas que aumenta la eficiencia y eficacia del proceso de enseñanza, además de ser una tecnología atractiva de manejo sencillo que aumenta la motivación y aprendizaje, ofrece un acercamiento de las tecnologías de información y comunicación a alumnos con discapacidades.

En conclusión, una pizarra digital es una herramienta utilizada para enriquecer distintas clases de contenidos utilizados en la enseñanza, debido a la intervención directa del usuario sobre la proyección sin que este deba

desplazarse. En este marco, el desarrollo de nuevas herramientas tecnológicas de bajo costo e investigación, en este campo, brindará al proceso de enseñanza los beneficios propios de las pizarras digitales existentes y contribuirá a disminuir la brecha que separa a las instituciones educativas de los últimos avances tecnológicos.

1.1.1. Estado del arte

En el estado del arte global investigado en torno al desarrollo de estudios en procesamiento digital de imágenes y/o visión por computador para implementar sistemas de pizarra interactiva se tienen proyectos como el desarrollado por Johnny Chung Lee, *“Low-Cost Multi-point Interactive Whiteboards Using the Wiimote”*, en la *Carnegie Mellon University, Pittsburgh, USA*, donde se logró hackear la *Wiimote* (mando del Nintendo Wii) para utilizarlo como herramienta para la adquisición de datos provenientes de un lapicero de led infrarrojo, debido a que el mando de Wii es un dispositivo de detección de movimiento que tiene una cámara de infrarrojos de alta calidad. El sistema de pizarra interactiva desarrollado por Chung Lee establece la conexión entre el computador y el mando de wii mediante bluetooth y detecta la luz infrarroja del lapicero led para determinar la posición del mouse.

En el año de 2010, en la *ACM International conference on Interactive Tabletops and surfaces*, realizado en USA, Otto Korkalo y Petri Honkkaa, del *VTT Technical research Center of Finland*, presentaron el proyecto *“Construction and evaluation of multi-touch screens using multiple cameras located on the side of the display”*, que aplica el método de visión por computador y tecnología multitáctil desarrollado para ser adoptado por varios tipos de pantalla. El diseño consta de múltiples cámaras colocadas en un eje óptico paralelo en un lado de la pantalla con los bordes opuestos iluminados por emisores infrarrojos. La detección de los dedos se realiza debido a que bloquean la luz en las imágenes de las cámaras, con lo que

brindan la información necesaria para determinar la posición donde se realizan los movimientos y/o tareas.

En el año de 2011, *“Electronic Whiteboard System”*, de los autores Jen-Tsorng Chang y Tu-Cheng, patentado con el Pub. No.: US 2011/0157101 A1, incluye un proyector, al menos dos dispositivos de captura de imágenes, una pizarra electrónica, un lapicero de luz y un procesador. El lapicero de luz posee al menos dos emisores de luz de diferente longitud de onda de luz (color), cada dispositivo de captura de imagen detecta color respectivo del lapicero de luz, transmite la señal de imagen al procesador, conectado a los dispositivos de imagen y al proyector, para que emita una marca de color a la pizarra electrónica.

Con referencia a diferentes métodos y/o técnicas encontrados, podemos apreciar muchos relacionados con el control del cursor del mouse, en el paper *“Mouse control using webcam based on color detection”*, publicado en Marzo del 2014 en el *International Journal of Computer Trends and technology* (UCTT), desarrollado en el *Sikkin Manipal Institute of Technology*, India, realizan el control del mouse del computador mediante la identificación de una marca en el dedo del usuario, mediante el uso técnicas de detección de color por métodos de visión por computador, utilizando una webcam para lograr la interacción con el computador.

Otro método es presentado en el paper *“Method for controlling mouse movement using real-time camera”*, desarrollado en el *Department of computer Science Brown University, Providence, RI, USA*, basado en segmentación de imágenes para la detección de la mano y el reconocimiento de gestos para lograr controlar las tareas del mouse, donde los diferentes gestos corresponden a los diferentes eventos del mouse, como clic izquierdo, derecho, doble clic, etc.

El proyecto de tesis *“Mouse cursor control with head and eye movements: A low-cost approach”*, presentado para obtener el grado de Maestría en el campo de Ciencias de Ingeniería Biomédica de la *University*

of Applied Sciences Technikum Wien, Austria, utiliza un sistema montado en la cabeza para controlar el cursor del mouse mediante seguimiento ocular, utiliza también seguimiento de cabeza, mediante sensores infrarrojos, para compensar los movimientos de la misma durante el seguimiento ocular. Esta tesis formó parte del programa denominado *AsTeRICS (Assistive Technologies)*, proyecto de educación intercultural e investigación en tecnología de apoyo a personas con discapacidades motrices y necesidades especiales.

En el entorno comercial, las pizarras interactivas se clasifican según la tecnología utilizada para su funcionamiento, como el caso de la marca InterWrite®, basada en tecnología electromagnética, combina una malla contenida en toda la superficie electrónica que detecta la señal electromagnética emitida por el puntero en cualquier parte de ella, permitiendo localizar el punto señalado. Dispositivos como los ofrecidos por la marca eBeam® utilizan sensores de Ultrasonido-Infrarrojo, colocados a un lado de la superficie, que detectan la señal emitida por el marcador y efectúan el sincronismo para calcular la posición del puntero.

Mientras, dispositivos como los de SmartBoard® funcionan con tecnología resistiva, donde el panel de la pizarra está formado por dos capas separadas, la exterior deformable al tacto. La presión aplicada facilita el contacto entre las láminas provocando una variación en la resistencia eléctrica para permitir localizar el punto señalado. Otro tipo de pizarra emplea tecnología óptica como la marca IQBoard®, con el perímetro de la superficie cubierto por barras de luz infrarroja y dos cámaras en la parte superior que controlan que no se rompa la línea de vista entre ellos. En el momento en que el usuario presiona sobre alguna zona del área interactiva, el sistema calcula donde se ha distorsionado la señal y calcula la posición en un plano cartesiano.

1.1.2. Análisis de costos en el mercado actual

Teniendo en cuenta que el proyecto se propone como una solución de bajo costo, se debe tener en consideración los precios de las pizarras interactivas comerciales. Para ello, se muestran algunos precios encontrados de diferentes productos.

Tabla 1: Evaluación de precios

Marca / Modelo	Precio en dólares
Mimio Teach	899.00
eBeam Edge	649.00
Hitachi link EZ	700.00
Hitachi 79E1	1245.00
Hitachi 89WE1	1745.00
Panasonic UB-8325	2410.00
SmartBoard	3200.00

Fuente: Amazon.com

1.2. Bases teóricas

1.2.1. Modelo de color RGB

Es el espacio de color más extendido y el que utilizan la gran mayoría de cámaras de video y fotográficas para construir una imagen de color. Y de ahí, su importancia en procesamiento de imágenes ya que trabajar con el mismo espacio de color con el que trabaja la cámara con la que se capturan las imágenes evita la alteración de las propiedades del color durante el proceso de segmentación, propio de los errores de conversión y transformación, y por otro lado, conseguir una mayor velocidad de segmentación por ahorro de esas operaciones de conversión y redondeo.

El espacio RGB, definido así por las siglas de sus colores primeros en inglés, se representa como un cubo donde un color viene definido por la mezcla de valores de intensidad de tres colores primarios, rojo, verde y azul. Un color viene descrito por una mezcla de 3-coordenadas en el cubo. Siendo el color negro representado por (0, 0, 0) y el color blanco por (255, 255, 255). (Gonzales & Woods, 1996).

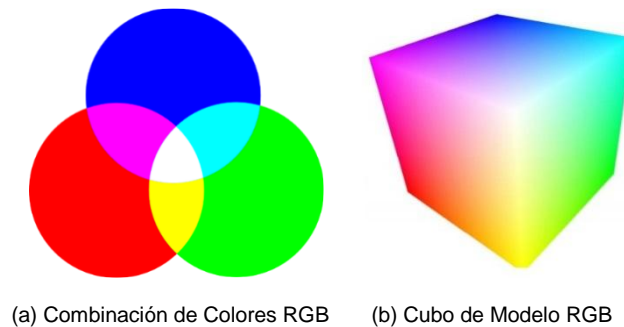


Fig. 1: Representación del modelo RGB.

Fuente: Gonzales & Woods, 1996

1.2.2. Segmentación de imágenes

La segmentación subdivide una imagen en sus partes constituyentes u objetos, con el fin de separar las partes de interés del resto de la imagen, por lo tanto el nivel al que se lleva a cabo esta subdivisión depende del problema por resolver. En el proceso de detectar las partes en una imagen, se identifican los bordes de la imagen, o se segmenta en regiones, líneas o curvas, etc.

Otra definición considera a la segmentación como la clasificación de los puntos de la imagen (píxeles), indicando las clases a la que pertenecen los diferentes píxeles. Los atributos básicos de segmentación de imagen son: la luminancia en imágenes monocromáticas, los componentes de color en imágenes en color, textura, forma, etc.

La segmentación automática es una de las tareas más difíciles del procesamiento de imágenes. Esta etapa determina el eventual éxito o fracaso del análisis, de hecho, rara vez, llega a alcanzar una solución satisfactoria, se debe buscar un método alternativo de comprobación para la verificación de los resultados. (Gonzales & Woods, 1996).

1.2.3. Umbralización de imágenes

La técnica más sencilla de umbralización es la partición del histograma de una imagen utilizando un umbral único, la segmentación se completa entonces escaneando la imagen pixel a pixel y etiquetando cada pixel como perteneciente al objeto o al fondo, dependiendo de que el nivel de gris, de ese pixel, sea mayor o menor que el valor del umbral.

Supóngase que una imagen contiene solamente dos regiones de brillo principales. El histograma de una imagen de este tipo puede considerarse como una estimación de la función de densidad de probabilidad del brillo. Esta función de densidad total es la suma o mezcla de dos densidades uni-modales, una para las regiones claras de la imagen y la otra para la oscuras. Además, los parámetros de la mezcla son proporcionales a las superficies de cada región de brillo. Si se conoce o se supone la forma de las densidades, se puede determinar un umbral óptimo (en términos del error mínimo) por segmentación de la imagen en las dos regiones de brillo. Basándose, en lo antes expuesto, la umbralización se puede contemplar como una operación que implica realizar comprobaciones frente a una función T de la forma:

$$T = T[x, y, p(x, y), f(x, y)] \quad (1)$$

Donde $f(x, y)$ es el nivel de gris del punto (x, y) , y $p(x, y)$ representa alguna propiedad local de este punto – por ejemplo, la media del nivel de gris de una vecindad centrada en (x, y) – una imagen umbralizada $g(x, y)$ se define como:

$$g(x,y) = \begin{cases} 1 & \text{si } f(x,y) > T \\ 0 & \text{si } f(x,y) \leq T \end{cases} \quad (2)$$

De este modo, los pixeles marcados con 1 (o con cualquier otro nivel de intensidad conveniente) corresponden a objetos, mientras que los pixeles marcados con 0 corresponden al fondo. Cuando T depende solamente de $f(x,y)$, el umbral se denomina global. Si T depende a la vez de $f(x,y)$ y $g(x,y)$, el umbral se denomina local. Si, además, T depende de las coordenadas espaciales x e y , el umbral se denomina dinámico. (Gonzales & Woods, 1996).

1.2.4. Transformaciones proyectivas

La geometría proyectiva 2D es el estudio de las propiedades del plano proyectivo P^2 que son invariantes bajo un grupo de transformaciones conocidas como proyectividades. Una proyectividad es una transformación invertible dada por $h: P^2 \rightarrow P^2$, de manera tal, que una línea recta es transformada como una línea recta. La proyectividad está definida como:

$$h(\mathbf{m}) = \mathbf{m}' = \mathbf{H}\mathbf{m} \quad (3)$$

Donde \mathbf{H} es una matriz 3 x 3 no singular. Se dice entonces que \mathbf{m}' es la transformación lineal \mathbf{H} de \mathbf{m} . Esta transformación es biunívoca entre dos planos 2D, cuyos puntos son representados homogéneamente por \mathbf{m} y \mathbf{m}' . Es decir, un punto en un plano 2D tiene una única correspondencia en un punto de otro plano 2D, y cada punto en un plano tiene un solo punto correspondiente en el otro plano.

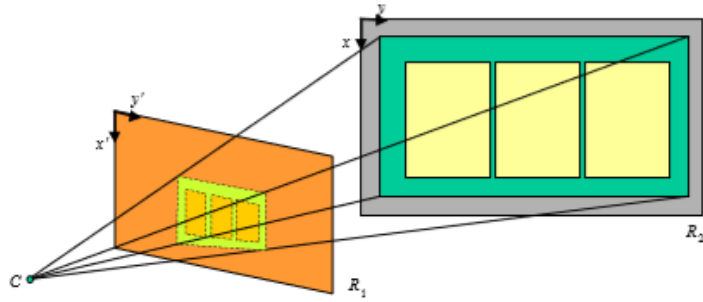


Fig. 2: Proyección de dos planos en paralelo.
Fuente: Mery, 2008

La condición invariante (una recta es transformada en una recta) puede comprobarse de la siguiente manera: si m_1, m_2, m_3 , están en la misma línea recta l , entonces $l^T m_i = 0$ para $i = 1, 2, 3$. Esta ecuación puede ser escrita como $l^T H^{-1} m_i = 0$, ya que $H^{-1} H = I$. La ecuación 3 puede escribirse como:

$$\begin{bmatrix} x'_1 \\ x'_2 \\ x'_3 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \quad (4)$$

$$x' = \frac{x'_1}{x'_3} = \frac{h_{11}x + h_{12}y + h_{13}}{h_{31}x + h_{32}y + h_{33}} \quad (5)$$

$$y' = \frac{x'_2}{x'_3} = \frac{h_{21}x + h_{22}y + h_{23}}{h_{31}x + h_{32}y + h_{33}} \quad (6)$$

En el problema interesa encontrar los nueve elementos de H . Sin embargo, como la matriz H es homogénea, kH también sería solución al problema. De esta manera es posible dividir cada elemento de H por h_{33} para obtener matriz H con solo 8 elementos desconocidos, ya que el último elemento sería igual a uno. (Mery, 2008)

1.3. Definición de términos básicos

1.3.1. Pizarra interactiva

También denominada pizarra digital, consiste en una computadora conectada a un proyector, que muestra una señal de dicha computadora sobre una superficie lisa y rígida, sensible al tacto o no, desde que se puede controlar el ordenador, hacer anotaciones manuscritas sobre cualquier imagen proyectada. Su principal función es controlar la computadora mediante dicha superficie con un bolígrafo, o el dedo (cuando se trata de tecnología táctil) como si se tratara del mouse, ofreciendo la posibilidad de interactuar con la imagen.

1.3.2. Luminancia

Indica la cantidad de energía luminosa que detecta el ojo mirando la superficie desde un ángulo particular. La luminancia es, por tanto, un indicador de qué tan brillante aparecerá la superficie. Según la definición de Albert Munsell, la luminancia o luminosidad es la cualidad por la cual distinguimos un color claro de un color oscuro.

1.3.3. Distorsión perspectiva

Es la transformación que sufre un objeto y su entorno circundante debido a la proximidad del mismo respecto al objetivo. Un objeto cercano sacado con un angular parecerá anormalmente más grande de lo que debería. Pero esta distorsión es debida a la proximidad o lejanía de la cámara con respecto al objeto.

CAPÍTULO II METODOLOGÍA

2.1. MATERIALES

2.1.1. Logitech Quickcam 9000 pro

Los criterios tomados en cuenta para utilizar la cámara se basa en sus características técnicas, que brinda diferentes recursos para la captura de imágenes y video, que resulta favorable ante los distintos procesos dentro de proyectos académicos de investigación.



Fig. 3: Webcam utilizada para adquisición de imágenes
Fuente: Logitech.com

La decisión de utilizar este dispositivo se da por recomendación y experiencias de personas que trabajaron con el mismo dispositivo para trabajos de procesamiento de imágenes con buenos resultados. A continuación, se muestran las características técnicas que posee la cámara elegida.

Tabla 2: Especificaciones técnicas webcam

Camera Specifications:	
Connection Type	Corded USB
Microphone	Built-in, Noise Cancellation
Lens and Sensor Type	Glass, CMOS
Focus Type	Auto
Field of View (FOV)	75° Diagonal FOV
Focal Length	2 mm
Optical Resolution (True)	2 MP (1600x1200)
Image Capture (4:3 SD)	320x240, 640x480, 800x600, HD (960x720), 1.3 MP (1280x1024), 2 MP (1600x1200) (JPG - True)3 MP (2048x1536), 4 MP (2304x1728), 5 MP (2592x1944), 8 MP (3264x2448) (JPG - Software Enhanced)
Image Capture (16:9 W)	320x180, 640x360, 800x450, 720p HD (1280x720) (JPG - True)
Video Capture (4:3 SD)	320x240, 640x480, 800x600, HD (960x720), 1.3 MP (1280x1024), 2 MP (1600x1200) (WMV - True)
Video Capture (16:9 W)	320x180, 640x360, 800x450, 720p HD (1280x720) (WMV - True)
Frame Rate (max)	30 fps @ 800x600, 15 fps @ HD 960x720 (Hardware Limit)

Fuente: Logitech.com

2.1.2. Proyector

Es un aparato que recibe una señal de video y proyecta la imagen correspondiente, en una pantalla de proyección, usando un sistema de lentes, permitiendo así mostrar imágenes fijas o en movimiento. Todos los proyectores de video utilizan una luz muy brillante para proyectar la imagen, y los más modernos pueden corregir curvas, borrones y otras inconsistencias a través de los ajustes manuales.

La señal de video puede provenir de diferentes fuentes, como un sintonizador de televisión (terrestre o vía satélite), una computadora, una laptop, etc.

Los Proyectores utilizados para el desarrollo de la presente tesis son los mismos con los que cuenta la universidad, utilizados normalmente para dictar las clases en las aulas y laboratorios.

2.1.3. Puntero LED

Puntero LED, (o LED Pen en inglés) se refiere a un puntero emisor de luz por medio de un LED instalado en el extremo del dispositivo, generalmente en forma de lapicero. El dispositivo LED es de la gama ultra brillante que emite luz visible, y cuenta con un botón para activar/desactivar. Su fuente de voltaje es, a base de pilas (de diferentes tipos según se prefiera) aunque también se han encontrado prototipos mediante conexión a USB.

2.2. Metodología

El funcionamiento del sistema requiere registrar en tiempo real mediante la cámara web la actividad del puntero LED que realice el usuario sobre la proyección y de esta manera, definir la posición del objeto de interés

y convertir dichas señales en comandos de control de eventos del mouse en el computador. A continuación, se presentarán los métodos utilizado para cada uno de los procesos a ejecutarse en el proyecto.

2.2.1. Segmentación por umbralización global para sustracción de fondo dinámico

La umbralización es uno de los métodos más importantes de la segmentación de imágenes. Los procedimientos básicos de este tipo de segmentación son el umbralizado (Thresholding) y el umbralizado basado en histograma (Histogramming).

La más sencilla de todas las técnicas es la partición del histograma de una imagen utilizando un umbral único, la segmentación se completa entonces recorriendo pixel a pixel y etiquetando cada pixel como parte perteneciente al objeto o al fondo, dependiendo del nivel de gris de ese pixel sea mayor o menor al valor de umbral

Considerando la imagen original como $f(x,y)$, donde T_L es el valor de luminancia utilizado como umbral, la imagen resultante $g(x,y)$ se define por:

$$g(x,y) = \begin{cases} 1 & \text{si } f(x,y) \geq T_L \\ 0 & \text{si } f(x,y) < T_L \end{cases} \quad (7)$$

Se debe tener en cuenta que la imagen $g(x,y)$ será una imagen binarizada que puede contener más de un objeto. (Gonzales & Woods, 1996).

2.2.2. Cálculo de luminancia basado en modelo de color RGB

La luminancia (sección 1.3.2) es comúnmente utilizada bajo el modelo de color HSL, que define el color, en términos, de sus componentes constituyentes y representado gráficamente como un cono doble, donde la Luminancia L corresponde a la distancia entre el eje blanco-negro.

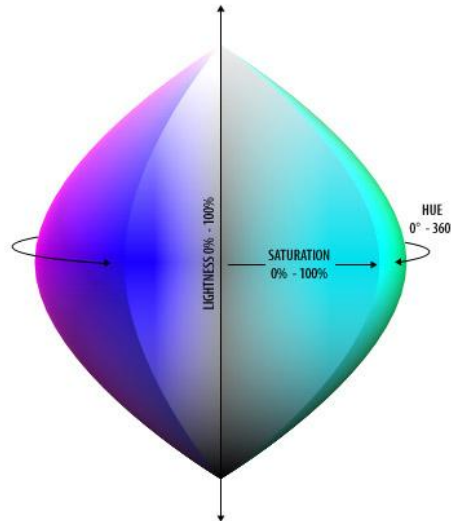


Fig. 4: Modelo de color HSL.

Fuente: http://dba.med.sc.edu/price/irf/Adobe_tg/models/hsb.html

Para la presente tesis, se decidió basar la segmentación en un umbral de valor de luminancia T_L , tal como se muestra en la Ecuación 8, pero definido sobre el modelo de color RGB (Sección 1.2.1) en vez de HSL debido a que el dispositivo de adquisición de imágenes captura sobre este modelo, para ellos se define la Luminancia L como el promedio entre el mayor y menor componente de color RGB. (Gonzales & Woods, 1996)

$$L = \frac{1}{2} (MAX(RGB) + MIN(RGB)) \quad (8)$$

2.2.3. Etiquetado de objetos por búsqueda de conectividad para estimación de objetos detectados

Para la evaluación de áreas de los objetos detectados es necesario agrupar los píxeles en objetos o componentes para identificar, mediante la evaluación de las áreas respectivas, cual es el objeto de interés.

El presente método consiste en establecer los componentes conectados de una imagen binaria asignándole una etiqueta diferente (normalmente número entero) a cada componente u objeto de la imagen. A continuación se indica el algoritmo básico utilizado (Ramirez Manzanarez):

1. Se inicializan las etiquetas de todos los píxeles que pertenecen a la clase objeto a -1. Se inicializa un identificador secuencial de etiquetas a $C = 0$. Se inicializa una lista de equivalencia de etiquetas a vacía ($LE = \emptyset$).
2. Se recorre la imagen renglón a renglón de arriba a abajo y de izquierda a derecha, hasta encontrar un píxel de p que no tiene etiqueta asignada, es decir, $E(p) = -1$.
3. Se establecen los cuatro posibles vecinos que ya han sido visitados por el algoritmo. Y por lo tanto que ya han sido etiquetados. Esos 4 vecinos son: izquierda, arriba, diagonal izquierda-arriba y diagonal derecha-arriba.
4. Si ninguno de los 4 vecinos tiene un identificador de etiqueta, se asigna la nueva etiqueta actual al punto $p(E(p) = C)$ y se incrementa el número de etiqueta actual: $C = C + 1$.
5. Si solo uno de los 4 vecinos de q tiene un identificador de etiqueta, entonces asignar esta etiqueta actual $E(p) = E(q)$.
6. Si hay diferentes etiquetas en los 4 vecinos asignar a p la etiqueta de cualquiera de ellos, digamos la de r . Ya que estas regiones están

conectadas tomar nota de la equivalencia, es decir, agregar a la lista estas nuevas equivalencias, por ejemplo $EL.add(E(p) == E(q))$.

7. En el momento en que no se encuentran más puntos para procesar se ha terminado la asignación.
8. Se debe de recorrer la lista de equivalencias y se asigna una nueva etiqueta para todas las etiquetas anteriores que son equivalentes.
9. Se recorre la imagen nuevamente y para cada etiqueta vieja se le asigna una nueva que junta componentes equivalentes.
10. Este algoritmo tiene la desventaja que hay que recorrer la imagen dos veces respecto al rastreador de contorno que etiqueta los componentes de una sola pasada. Sin embargo, se tomó en cuenta este algoritmo ya que representa un procedimiento más simple.

2.2.4. Segmentación por umbralización local para eliminación de interferencias

Una de las características más simples e intuitivas de un objeto es el área, teniendo en cuenta que el área dentro de una imagen binarizada y/o etiquetada $g(x, y)$ el área A es la cantidad de pixeles de color blanco que compone el objeto, que define un umbral local T_A basado en la medida del área y obtiene una imagen $h(x, y)$ definida por:

$$A = \sum_x \sum_y g(x, y), \quad \forall g(x, y) \geq 1 \quad (9)$$

$$h(x, y) = \begin{cases} 0 & \text{si } Area \leq T_A \\ 1 & \text{si } Area > T_A \end{cases} \quad (10)$$

2.2.5. Cálculo de centroide basado en peso promediado de pixeles para determinar posición

El programa deberá obtener la posición del puntero LED en tiempo real y obtendrá dichas coordenadas mediante el centroide del objeto de interés en $h(x, y)$, resultado del procesamiento de imágenes.

El método convencional del algoritmo para el cálculo de centroide se basa en el centro de masa, y es conocido como el peso promediado de pixeles de objeto de interés de una imagen binarizada, cuyos valores de coordenadas estarán definidos por:

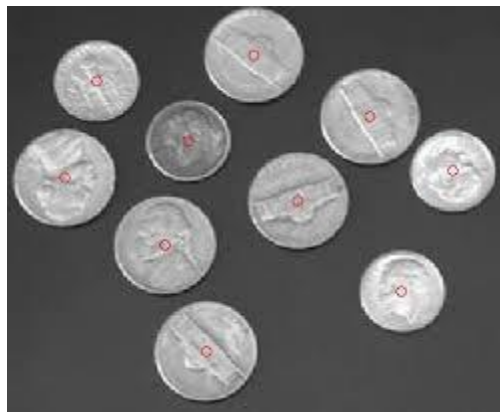


Fig. 5: Centroides de un objeto.

Fuente: <http://www.mathworks.com/matlabcentral>

$$C_x = \frac{\sum \text{Coordenadas pixeles en } x}{\text{Cantidad de pixeles del objeto}} \quad (11)$$

$$C_y = \frac{\sum \text{Coordenadas pixeles en } y}{\text{Cantidad de pixeles del objeto}} \quad (12)$$

Teniendo en cuenta que la cantidad de pixeles del objeto equivale a su área, se define el centroide mediante:

$$C_x = \frac{\sum_x \sum_y x \cdot h(x, y)}{\sum_x \sum_y h(x, y)} \quad (13)$$

$$C_y = \frac{\sum_x \sum_y y \cdot h(x, y)}{\sum_x \sum_y h(x, y)} \quad (14)$$

2.2.6. Cálculo de homografías para corrección de distorsión proyectiva en adquisición de imágenes

Las homografías son transformaciones proyectivas desde un espacio a otro. Este método puede ser usado para adjuntar imágenes panorámicas o para transformaciones de puntos capturados por una cámara en el mundo real. La homografía se estima entre imágenes mediante la búsqueda de sus correspondencias de entidad en esas imágenes. La matriz H de homografía es definida como una matriz 3x3 y puede ser expresada como una multiplicación de matrices. (Hartley & Zisserman, 2004)

Hay un punto $A(x_a, y_a)$, que es un punto del objeto O. el punto $B(x_b, y_b)$ está relacionado con el mismo objeto O, pero este es capturado desde una vista diferente (Otro sistema de coordenadas). La matriz H está definida por:

$$\begin{bmatrix} x_a \\ y_a \\ 1 \end{bmatrix} = \begin{bmatrix} x_a'/m \\ y_a'/m \\ m/m \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x_b \\ y_b \\ 1 \end{bmatrix} \quad (15)$$

$$\begin{bmatrix} \text{sacos}\phi & \text{sacos}\phi & t_z \\ \text{sacos}\phi & \text{sacos}\phi & t_y \\ p_0 & p_1 & 1 \end{bmatrix} \quad (16)$$

$$x'_a = h_{11}x + h_{12}y + h_{13} \quad (17)$$

$$y'_a = h_{21}x + h_{22}y + h_{23} \quad (18)$$

$$m = h_{31}x + h_{32}y + h_{33} \quad (19)$$

$$\frac{(x'_a)}{m} - \frac{(h_{11}x + h_{12}y + h_{13})}{(h_{31}x + h_{32}y + h_{33})} = 0 \quad (20)$$

$$\frac{(y'_a)}{m} - \frac{(h_{21}x + h_{22}y + h_{23})}{(h_{31}x + h_{32}y + h_{33})} = 0 \quad (21)$$

$$(x'_a/m)(h_{31}x + h_{32}y + h_{33}) - (h_{11}x + h_{12}y + h_{13}) = 0 \quad (22)$$

$$(y'_a/m)(h_{31}x + h_{32}y + h_{33}) - (h_{21}x + h_{22}y + h_{23}) = 0 \quad (23)$$

La matriz codifica las transformaciones geométricas en los parámetros en la Ecuación 16, ángulo de rotación ϕ , escala s , distorsión a , b y traslaciones t_z , t_y . La matriz de Homografía puede ser evaluada usando un conjunto homogéneo de ecuaciones lineales, tal como se define en las ecuaciones 24 y 25.

$$Ah = \begin{bmatrix} x & y & 1 & 0 & 0 & 0 & -x'x & -x'y \\ 0 & 0 & 0 & x & y & 1 & -y'x & -y'y \end{bmatrix} \begin{bmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32} \end{bmatrix} = \begin{bmatrix} x' \\ y' \end{bmatrix} \quad (24)$$

$$Ah = b, \quad Ah = 0, h \neq 0 \quad (25)$$

Para realizar la corrección basada en homografía se debe conocer al menos cuatro puntos con sus respectivas correspondencias entre los dos sistemas de coordenadas para ser evaluados mediante la matriz de homografía.

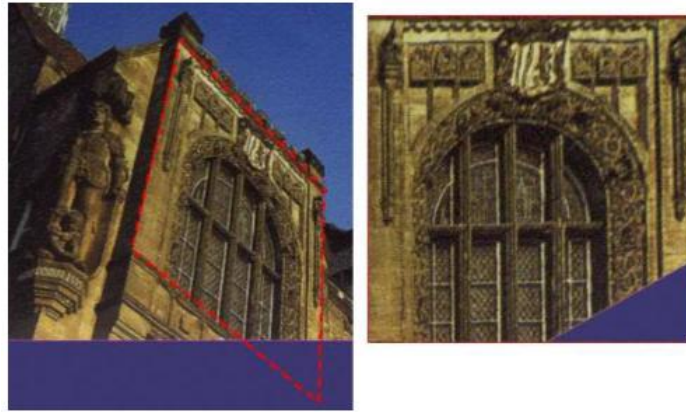


Fig. 6: Corrección de distorsión proyectiva.

Fuente: Mery, 2008

2.2.7. Interpolación lineal para concatenación de puntos

Durante el seguimiento del puntero LED para determinar su posición en tiempo real es posible que se tengan “huecos” entre un punto y otro, debido a la velocidad de movimiento del puntero respecto a la velocidad de captura de la webcam (más el tiempo de ejecución del programa).

La Interpolación lineal es el método más simple en uso hoy. La idea básica es conectar los 2 puntos dados en x_i , es decir (x_0, y_0) y (x_1, y_1) .

La función interpolante es una línea recta entre los dos puntos. Para cualquier punto entre los dos valores de x_0 y x_1 se debe seguir la ecuación de la línea, que se puede derivar geoméricamente, donde el único valor desconocido es y , despejando queda:

$$\frac{y - y_0}{y_1 - y_0} = \frac{x - x_0}{x_1 - x_0} \quad (26)$$

$$y = y_0 + (x - x_0) \frac{y_1 - y_0}{x_1 - x_0} \quad (27)$$

2.2.8. Control de eventos de mouse mediante uso de librerías de Windows MSDN

La biblioteca de desarrollo de Microsoft (MSDN Library) ofrece los formularios de Windows Form con la información técnica dirigida a desarrolladores de la plataforma Microsoft Windows. Para recibir y controlar la entrada del mouse y/o posición del cursor se utilizan las siguientes funciones:

- `Cursor.Position` – define la posición del cursor.
- `MOUSEEVENTF_LEFTDOWN` – define acción de botón IZQ activado.
- `MOUSEEVENTF_LEFTUP` – define acción de botón IZQ desactivado.

Estas funciones serán integradas en una función en lenguaje de programación C, que será convertida en un archivo DLL para utilizarlo desde Matlab. (Library, Eventos de mouse windows form, s.f.).

En referencia a los métodos indicados, podemos observar en el diagrama siguiente como se compone el sistema de métodos de la presente tesis.

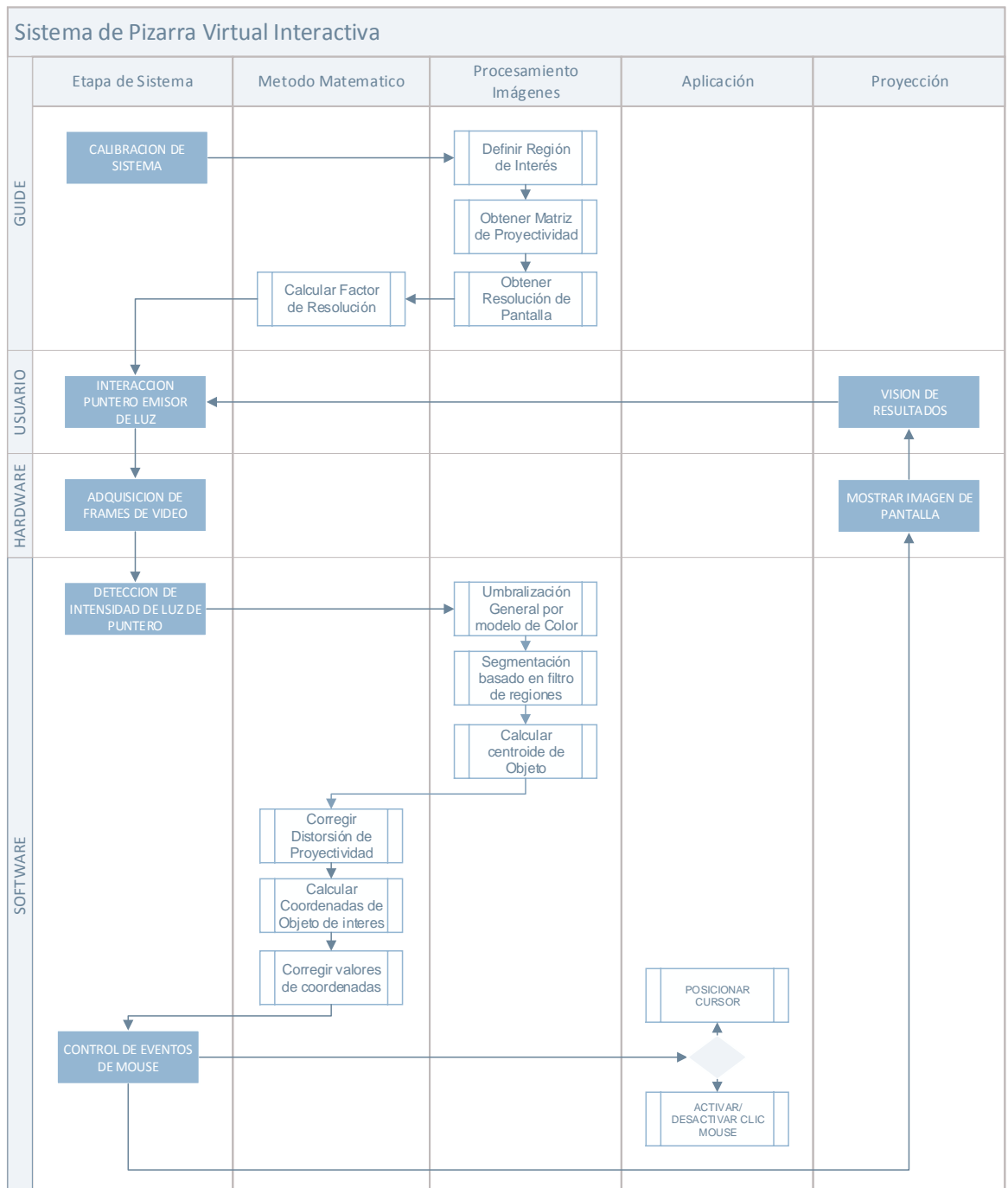


Fig. 7: Diagrama de bloques de sistema pizarra virtual

Elaboración: El autor

CAPÍTULO III

DESARROLLO DEL PROYECTO

Para el presente proyecto se deben desarrollar los algoritmos necesarios para obtener los datos suficientes que permitan lograr la interacción entre cada uno de los procesos que componen el sistema, utilizando los métodos descritos en el capítulo anterior.

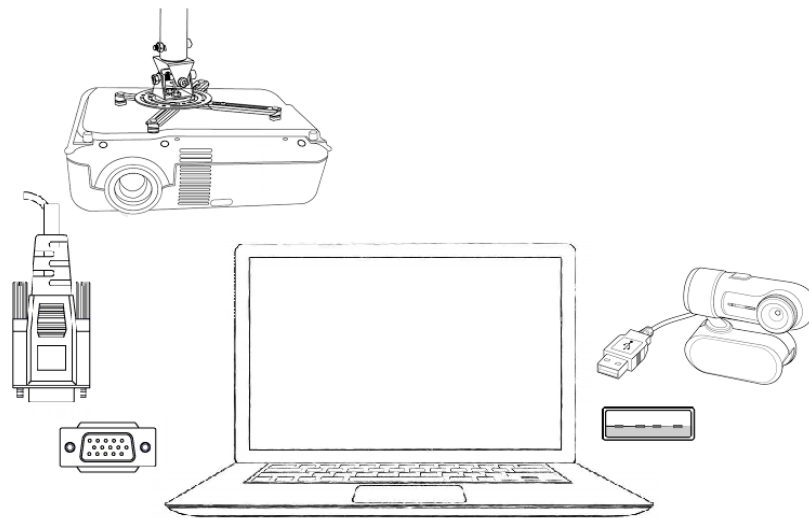


Fig. 8: Diagrama gráfico del sistema
Elaboración: El autor

El sistema propuesto será capaz de convertir cualquier superficie en una pizarra interactiva utilizando técnicas de procesamiento digital de imágenes para poder detectar las acciones del usuario con un puntero

emisor de luz sobre una proyección y traducirlo en comandos para el computador, permitiendo, de esta manera, que el usuario maneje funciones del computador a distancia mediante el control cursor y eventos del mouse.

A continuación, se presentan los diferentes procesos realizados durante el desarrollo del proyecto.

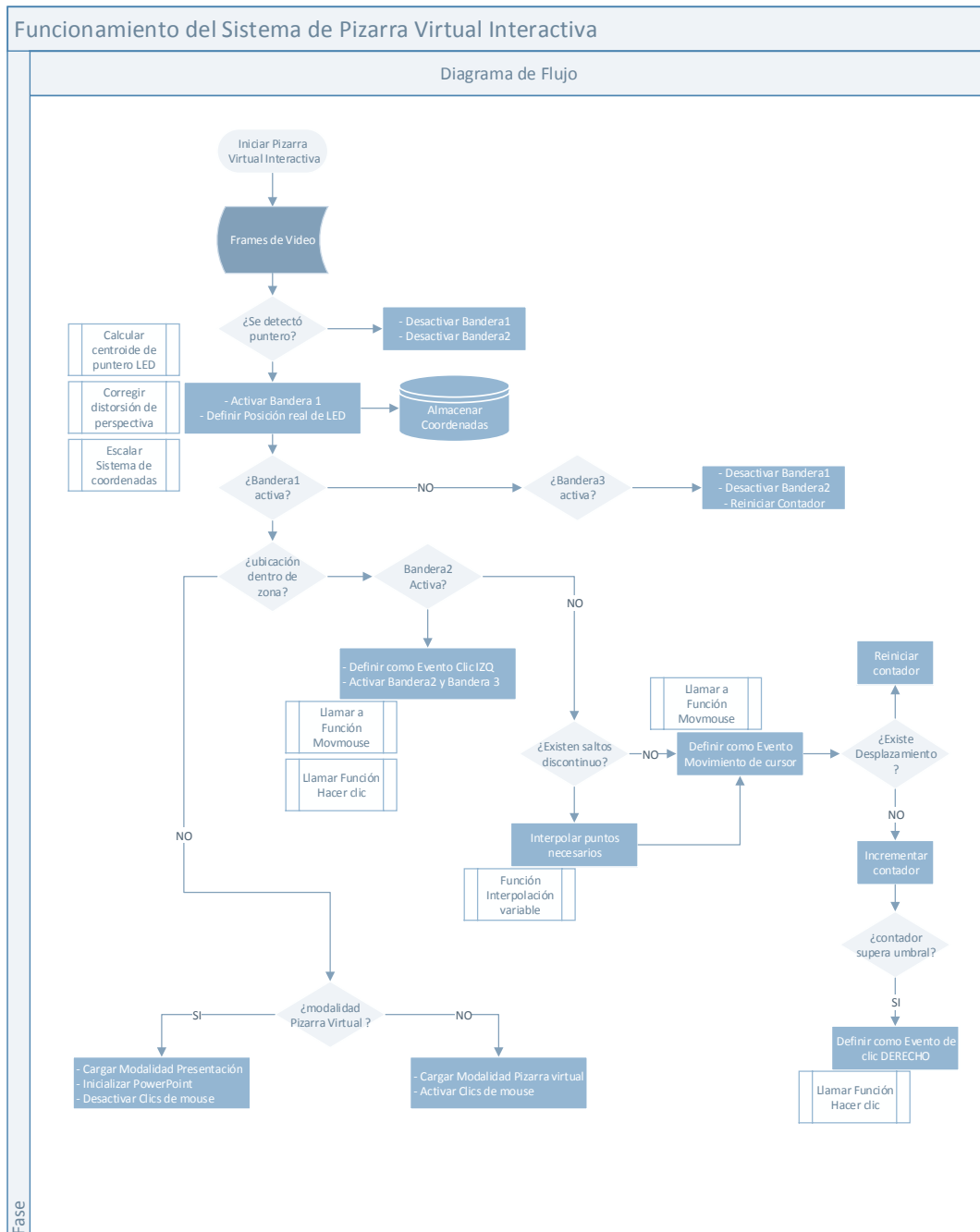


Fig. 9: Diagrama de flujo de algoritmo de pizarra virtual
Elaboración: El autor

El sistema cuenta con dos modalidades de funcionamiento: pizarra virtual y presentación. Esta última, dedicada a brindar al usuario mayor funcionalidad al momento de usar presentaciones utilizando Power Point.

3.1. Configuración de Logitech Quickcam

Este proceso se basa en modificar algunos valores de imagen que ofrece la webcam de manera que faciliten el procesamiento de imagen posterior.

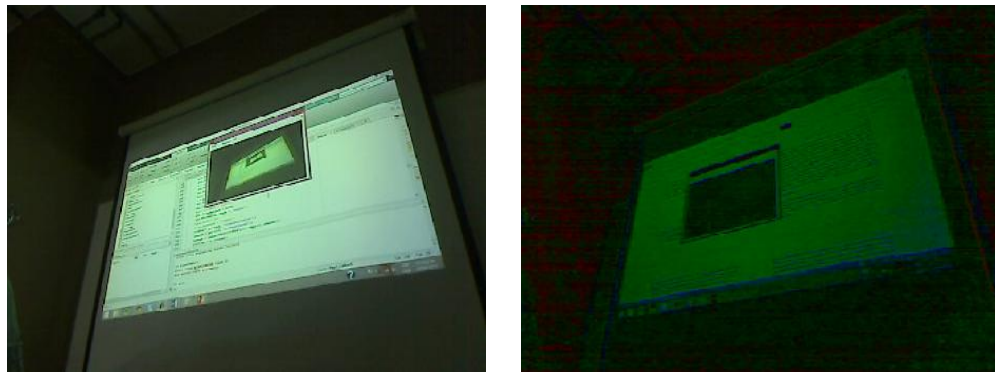
Para fines prácticos del sistema adquiere imágenes a una tasa de 30 cuadros por segundo en una resolución de 800x600 pixeles.

Se decidió utilizar dicha resolución para evitar carga computacional excesiva respecto a resoluciones superiores con las que la webcam cuenta, tal como se muestra:

Resolución Máxima Webcam:	$1680 \times 1200 = 2016000$	pixeles
Resolución utilizada:	$800 \times 600 = 480000$	pixeles
Diferencia de valores:	1536000	pixeles

Con lo que se demuestra el ahorro en carga computacional de aproximadamente un 76 % respecto a la máxima resolución. Teniendo en cuenta que la resolución elegida es suficiente para los procesos a realizarse, ya que no refleja una pérdida de información considerable para el sistema.

En la Figura 10 se muestra un ejemplo de los resultados obtenidos en el proceso de adquisición de imágenes.



(a) Imagen Original

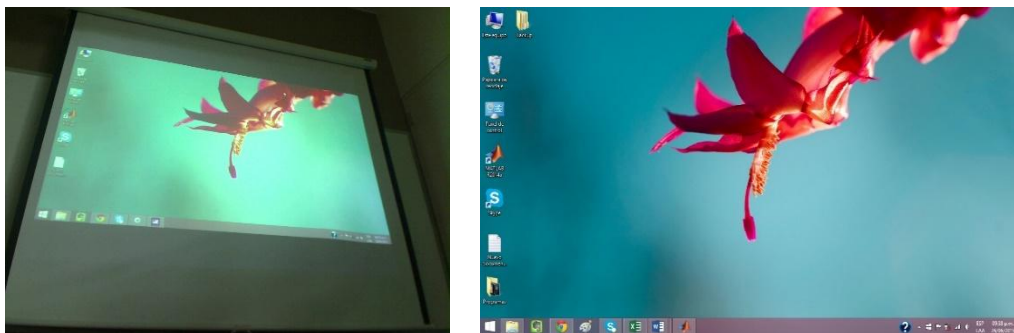
(b) resultado configuración valores de cámara

Fig. 10: Resultado de configuración de webcam

Elaboración: El autor

3.2. Calibración del sistema

Este proceso permite corregir los datos obtenidos por el sistema mediante los métodos de transformación proyectiva (Sección 2.2.6) (debido a que las imágenes capturadas poseen distorsión proyectiva) pasando dichos datos de un plano distorsionado a uno sin distorsión. Estos planos mencionados son el de la webcam y el de la pantalla del computador, que aparte de distorsión, también tienen distinta resolución.



(a) Plano de Cámara 800 x 600

(b) Plano de computador 1366 x 768

Fig. 11: Muestra de diferentes sistemas de coordenadas presentes

Elaboración: El autor

El proceso de calibración requiere la información de coordenadas de las esquinas de la zona de interés del plano con distorsión para posteriormente mediante métodos matemáticos de homografía se obtenga la matriz H de proyectividad, teniendo en cuenta la Ecuación 24 para los cuatro puntos definidos por la zona de interés. Se tiene:

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & -143 & -143 \\ 0 & 0 & 0 & 1 & 1 & 1 & -82 & -82 \\ 600 & 1 & 1 & 0 & 0 & 0 & -116000 & -145 \\ 0 & 0 & 0 & 600 & 1 & 1 & -21420 & -357 \\ 600 & 800 & 1 & 0 & 0 & 0 & -355800 & -474400 \\ 0 & 0 & 0 & 600 & 800 & 1 & -194400 & -259200 \\ 1 & 800 & 1 & 0 & 0 & 0 & -526 & -420800 \\ 0 & 0 & 0 & 1 & 800 & 1 & -122 & -97600 \end{bmatrix} * \begin{bmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32} \end{bmatrix} = \begin{bmatrix} 143 \\ 82 \\ 145 \\ 357 \\ 593 \\ 324 \\ 526 \\ 122 \end{bmatrix}$$

$$\begin{bmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & -143 & -143 \\ 0 & 0 & 0 & 1 & 1 & 1 & -82 & -82 \\ 600 & 1 & 1 & 0 & 0 & 0 & -116000 & -145 \\ 0 & 0 & 0 & 600 & 1 & 1 & -21420 & -357 \\ 600 & 800 & 1 & 0 & 0 & 0 & -355800 & -474400 \\ 0 & 0 & 0 & 600 & 800 & 1 & -194400 & -259200 \\ 1 & 800 & 1 & 0 & 0 & 0 & -526 & -420800 \\ 0 & 0 & 0 & 1 & 800 & 1 & -122 & -97600 \end{bmatrix}^{-1} * \begin{bmatrix} 143 \\ 82 \\ 145 \\ 357 \\ 593 \\ 324 \\ 526 \\ 122 \end{bmatrix}$$

$$\begin{bmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32} \end{bmatrix} = \begin{bmatrix} -0.067 \\ 0.694 \\ 142.379 \\ 0.329 \\ 0.099 \\ 81.574 \\ -0.0007 \\ 0.0009 \end{bmatrix}, \text{ donde } H = \begin{bmatrix} -0.067 & 0.694 & 142.37 \\ 0.329 & 0.099 & 81.574 \\ -0.0007 & 0.0009 & 1 \end{bmatrix}$$

Con la matriz de proyectividad, utilizando la Ecuación 15, se puede definir para cada punto en el sistema de coordenadas con distorsión su punto respectivo en el otro sistema libre de distorsión y acotado a la zona de trabajo definida por las esquinas, previamente seleccionadas, se puede reconstruir la imagen, tal como se muestra en la Figura 12.

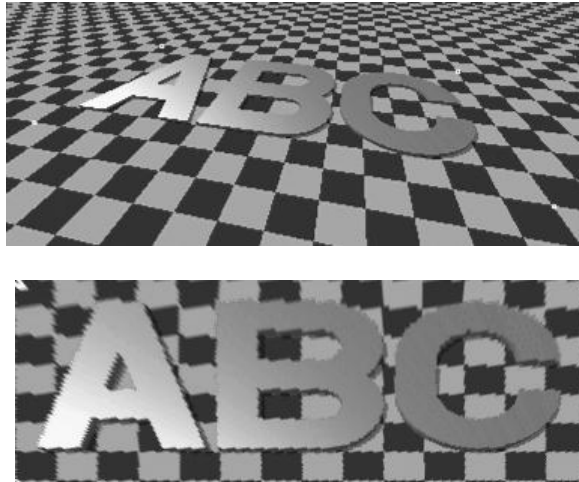


Fig. 12: Muestra de corrección de proyectividad
Fuente: Mery, 2008

Dentro del proceso de calibración se ha considerado dos métodos para la obtención de puntos de la zona de interés: de forma manual, indicada por el usuario sobre la imagen, o de forma automática, utilizando la segmentación por detección de bordes sobre el operador de Canny.

3.3. Algoritmo de sustracción de fondo para detección de puntero LED

Es común en proyectos de procesamiento de imágenes, normalmente se desea aislar el/los puntos de interés. En este, caso se tiene la particularidad que se tiene un fondo dinámico, lleno de distintos colores y brillo que es variable en el tiempo.

En la figura 13, se muestra el resultado del histograma de fondo que servirá para el proceso de análisis.

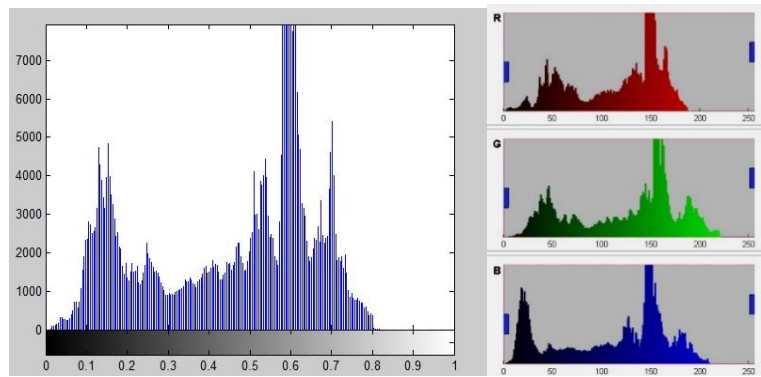


Fig. 13: Histograma de la luminancia y RGB del fondo
Elaboración: El autor

Teniendo en cuenta las ecuaciones 7 y 8 del capítulo anterior se obtendrá un valor de luminancia bajo el modelo RGB para utilizarlo como umbral, en base al análisis por Histograma, correspondiente a un 85%, donde la Imagen resultante $g(x, y)$ está definida por:

$$g(x, y) = \begin{cases} \text{si } L \rightarrow f(x, y) \geq 0.85, \text{ Punto pertenece al objeto de interes} \\ \text{si } L \rightarrow f(x, y) < 0.85, \text{ Punto pertenece al fondo de imagen} \end{cases}$$

Luego se procede a recorrer la imagen pixel a pixel y evaluando la luminancia según los valores RGB correspondientes a la coordenada.

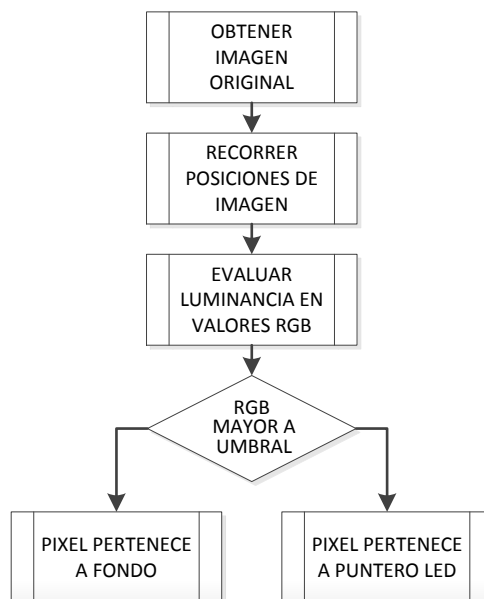


Fig. 14: Diagrama de flujo de umbralización
Elaboración: El autor

Se decidió utilizar un método de segmentación basado en luminancia, obtenida de la luz de un puntero LED, que será el objeto de interés dentro de las imágenes.

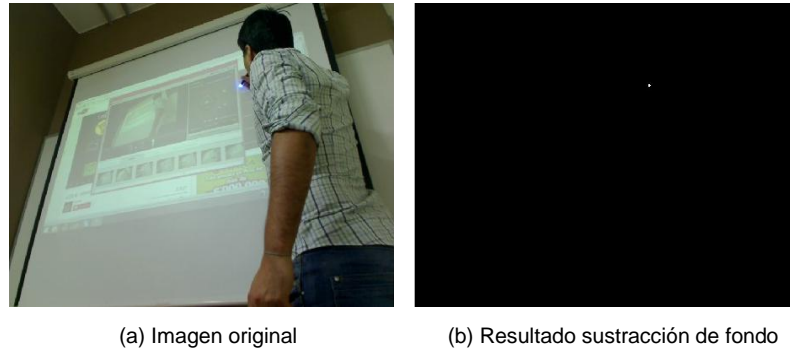


Fig. 15: Muestra de sustracción de fondo.
Elaboración: El autor

3.4. Algoritmo de evaluación de áreas para eliminación de interferencias

Cuando el proceso de sustracción de fondo tiene como resultado más de un objeto de interés, se requiere validar cuál de los objetos obtenidos pertenece a la luminancia producida por la luz del puntero led.

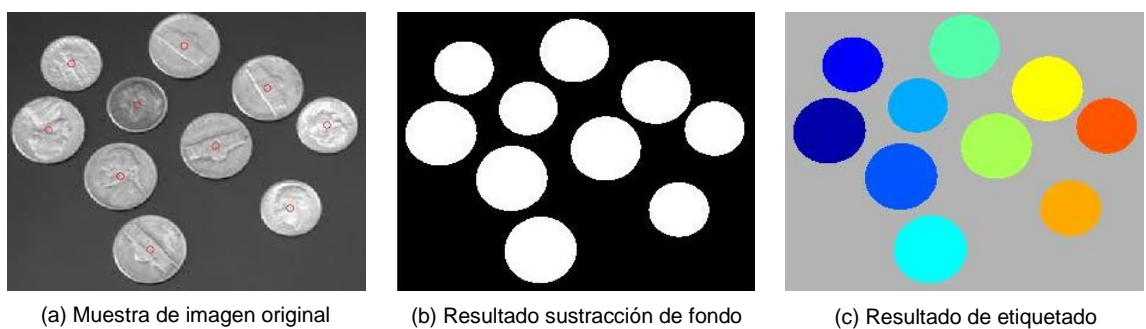


Fig. 16: Ejemplo de proceso de etiquetado
Fuente: <http://blogs.mathworks.com>

Para evaluar las propiedades de los objetos resultantes se utiliza el método de etiquetado de imágenes (Sección 2.2.3), se realiza el cálculo de áreas de los objetos y se eliminan los objetos que no son parte del puntero led por umbralización local (Sección 2.2.4).

Se establece el valor de umbral de área de objetos T_a en un rango de 0-5 píxeles, que es el rango donde están contenidas las interferencias.

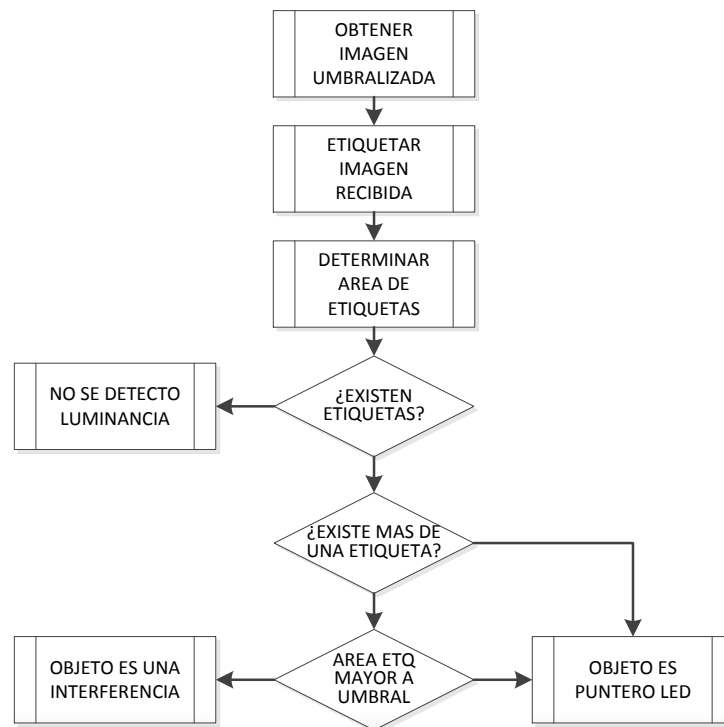
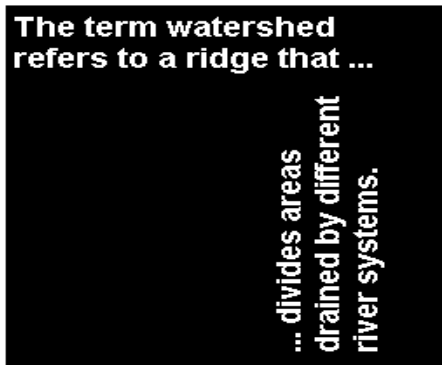


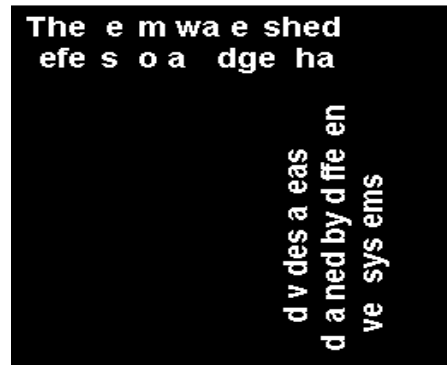
Fig. 17: Diagrama de flujo de eliminación de objetos
Elaboración: El autor

Una vez aislado el objeto de interés es necesario determinar su posición en el sistema de coordenadas correspondiente, es decir, en el plano de la cámara (resolución 600x800).

En la figura 18, se muestra un ejemplo de eliminación de objetos en una imagen según el valor de área, definida por la cantidad de píxeles que la componen.



(a) Imagen con objetos de distintos valores de área



(b) resultado de sustracción de objetos con áreas menores a 50 pixeles

Fig. 18: Ejemplo de proceso de eliminación de objetos

Fuente: <http://www.mathworks.com>

3.5. Algoritmo de evaluación de centroide para cálculo de posición

La posición del objeto de interés será definida mediante el cálculo matemático del centroide mediante el método de peso promediado de pixeles, tal como se muestra:

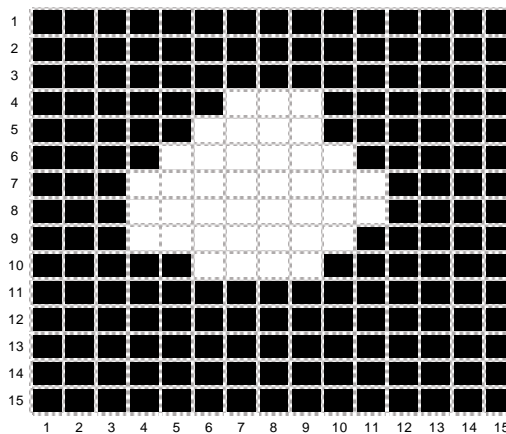


Fig. 19: Ejemplo de objeto de interés para cálculo de centroide

Elaboración: El autor

Área = 40 pixeles

\sum Coordenadas X = 291; \sum Coordenadas Y = 298

Siguiendo las ecuaciones 13 y 14, se obtiene:

Abscisa Centroide: $291/40 = 7.275$

Ordenada Centroide: $298/40 = 7.45$

La coordenada del centroide define la posición del objeto, que hasta el momento se encuentra en el sistema de coordenadas de la cámara, que presenta distorsión, por lo que será corregida, posteriormente, mediante proyectividad y luego escalado a los valores de la pantalla del computador que se esté utilizando para la proyección de la zona de trabajo, tal como se muestra:

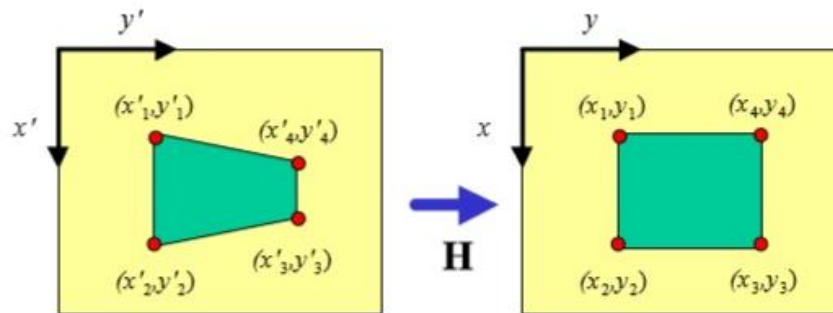


Fig. 20: Cambio de sistemas de coordenadas.

Fuente: Mery, 2008

La corrección por homografía se realiza utilizando la matriz de proyectividad hallada por el proceso de calibración del sistema, utilizando la Ecuación 15, se tiene para la coordenadas 300,300:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} * \begin{bmatrix} -0.067 & 0.694 & 142.37 \\ 0.329 & 0.099 & 81.574 \\ -0.0007 & 0.0009 & 1 \end{bmatrix} = \begin{bmatrix} 300 \\ 300 \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} -0.067 & 0.694 & 142.37 \\ 0.329 & 0.099 & 81.574 \\ -0.0007 & 0.0009 & 1 \end{bmatrix}^{-1} * \begin{bmatrix} 300 \\ 300 \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} -0.1175 & 2.5999 & -195.352 \\ 1.7740 & -0.1501 & -240.321 \\ -0.0017 & 0.0020 & 1.0795 \end{bmatrix} * \begin{bmatrix} 300 \\ 300 \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 549.36 \\ 246.85 \\ 1.1624 \end{bmatrix} = \begin{bmatrix} 549.36/1.1624 \\ 246.85/1.1624 \\ 1.1624/1.1624 \end{bmatrix}$$

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 473 \\ 212 \\ 1 \end{bmatrix}$$

El resultado nos brinda la posición (x, y) corregida del objeto, aun en el sistema de coordenadas de la cámara. Para utilizar esta posición en el sistema de coordenadas de la zona de interés (es decir, la resolución de pantalla) se realiza un cambio de escala de las coordenadas:

Coordenadas cámara con distorsión	: (200, 300)
Coordenadas cámara sin distorsión	: (473, 212)
Resolución de cámara	: 800 x 600
Resolución de pantalla	: 1366 x 768
Factor de escala en abscisas	: $768 / 600 = 1.28$
Factor de escala en ordenadas	: $1366 / 800 = 1.71$
Coordenada X escalada	: $473 * 1.28 = 605$
Coordenada Y escalada	: $212 * 1.71 = 362$

3.6. Algoritmo de control de funciones de mouse

El control de funciones de mouse requerido para lograr la interacción con la computadora por intermedio del mouse se basa en 2 puntos:

- Mover el cursor del mouse
- Activar los botones de mouse

Estas propiedades propias de Windows serán llamadas como funciones utilizando los elementos de la clase "MouseEvent" Y "SetCursorPos".

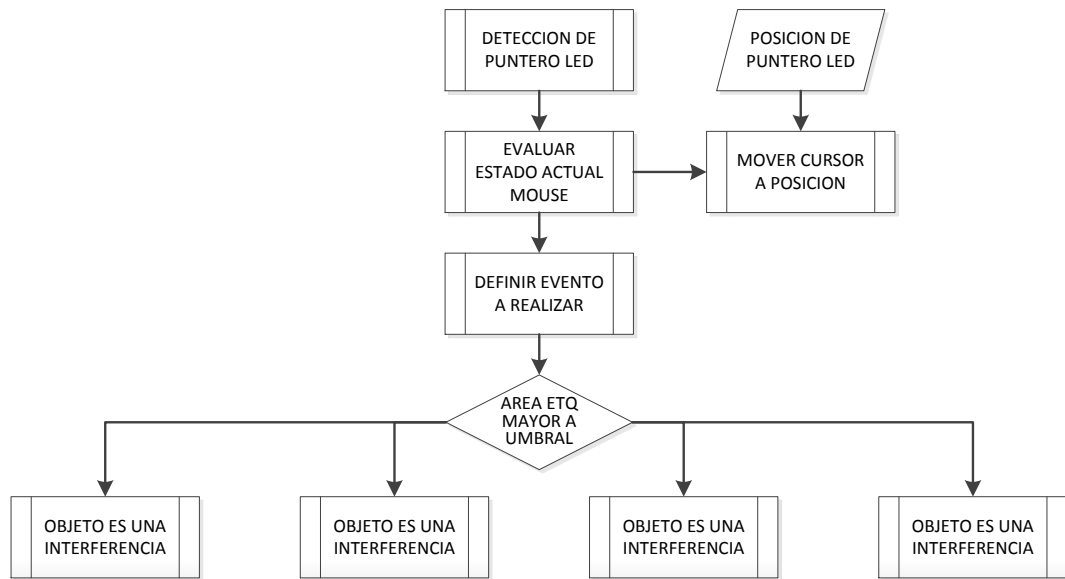


Fig. 21: Diagrama de flujo de control de mouse

Elaboración: El autor

El movimiento del cursor requiere las coordenadas del puntero para posicionar el cursor en el lugar que se desea, mientras que la elección de evento a ejecutarse se define por reconocimiento acciones realizadas con el puntero LED. La siguiente tabla muestra los tipos de eventos que ejecuta el programa.

Tabla 3: Detalle de eventos de mouse identificados

Evento	Descripción de Gestos
Clic Izquierdo	Detección de Luz (Evento por defecto)
Doble Clic	Clic izquierdo activado x2 sobre el mismo punto
Clic Derecho	Clic izquierdo activado continuo sobre el mismo punto (aproximadamente entre 1 a 2 segundos)
Arrastre	Clic izquierdo activado continuo + desplazamiento de cursor

Elaboración: El autor

3.7. Construcción puntero LED

Se requiere fabricar un puntero que genere luz para que pueda ser detectado por el programa, y a su vez, sea de fácil control para el usuario. Por ello se decidió fabricar un puntero en forma de bolígrafo (o plumón), tal como se muestra en la Figura 22, debido a que es un objeto comúnmente utilizado en clases o pizarras acrílicas.



Fig. 22: Prototipo de puntero led

Elaboración: El autor

El prototipo utiliza un LED de la gama Ultra brillante y se activa mediante un pulsador. En otros proyectos estudiados en el estado del arte es común utilizar luz infrarroja, para este proyecto se descartó el uso de LED infrarrojo debido a que requiere de un hardware especial para su detección, además, implementar una solución basada en infrarrojo tiene una desventaja al momento de la detección, ya que la luz solar consta de una luz infrarroja interferiría con el proceso de detección y solo se podría utilizar en habitaciones con poca luz solar.

Luego de los análisis realizados para cada uno de los procesos durante el presente capítulo, se definen los valores de cada variable, considerados como parámetros de diseño, que se describen en la siguiente tabla.

Tabla 4: Parámetros de diseño

Parámetros de Diseño	Definición	Valor
Umbral de Luminancia o Intensidad de Luz	L	220
Umbral de área de ruido	Ta	0-5 pixeles
Coordenada de centroide	(Cx, Cy)	Variable
Matriz de Proyectividad	H	Variable
Velocidad de captura	Frames de video	30 fps
Resolución de captura	resolución Webcam	600 x 800

Elaboración: El autor

CAPÍTULO IV

PRUEBAS Y RESULTADOS

En el presente capítulo se presentaran los resultados de las pruebas realizadas para cada uno de los procesos utilizados.

4.1. Proceso de calibración de sistema

La calibración del sistema cumple la función de obtener el parámetro necesario para la corrección de distorsión presente en el plano de la webcam al momento de la adquisición de imágenes.

Teniendo en cuenta las operaciones detalladas en la Sección 3.2. Para demostrar el correcto funcionamiento del algoritmo se realizó la reconstrucción de la imagen interpolando uno a uno los valores corregidos utilizando la matriz H para cada uno de los puntos dentro de la zona de interés indicada en el proceso.

En la figura 23, se muestra el proceso de calibración y su resultado como una imagen sin distorsión proyectiva y acotada en la zona de interés delimitada por el inicio de la calibración. Cabe indicar que para los intereses del proyecto no es necesario corregir toda la imagen, sino solo el punto de coordenada resultante del cálculo del centroide.

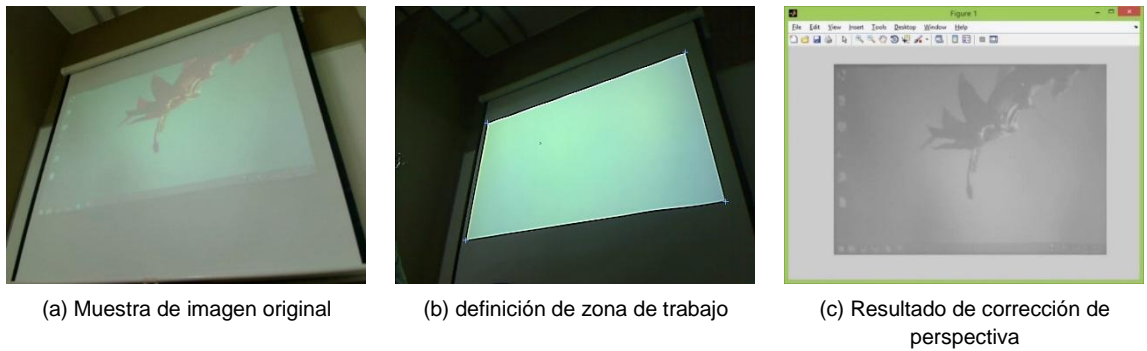


Fig. 23: Resultado de proceso de calibración
Elaboración: El autor

4.2. Proceso de detección de puntero

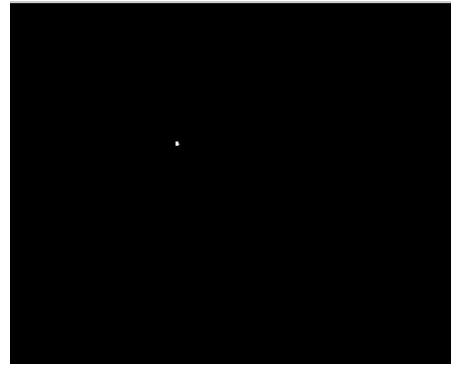
4.2.1. Sustracción de fondo

La detección del puntero se realiza mediante el algoritmo de sustracción de fondo, donde se aísla el objeto de interés luego del proceso de umbralización por evaluación de luminancia.



Fig. 24: Resultado prueba con diferentes valores de luminancia
Elaboración: El autor

Tal como se muestra en la figura 25, los resultados obtenidos del proceso de sustracción de fondo son indiferentes al color de la luz del LED, esto debido a que la detección no está basada en color sino en luminancia producida.



(a) Imagen con objetos de distintos valores de área (b) resultado de sustracción de objetos con áreas menores a 50 píxeles

Fig. 25: Resultado de proceso de sustracción de fondo
Elaboración: El autor

A continuación, se muestran los resultados obtenidos para diferentes valores de umbral (L) de luminancia.

Tabla 5: Detección de objetos vs valor de umbral

Nº Muestra	L= 190	L= 200	L= 210	L= 220
1	98	1	1	1
2	52	1	1	1
3	177	17	1	1
4	106	4	1	1
5	8	1	1	1
6	41	4	2	1
7	46	35	1	1
8	35	1	1	1

Elaboración: El autor

De la comparación de resultados obtenidos se puede apreciar en la figura 26 qué valor de umbral de luminancia aporta el mejor funcionamiento para el proceso actual, que se encuentra dentro del valor definido en el capítulo anterior.

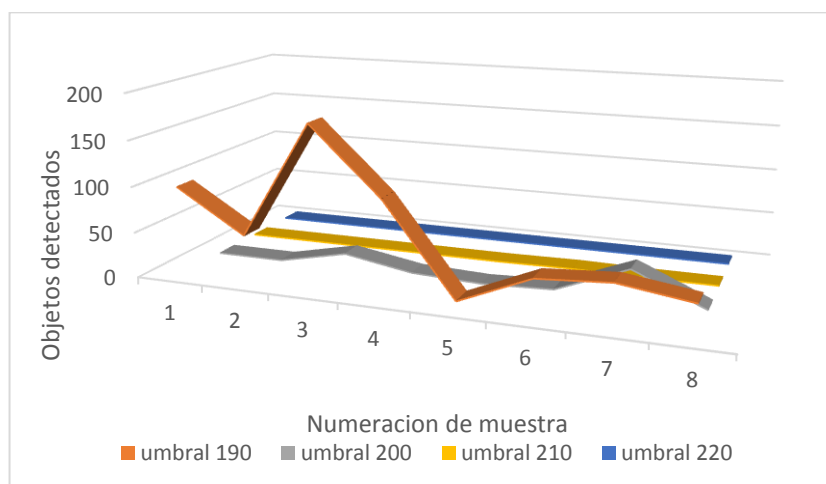


Fig. 26: Evaluación de resultados de sustracción de fondo
Elaboración: El autor

4.2.2. Eliminación de interferencias

La imagen resultante de la sustracción de fondo puede presentar interferencias por diversos motivos, es decir, detectar más de un objeto de interés en una misma imagen. Por ello se realiza el proceso de eliminación de objetos mediante la evaluación de áreas.



Fig. 27: Resultado de eliminación de interferencias
Elaboración: El autor

Para definir qué objetos son interferencias y cuáles no, se definió un rango de valor de área donde se encuentran las interferencias. Del análisis de los resultados de detección se determinó que las interferencias

presentes tiene un área menor en comparación con el objeto de interés. A continuación se aprecian los resultados de 15 muestras de detección.

Tabla 6: Evaluación de áreas (en píxeles) de objeto de interés

Distancia\ Muestra	1º	2º	3º	4º	5º	6º	7º	8º	9º	10º	11º	12º	13º	14º	15º
0.5m	34	39	19	23	16	63	36	35	72	51	20	19	10	8	13
m	4	22	21	23	8	21	11	4	22	21	16	11	4	15	11
1.5m	15	10	12	10	3	22	12	9	7	16	19	6	3	8	11
1.8 m	9	11	16	8	4	17	3	11	2	9	10	9	1	3	9

Elaboración: El autor

Para poder realizar una evaluación real, estas muestras tuvieron como variable la distancia de la webcam respecto a la zona de interés.

4.2.3. Proceso de cálculo de centroide

Los resultados del cálculo de centroide utilizando las ecuaciones correspondientes del algoritmo propuesto fueron comparados con los resultados arrojados por la función Centroid de Matlab para 6 distintas muestras, tal como se muestra:

Tabla 7: Comparación cálculo de centroide

Imagen	área en píxeles	Algoritmo centroide (Cx, Cy)	Matlab Centroid Function (Cx,Cy)
muestra 1	74 p	[337 315]	[336, 315]
muestra 2	51 p	[317 319]	[316, 318]
muestra 3	27 p	[482 163]	[482, 162]
muestra 4	47 p	[428 181]	[428, 181]
muestra 5	27 p	[410 197]	[410, 196]
muestra 6	43 p	[303 232]	[302, 232]

Elaboración: El autor

Tal como se aprecia en la tabla 7, los resultados obtenidos del algoritmo propuesto coinciden con los resultados de la herramienta de procesamiento de imágenes de Matlab.

4.2.4. Proceso de interpolación variable

Se propone implementar una función de interpolación variable para el programa, ya que con la interpolación regular, a pesar de funcionar correctamente, se presentan discontinuidades.

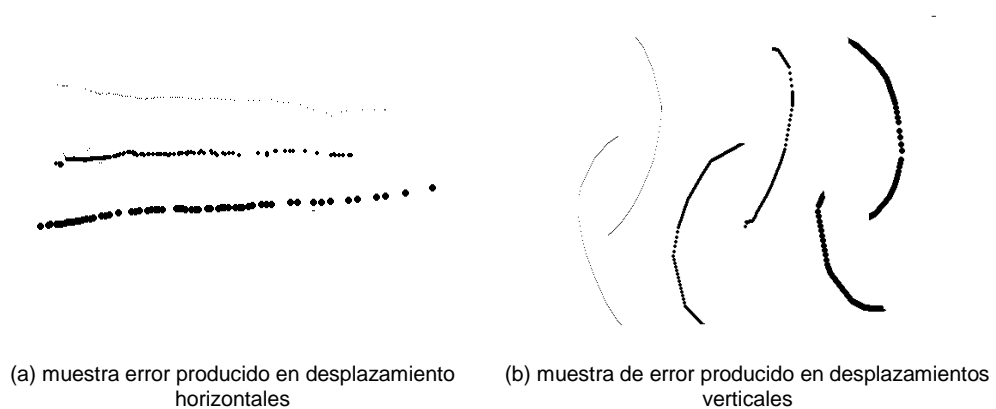


Fig. 28: Muestra de errores de interpolación regular
Elaboración: El autor

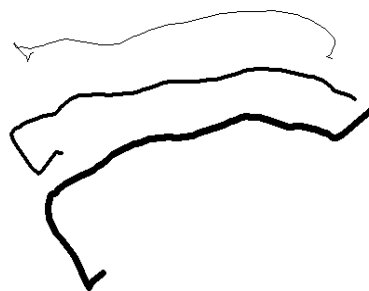


Fig. 29: Resultado de interpolación variable
Elaboración: El autor

El algoritmo de interpolación propuesto decide qué valor debe ser interpolado teniendo en cuenta el recorrido detectado, diferenciando si conviene interpolar los valores horizontales o verticales, tal como se muestra

en el diagrama de flujo de la figura 30. Esta función realiza la evaluación de los puntos de detección y sus coordenadas respectivas, llamando a las funciones del mouse correspondientes para cada punto interpolado, logrando un movimiento acorde con las acciones realizadas por del puntero.

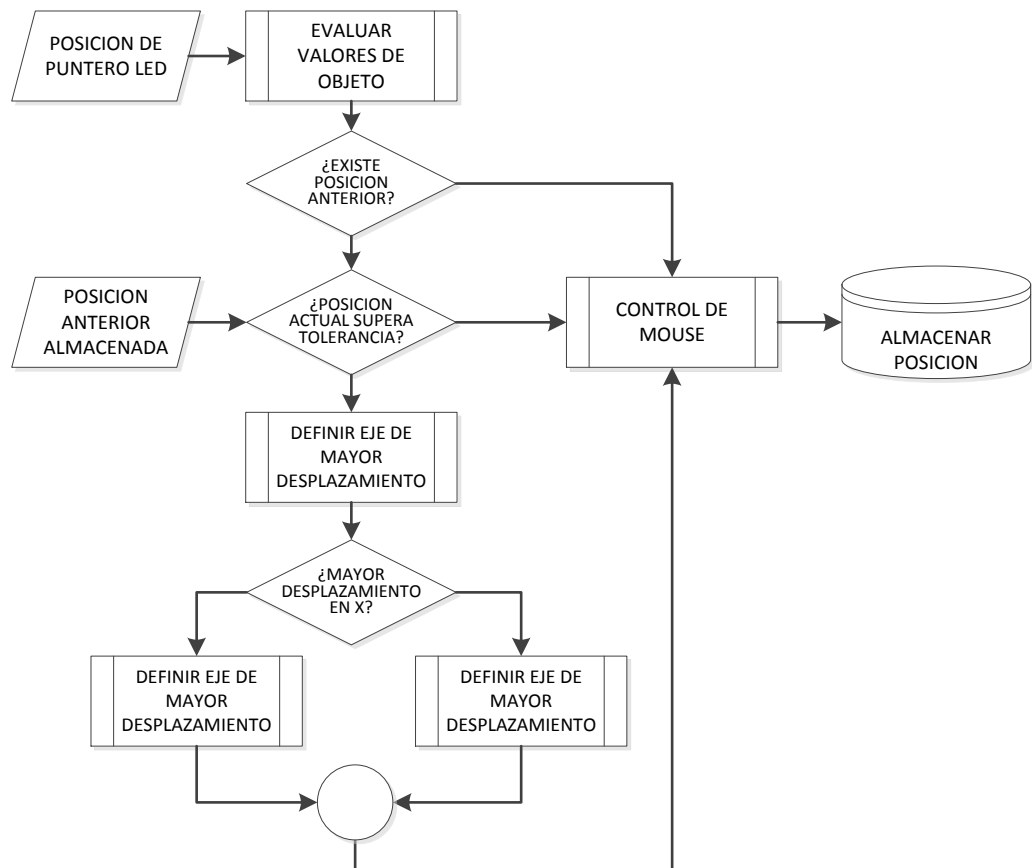


Fig. 30: Diagrama de flujo de proceso de interpolación variable
Elaboración: El autor

4.3. Pruebas de distancia de funcionamiento

Para el uso del sistema, se debe tener en cuenta la distancia de la cámara respecto a la proyección. Se ha determinado que la distancia mencionada influye directamente con la cantidad de luz detectada, variando el valor del área del puntero LED reconocido. Para las pruebas de distancia

se determinó evaluar la variación de áreas del objeto de interés según distancia y color de superficie de proyección.

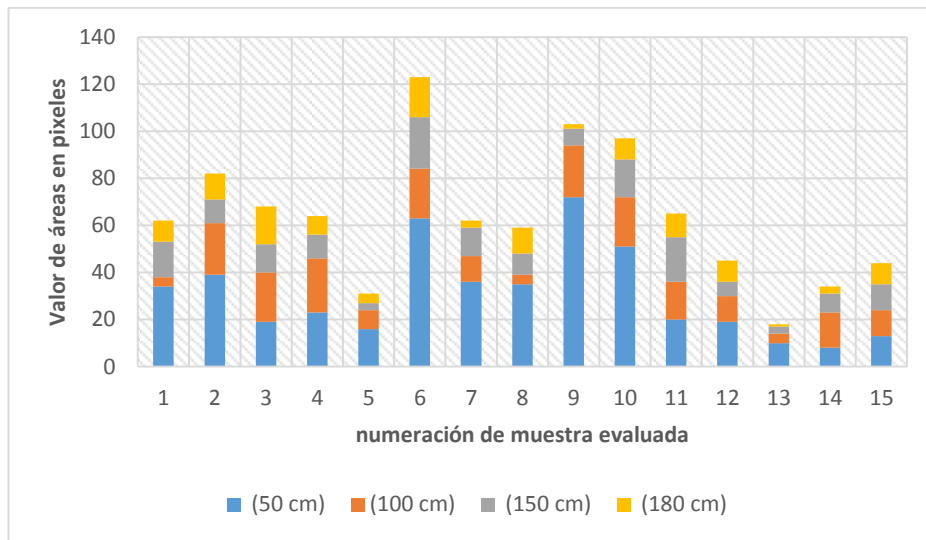


Fig. 31: Resultados de áreas sobre fondo oscuro
Elaboración: El autor

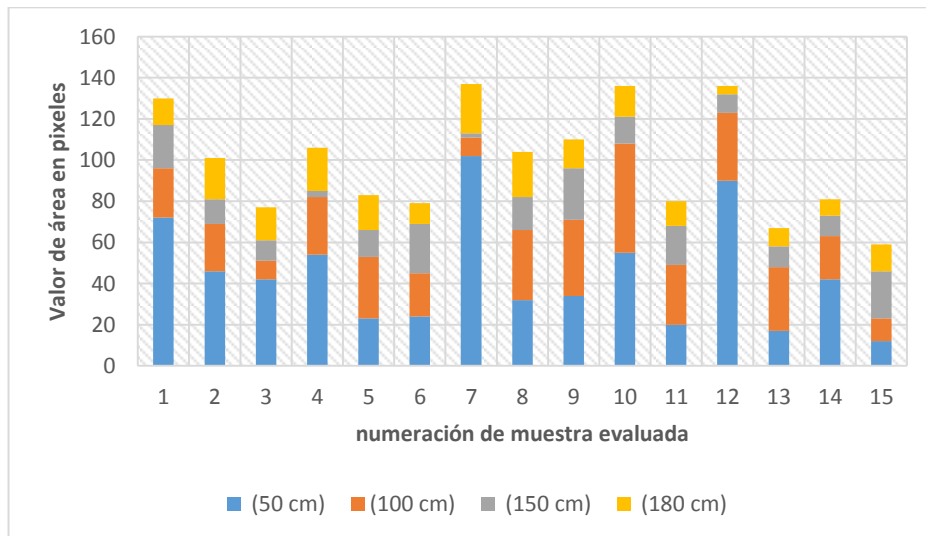


Fig. 32: Resultados de áreas sobre fondo claro
Elaboración: El autor

De la evaluación se comprueba que la relación de áreas es inversamente proporcional a la distancia, donde a mayor distancia se detectan objetos con área de menor cantidad de pixeles, respecto a la

distancia se realizaron las siguientes pruebas de detección. Tal muestra la tabla 8 a continuación.

Tabla 8: Resultados de variación de distancias

Distancia\ Muestra	1°	2°	3°	4°	5°	6°	7°	8°	9°	10°	11°	12°	13°	14°	15°
0.5m	34	39	19	23	16	63	36	35	72	51	20	19	10	8	13
m	4	22	21	23	8	21	11	4	22	21	16	11	4	15	11
1.5m	15	10	12	10	3	22	12	9	7	16	19	6	3	8	11
1.8 m	9	11	16	8	4	17	3	11	2	9	10	9	1	3	9

Elaboración: El autor

Como se puede apreciar en los resultados, a 200 cm el proceso de detección de luminancia ya no ofrecen valores adecuados para que se considere como objeto de interés, ya que dentro de ese rango de valores de área encuentran las intermitencias.

4.4. Pruebas de tiempo de ejecución

El sistema implementado funciona en tiempo real, y la cámara tiene una velocidad de respuesta de 30 fps. Por ello se procedió a evaluar la carga computacional que ejerce el algoritmo, mediante la medición de 50 muestras tanto cuando existe detección de puntero o no. De los cuales se obtienen los siguientes resultados, mostrados en la figura 33.

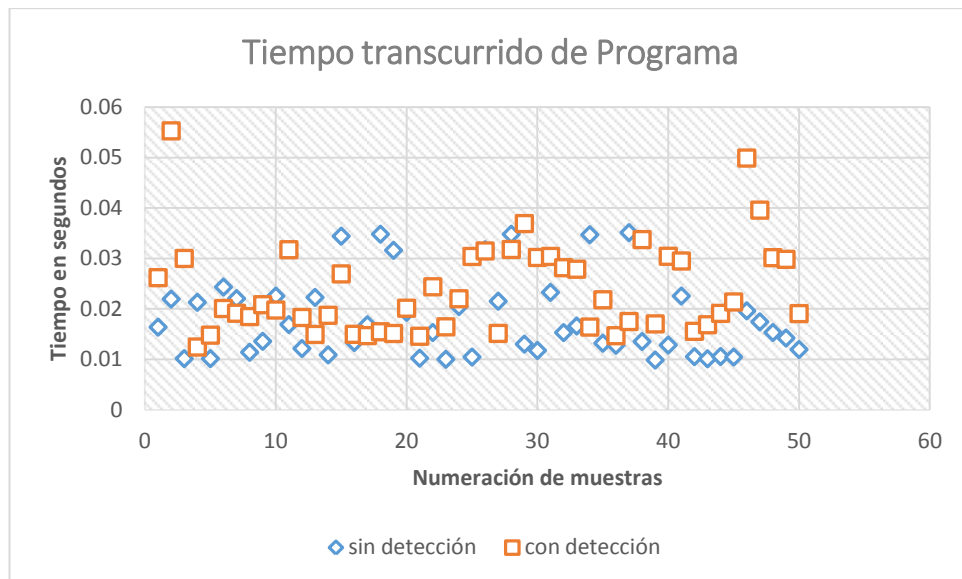


Fig. 33: Resultados de carga computacional
Elaboración: El autor

De los resultados, se obtuvo que la carga computacional promedio que ejerce el programa es de 0.018 segundos, aproximadamente, cuando no se tiene detección del puntero, mientras que cuando existe detección se tiene un promedio de 0.024 segundos. Ambos valores quedan por debajo del tiempo de respuesta de trabajo de 0.033 de la cámara web.

4.5. Pruebas de retardo de programa

Otro punto importante del programa es el proceso de interpolación variable, el cual produce un delay durante la ejecución de los procesos del sistema. Para su evaluación se obtuvieron los valores de 500 muestras, obteniendo la curva mostrada en la figura 34.

Los valores representados muestran el valor de tiempo máximo y mínimo que representa interpolar un punto durante la ejecución del sistema.

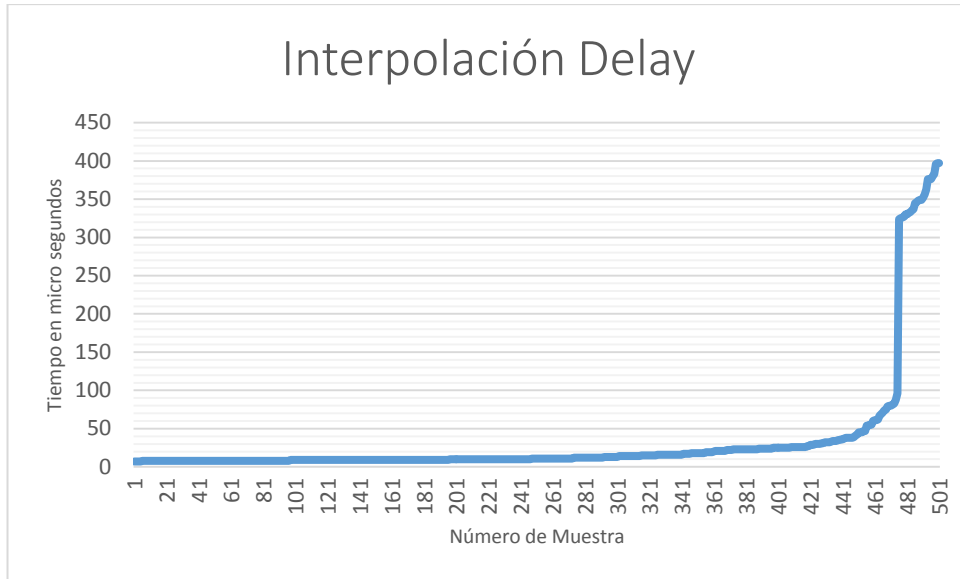


Fig. 34: Resultados de retardo por interpolación
Elaboración: El autor

4.6. Pruebas de funcionamiento de prototipo

Durante el desarrollo del programa se logró alcanzar los objetivos de cada proceso por separado. Posteriormente se unieron los distintos bloques para el funcionamiento como sistema.



(a) Ejemplo de uno de las diferentes formas como se puede implementar el sistema



(b) Muestra enfoque de la cámara y del proyector para lograr funcionamiento

Fig. 35: Pruebas de instalación de sistema.
Elaboración: El autor

Se realizaron pruebas de funcionamiento sobre diferentes entornos de luz, y proyectando las imágenes sobre superficies de distintos materiales y/o colores. La Figura 36, muestra algunos gráficos realizados por usuarios utilizando el sistema.



(a) Grafico realizado en sobre proyección en tamaño reducido en superficie de madera



(b) Imagen de escritura realizada sobre proyección en aula utilizando el Ecram como superficie

Fig. 36: Pruebas de uso de programa
Elaboración: El autor

4.7. Sustento desde el punto de vista comercial

Para poder definir si el diseño propuesto es solución de bajo costo, se requiere definir el costo del proyecto viéndose como si fuera un producto comercial, detallado en la Tabla 9.

Tabla 9: Costo de prototipo

Componente	Precio en dólares
Logitech Quickcam	105.00
Puntero LED	5.00
Soporte de cámara*	15.00
Total	125.00

Elaboración: El autor

Como se muestra, el valor del sistema propuesto está constituido principalmente por el costo de la webcam utilizada, cuya elección no tuvo como criterio el punto de vista comercial.

Teniendo en cuenta que la tesis presentada fue concebida únicamente con fines académicos, debe verse bajo un aspecto comercial comparativo para evaluar si es una solución de bajo costo.

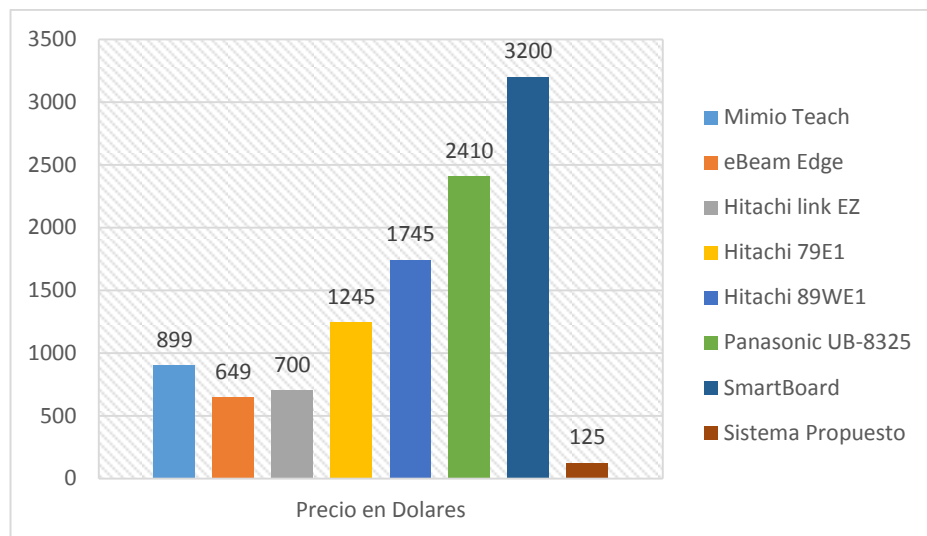


Fig. 37: Comparación costos productos comerciales
Elaboración: El autor

CAPÍTULO V

DISCUSIÓN Y APLICACIONES

En un primer momento se vio la opción de basar la detección del puntero por el color de la luz, pero por la amplia gama de colores que posee el fondo, se determinó no utilizar un color en particular, sino un umbral basado en el valor de Luminancia de las imágenes capturadas por la cámara, debido a que, una gran intensidad producida por cualquier color de luz que use el puntero LED se aproxima a la luz blanca, y es por ello que se podrá aislar del fondo.

Se realizaron pruebas utilizando el modelo de color HSL, pero los resultados no fueron los mejores y se optó por implementar el algoritmo mediante el uso del modelo RGB, obteniendo resultados positivos para las necesidades.

El uso del modelo RGB implica ahorrar carga computacional al sistema debido a que evita realizar un pre-proceso a las imágenes ya que la webcam realiza la captura de imágenes bajo este modelo de color.

El sistema no funciona con un puntero de luz infrarroja, ya que a pesar que la luz infrarroja puede ser detectada por las cámaras, los emisores infrarrojos LED no proporcionan una intensidad de luz suficiente para el proceso de detección. Aparte se tiene en cuenta que los sistemas infrarrojos

tienen problemas de interferencias debido a que la luz solar y algunos elementos lumínicos emiten luz infrarroja.

Las aplicaciones para el proyecto se extienden a todo proceso dentro de instituciones educativas así como la ejecución interactiva de presentaciones, muy comunes hoy en día en el ambiente laboral entre otros.

El tema de investigación abarcado en la presente tesis puede ser aplicado a otros proyectos sobre detección en fondos dinámicos, control de eventos del mouse o detección de luz brillante (generada por diversos emisores) por separado y en entornos totalmente distintos.

También se logró tener éxito en la detección de luminancia utilizando la cámara integrada de la laptop, de manera que se puede ver la posibilidad de integrar el sistema en una aplicación para otros dispositivos con cámaras integradas como, por ejemplo, en los teléfonos celulares.

CONCLUSIONES

1. Se implementó un sistema de detección de luminancia utilizando el modelo de color RGB, capaz de aislar el objeto de interés de un fondo dinámico, variable en el tiempo, que produce brillo y distintas tonalidades de color. En la investigación realizada respecto al estado del arte de sustracción de fondo no se encontraron proyectos que tengan fondos con las características mencionadas, por lo que la sustracción de fondo basado en luminancia podría ser un aporte importante en el tema.
2. El sistema es capaz de detectar la intensidad de luz de cualquier color, ya sea producida por un LED ultra brillante o un láser sobre el área de proyección siempre y cuando cumpla con los valores determinados para ser considerado como objeto de interés.
3. Se consiguió implementar el control de movimientos del cursor y eventos del mouse mediante el desarrollo de dos algoritmos en C que son llamados como funciones propias de Matlab® gracias al compilador 'MEX', que convierte dichos archivos en DLL.
4. Se comprobó, mediante la comparación de resultados, el óptimo funcionamiento del cálculo realizado para determinar el centroide de objeto de interés, ya que el cursor se posiciona en el lugar exacto (más de un 95% de las veces) donde se indicó con el puntero. La exactitud del posicionamiento es dependiente del proceso de calibración.

5. Se logró que el sistema se pueda ejecutarse bajo cualquier resolución de computador utilizado sin necesidad de hacer cambios, siempre y cuando se realice la calibración del sistema sobre dicha resolución de pantalla. Por lo mismo, el sistema puede ejecutarse sobre cualquier tamaño de proyección, siempre que se pueda enfocar completo con la cámara.
6. Trabajar con cámaras convencionales, en comparación de las cámaras infrarrojas, aumenta considerablemente la funcionalidad y disponibilidad de uso del sistema.
7. Trabajar en el espectro visible y basar el sistema a la detección de luminancia, permite que el programa sea capaz de trabajar con emisores de luz de cualquier color y de distinto tipo, ya que el sistema no solo detecta la luz producida por el puntero LED, sino que también es capaz de detectar luz producida por los punteros laser, comúnmente utilizados por profesionales que dictan clases o brindan presentaciones, apoyando nuestro fundamento que el sistema se acomoda al usuario.
8. El diseño del proyecto y tema de investigación fueron realizados únicamente fines académicos. Sin embargo, de considerar una salida al mercado y/o utilización en distintas instituciones se recomendaría el implementar, como en pizarras interactivas comerciales, un software asociado exclusivo para el sistema. El software asociado es lo que aumenta la capacidad de alcance y funcionalidad del sistema, permitiendo entre otras cosas, realizar manuscritos, guardar lo trabajado sobre la superficie, etc.

RECOMENDACIONES

1. El proceso de calibración automática puede mejorarse y/u optimizarse para obtener mejores resultados en ambientes con iluminación poco favorable para el proceso. Se recomienda la investigación de proyectos o métodos que utilicen un sistema de detección de formas más riguroso.
2. Para el correcto funcionamiento del sistema se recomienda no sea utilizado sobre superficies irregulares ya que producirá un error en la precisión. Tampoco utilizar superficies brillantes o que reflejen brillo intenso, ya que producirá interferencias sobre la detección de luminancia.
3. Se recomienda ubicar la cámara con un ángulo respecto a la zona de interés con el fin que el movimiento normal del usuario no obstruya la línea de vista entre la cámara y el puntero led. Se considera una buena práctica ubicar la cámara en el lado opuesto de la mano con la que el usuario "escribe". Es decir, si el usuario es diestro, ubicar la cámara en el lado izquierdo y viceversa.
4. El proyecto tiene como limitante la indisponibilidad de utilizar el teclado en pantalla de manera normal, debido a razones aún desconocidas. A pesar que es una herramienta poco común en la práctica, se recomienda la investigación de las características propias de la herramienta en mención para encontrar la base del fallo.

5. El costo para el desarrollo del proyecto se basa únicamente en el valor de la cámara que se desee utilizar. Teniendo en cuenta la Tabla de evaluación de costos de los productos comerciales presentada en el primer capítulo, se puede considerar alcanzado el objetivo de que el sistema diseñado es una solución de bajo costo.

6. Una manera de reducir el costo del producto sería necesario realizar el análisis sobre las propiedades de la cámara utilizada y llegar a estimar los valores mínimos necesarios para la adquisición de imágenes y así poder evaluar el producto con cámaras de menor costo. Esto es posible en vista que durante el desarrollo de la tesis se obtuvieron resultados confiables de detección utilizando la cámara integrada de la laptop, cuyas especificaciones técnicas son muy limitadas en comparación con la cámara Logitech utilizada durante el proyecto.

FUENTES DE INFORMACIÓN

Bibliográficas

1. Banerjee, A., Ghosh, A., Bharadwaj, K., & Saiki, H. (2014). *Mouse control using webcam based on color detection*. India: International journal of computer trends and technology (UCTT).
2. British Educational Communications and technology agency. (2006). *The benefits of an interactive whiteboard*.
3. Chang, J.-T., & Cheng, T. (2011). *USA Patente nº US 2011/0157101 A1*.
4. Chung Lee, J. (s.f.). *Low-cost multipoint interactive whiteboard using the wiimote*. Pittsburgh, USA: Carnegie Mellow University.
5. Gonzales, R. C., & Woods, R. E. (1996). *Tratamiento digital de imagenes*. Addison-Wesley Iberoamericana S.A.
6. Grupo de análisis de políticas educativas y formación (GAPEF). (2010). *Evaluación pizarra digital en Aragón*.
7. Korkalo, O., & Honkaa, P. (2010). *Construction and evaluation of multitouch screen using multiple cameras located on the side of th*

display. ACM International conference on interactive tabletops and surfacces.

8. Marqués Graells, P. (2008). *Impacto de las TIC en educación: Funciones y limitaciones*.
9. Mery, D. (2008). *Visión por computador*. Departamento de Ingeniería Informatica, Universidad Católica de Chile.
10. Park, H. (s.f.). *Method for controlling mouse movement using real-time camera*. Providence, USA: Department of computer science Brown University.
11. Yeung, Y.-S. (2012). *Mouse cursor control with head and eye movements: A low-cost approach*. Austria: University of applied science Technikun Wien.

Electrónicas

12. Library, M. M. (s.f.). *Eventos de mouse windows form*. Obtenido de [http://msdn.microsoft.com/es-es/library/ms171542\(v=vs.110\).aspx](http://msdn.microsoft.com/es-es/library/ms171542(v=vs.110).aspx)
13. Library, M. M. (s.f.). *Procedimiento para usar controles ActiveX en diapositivas*. Obtenido de <http://msdn.microsoft.com/es-es/library/office/ff746186.aspx>
14. *Writing C functions in MATLAB (MEX-flies)*. (s.f.). Obtenido de <http://cnx.org/content/m12348>

ANEXOS

ANEXO 1

Especificaciones del sistema

Características del Sistema

- El sistema de pizarra virtual interactiva está basado en Procesamiento Digital de Imágenes.
- Es indiferente al color de la luz producida por el puntero, ya sea LED o Laser.
- Se puede trabajar sobre cualquier resolución de computador y/o Proyector.
- Se puede trabajar sobre diferentes tamaños de zona de proyección, hasta sobre pantallas de PC, Laptops o televisores.
- Trabaja sobre cualquier tipo de superficie*, siempre y cuando sea plana.
- Es indiferente a las sombras producidas por el usuario.
- Requiere ser calibrado antes de comenzar a utilizarse y al variar de resolución durante su uso.

ESPECIFICACIONES TÉCNICAS	
Tecnología	Procesamiento Digital de Imágenes
Dimensiones	NA
Peso	NA
Puntos de contacto	1
Tiempo de respuesta	23 ms
Adquisición de imagen	30 fps
Resolución optima	Adaptable
Distancia Máxima	180 cm
Distancia Optima	100 – 150 cm
Angulo webcam Recomendado	± 45°
Accesorios	lapicero LED
Conexión	USB (p/ webcam)
Alimentación de puntero LED	3V
Fuente de Energía	5V (p/ webcam)
Consumo máximo	-
Temperatura	NA

Limitaciones del Sistema

- Objetos brillantes o luz de focos o fluorescentes pueden ocasionar distorsión, siempre y cuando estén dentro de la visión de la cámara (cercano a la zona de proyección)
- Una vez calibrado el sistema no se puede mover la posición de la cámara, de moverse, tendrá que reiniciar el proceso de calibración
- No es multipunto
- Compatible solo con Windows