

**DISEÑO Y CONSTRUCCIÓN DE PLATAFORMAS MULTITOQUE DE TIPO ÓPTICO,
UTILIZANDO TÉCNICAS DE VISIÓN ARTIFICIAL EN TIEMPO REAL, PARA FACILITAR LA
INTERACCIÓN DE LOS USUARIOS CON EL COMPUTADOR**



**CARLOS DIEGO FERRÍN BOLAÑOS
FRANCISCO JAVIER BUCHELLY IMBACHÍ**

**UNIVERSIDAD DEL CAUCA
FACULTAD DE CIENCIAS NATURALES, EXACTAS Y DE LA EDUCACIÓN
DEPARTAMENTO DE FÍSICA
INGENIERÍA FÍSICA
POPAYÁN
2010**

**DISEÑO Y CONSTRUCCIÓN DE PLATAFORMAS MULTITOQUE DE TIPO ÓPTICO,
UTILIZANDO TÉCNICAS DE VISIÓN ARTIFICIAL EN TIEMPO REAL, PARA FACILITAR LA
INTERACCIÓN DE LOS USUARIOS CON EL COMPUTADOR**

**TRABAJO DE GRADO EN MODALIDAD DE INVESTIGACIÓN PRESENTADO COMO
REQUISITO PARCIAL PARA OPTAR AL TÍTULO DE INGENIERO FÍSICO**

**CARLOS DIEGO FERRÍN BOLAÑOS
FRANCISCO JAVIER BUCHELLY IMBACHÍ**

**Director
Ing. LEONAIRO PENCUE FIERRO**

**UNIVERSIDAD DEL CAUCA
FACULTAD DE CIENCIAS NATURALES, EXACTAS Y DE LA EDUCACIÓN
DEPARTAMENTO DE FÍSICA
INGENIERÍA FÍSICA
POPAYÁN
2010**

Nota de aceptación

Mg. César A. Quinayás B.

Jurado

Mg. Jauri León Téllez.

Jurado

Ing. Leonairo Pencue Fierro.

Director

Fecha de Sustentación: Popayán, 09 de Junio de 2010

AGRADECIMIENTOS

A Jehová Dios, por darme la existencia y permitirme lograr todas mis metas académicas.

A mis padres, Segundo Euclides Ferrín Villota (Q.E.P.D) y Yolanda Bolaños de Ferrín, al igual que a mis hermanos, Jonathan (Q.E.P.D), Jesús (Q.E.P.D), Liliana, Janeth, Yesenia, Marina, Wiston y Héctor por darme todo su apoyo en esta importante etapa de mi vida académica y personal.

A los profesores: Jauri León Téllez, Leonairo Pencue Fierro, por darme orientación y apoyo en los momentos más difíciles de mi carrera y demostrarme su sincera amistad.

A los profesores: Leonairo Pencue Fierro, Luis Fernando Echeverri, Jauri León Téllez, Diego Bravo, Wilson Mafla, Jorge Rodríguez, Rubiel Vargas, Mario Patiño, Luz Elena Bolívar, Servio Tulio Pérez, María Elena Vivas y a todo el Departamento de Física por brindarme bases sólidas en mi formación.

A mis amigos y compañeros: Francisco Buchelly, Dagoberto Mayorca, Didian Valdez, Alex Ramírez, Sandra Juliana Vivas, German Vivas, Lucedy Idrobo, Diana Chantre, Jorge Posada, Hader Molina, Paola Navia, Lina Jaller, Karen Pérez, Yamileth Ortégón, Diego Correa, Carolina Quintana y todos aquellos quienes han compartido conmigo esta agradable experiencia.

Carlos Diego Ferrín Bolaños.

AGRADECIMIENTOS

A la Naturaleza (Dios) por permitirme estudiarle y por darme la existencia.

A mis padres, Luz Angélica Imbachí Narváez y Francisco Bucheli Chamorro, al igual que a mis hermanos, Diana Marcela y Jhon Edisson, por apoyarme en esta importante etapa de mi vida académica y personal.

A los profesores: Jauri León Téllez, Luis Fernando Echeverri, Jorge W. Coronel, Alberto Bravo Bucheli, Jairo Alfonso Vásquez, por darme orientación en momentos importantes de mi carrera.

A los profesores: Leonairo Pencue Fierro, Luis Fernando Echeverri, Jauri León Téllez, Luis Ernesto López, Jorge Rodríguez, Mario Andrés Córdoba y a todo el Departamento de Física por brindarme bases sólidas en mi formación.

A mis amigos y compañeros: Carlos Diego Ferrín, Sandra Juliana Vivas, German Vivas, Lucedy Idrobo, Dagoberto Mayorca, Diana Constáin, Jorge Adrada, Lina Jaller, Alexander Ramírez, Yamileth Ortegón y todos aquellos quienes han compartido conmigo esta agradable experiencia.

Francisco Javier Buchelly Imbachí.

TABLA DE CONTENIDO

	pág.
RESUMEN	14
INTRODUCCIÓN	15
CAPÍTULO I. INTERACCIÓN HUMANO-COMPUTADOR Y MULTITOQUE ÓPTICO	18
1.1. INTERACCIÓN HUMANO-COMPUTADOR	18
1.1.1. Propósitos de la interacción Humano-Computador	19
1.1.2. Criterios para diseñar un buen sistema de IHC	19
1.2. DISPOSITIVOS DE ENTRADA COMUNES	20
1.3. TECNOLOGÍAS MULTITOQUE	21
1.3.1. Estado del arte en tecnologías multitoque	21
1.3.2. Plataformas ópticas para tecnologías multitoque	24
CAPÍTULO II. CONCEPTOS SOBRE VISIÓN ARTIFICIAL	26
2.1. EL PROCESO DE FORMACIÓN DE LAS IMÁGENES	26
2.1.1. Modelo ideal de la cámara (modelo <i>pin-hole</i>)	26
2.2. DIGITALIZACIÓN DE LAS IMÁGENES	28
2.3. DISTORSIONES DEBIDAS A LAS LENTES Y CALIBRACIÓN DE CÁMARA	28
2.4. PROCESAMIENTO DIGITAL Y ANÁLISIS DE IMÁGENES	32
2.4.1. Imagen	32
2.4.2. Región de una imagen	33
2.4.3. Señal unidimensional	33
2.4.4. Perfil de línea	33
2.4.5. Integrales proyectivas	34

2.4.6. Filtrado de imágenes	35
2.4.7. Conversión a escala de grises	36
2.4.8. Binarización	37
2.4.9. Operaciones con imágenes binarias	38
CAPITULO III. CONSTRUCCIÓN DE PLATAFORMAS TÁCTILES MEDIANTE VISIÓN POR COMPUTADOR	39
3.1. REFLEXIÓN TOTAL INTERNA FRUSTRADA	39
3.1.1. Principio físico	39
3.1.2. Esquema experimental propuesto	40
3.1.3. Método de detección	41
3.2. ILUMINACIÓN DIFUSA	42
3.2.1. Principio físico	42
3.2.2. Esquema experimental propuesto	44
3.2.3. Método de detección	47
3.3. PROYECCIÓN DE SOMBRAS	49
3.3.1. Esquema experimental propuesto	49
3.3.2. Método de detección	51
3.4. TRIANGULACIÓN CON DOS CÁMARAS	56
3.4.1. Esquema experimental propuesto	56
3.4.2. Método de detección	57
CAPÍTULO IV. TÉCNICAS DE ANÁLISIS DE PUNTOS DE TOQUE Y GENERACIÓN DE ÓRDENES	62
4.1. SEGUIMIENTO DE OBJETOS EN DOS DIMENSIONES	62
4.1.1. Primeros vecinos y criterio de la mínima distancia	63
4.1.2. Criterio de la trayectoria más suave	64
4.2. IDENTIFICACIÓN DE GESTOS BASADOS EN POSICIÓN Y TRAYECTORIAS	66

4.2.1. Iconos gestuales con un toque	67
4.2.2. Gestos de traslación, rotación y redimensionamiento con dos toques	69
4.3. GENERACIÓN DE EVENTOS Y ORDENES AL COMPUTADOR	74
4.3.1. Funciones de control del sistema operativo	74
4.3.2. Métodos de comunicación entre aplicaciones	76
CAPITULO V. ANÁLISIS DE RESULTADOS	79
5.1. DESEMPEÑO Y LIMITACIONES DE LOS MÉTODOS DE DETECCIÓN	79
5.1.1. Montaje de FTIR	79
5.1.2. Montaje de iluminación difusa	81
5.1.3. Montaje de proyección de sombras	82
5.1.4. Montaje de triangulación con dos cámaras	85
5.1.5. Comparación entre los distintos métodos.	87
5.2. EFICIENCIA Y ROBUSTEZ DE LOS ALGORITMOS DE PROCESAMIENTO	88
5.2.1. Seguimiento	89
5.2.2. Identificación de gestos y generación de eventos	96
5.3. FUNCIONAMIENTO CONJUNTO DE HARDWARE Y SOFTWARE	96
5.3.1. Realimentación visual en la superficie táctil	
5.3.2. Ejemplos	99
CONCLUSIONES	102
TRABAJOS FUTUROS	104
BIBLIOGRAFÍA	105

LISTA DE TABLAS

	pág.
Tabla 3.1. Parámetros reales con los cuales se puso a prueba el sistema de triangulación	61
Tabla 5.1. Comparación entre las características de detección de toques de los cuatro esquemas elaborados en sus condiciones de funcionamiento	87
Tabla 5.1. Diversas trayectorias utilizadas para el aprendizaje de íconos gestuales con un dedo.	90
Tabla 5.2. Resultados obtenidos de la prueba del perceptrón utilizado con muestras distintas a las del entrenamiento.	93
Tabla 5.3. Algunos fallos del perceptrón en el reconocimiento de íconos gestuales.	94

LISTA DE FIGURAS

	pág.
Figura 1.1. Diagrama de flujo del funcionamiento de un sistema multitoque usando visión artificial.	25
Figura 2.1. Modelo <i>pin-hole</i> de la cámara.	26
Figura 2.2. Relación geométrica entre el ancho de la matriz de fotodetectores y el ángulo de apertura.	27
Figura 2.3. Formación de una imagen con aberraciones geométricas comunes.	29
Figura 2.4. Corrección de distorsiones como resultado de la calibración de una cámara cuyo ángulo de apertura es de 30 grados.	31
Figura 2.5. Ejemplo de un perfil de línea de una imagen a través de la línea horizontal señalada.	34
Figura 2.6. Ejemplo de una imagen en escala de grises y la proyección vertical de la región señalada.	35
Figura 2.7. Ejemplo de filtros pasa bajas y pasa altas aplicados a una misma imagen, respectivamente.	36
Figura 2.8. Ejemplo de umbralización de una imagen en escala de grises,	37

utilizando un umbral de 180 y otro de 200.

Figura 3.1. Componentes mínimos empleados en un sistema FTIR para tecnologías multitoque.	40
Figura 3.2. A la izquierda: fenómeno de FTIR en infrarrojo debido a dos dedos tocando una superficie de acrílico cerca de los bordes del mismo. A la derecha: detección de las coordenadas de toque en la imagen binaria.	41
Figura 3.3. Esquema básico de Iluminación Difusa por reflexión.	43
Figura 3.4. Esquema propuesto para detección de toques mediante Iluminación difusa por reflexión.	44
Figura 3.5. Vista de la superficie de contacto para iluminación difusa.	45
Figura 3.6. Comparación entre uso de luz infrarroja y visible para reflexión difusa.	46
Figura 3.7. A la izquierda está el gradiente de la imagen original. A la derecha, la imagen binaria y se identifica cada región de toque con sus coordenadas.	49
Figura 3.8. Funcionamiento simplificado del sistema de proyección de sombras visto de frente y de perfil.	50
Figura 3.9. Vista frontal y de perfil del sistema completo en funcionamiento.	50
Figura 3.10. Las zonas blancas corresponden a las lámparas vistas por la cámara,	52

en verde los umbrales y en violeta las proyecciones en un instante dado.

Figura 3.11. Modelo óptico geométrico del sistema de proyección de sombras.	54
Figura 3.12. Disposición de los componentes para el método de triangulación.	56
Figura 3.13. Obtención del perfil de línea. La línea amarilla representa la fila a escanear. Las graficas en la parte de abajo representan los perfiles de línea.	57
Figura 3.14. Cálculo de x_1 y x_2 en pixeles. La línea azul es el umbral y la curva violeta es la derivada del perfil de línea suavizado.	58
Figura 3.15. Modelo geométrico para determinar las coordenadas (x, y) en el tablero.	59
Figura 4.1. Seguimiento mediante el criterio de mínima distancia.	64
Figura 4.2. Arquitectura de una Red Neuronal Artificial.	67
Figura 4.3. Ejemplo de un perceptrón de tres capas.	68
Figura 4.4. Operación de traslación con dos toques.	70
Figura 4.5. Operación de rotación con dos toques.	71
Figura 4.6. Operación de redimensionamiento con dos toques.	73
Figura 5.1. Numerosos LED infrarrojos para iluminar la superficie de detección	80

para FTIR.

Figura 5.2. Casos de ambigüedad con dos toques para un sistema de proyecciones.	84
Figura 5.3. Sistema de iluminación y de proyección en el esquema de reflexión difusa.	97
Figura 5.4. Superficie difusora siendo proyectada.	98
Figura 5.5. Componentes del sistema de proyección en el esquema con dos cámaras.	98
Figura 5.6. Simulación de ondas de agua sobre la superficie	99
Figura 5.7. Esquema de triangulación con dos cámaras siendo utilizado en distintas aplicaciones de Windows.	100
Figura 5.8. Sombras proyectadas cuando se presenta un toque, dos toques o más de dos toques.	101
Figura 5.9. Interacción entre uno y dos usuarios con el esquema de proyección de sombras.	101

RESUMEN

Este documento muestra el diseño y la construcción de plataformas multitáctiles de principios tecnológicos ópticos, basados en visión artificial, que constan de módulos materiales o hardware y algoritmos de procesamiento e inteligencia (detección de toques, seguimiento, reconocimiento de gestos y generación de eventos), que constituyen distintos conjuntos de elementos conocidos como interfaces hombre-máquina. Esto con el fin de mejorar la forma como el usuario interactúa con el computador para hacerlo de una forma más directa, sencilla y natural por medio del contacto de los dedos sobre una pantalla, utilizando también realimentación visual. Se muestra además cómo es posible con estos sistemas permitir la intervención de múltiples usuarios en una misma plataforma.

Lo anterior se justifica con base al impacto positivo que se puede generar para la sociedad regional, lo cual comprende características como la facilidad en el aprendizaje de informática hacia los nuevos usuarios, la enseñanza en general, el entretenimiento, la publicidad; junto a una buena relación costo/beneficio que permite que este tipo de tecnologías de última generación se conviertan en productos económicamente accesibles para las empresas, instituciones y personas en general.

INTRODUCCIÓN

El desarrollo de nuevas interfaces de interacción humano-computador es un reto de gran importancia para la sociedad tecnológica actual, que muchos centros de investigación y empresas a nivel mundial asumen en su organización. El objetivo principal en este campo de la ingeniería es lograr que la interacción entre el hombre y la máquina sea más directa, intuitiva y natural. Los avances en este campo han permitido que el aprendizaje de los nuevos usuarios frente al manejo del computador sea cada día más rápido; esto se debe principalmente a que tales avances piensan más en la psicología del hombre y no sólo en la complicada lógica del computador.

En las últimas décadas se han hecho importantes avances en este campo, como la invención de las interfaces gráficas de usuario en 1973 y su utilización mediante dispositivos apuntadores, como el ratón [1]. Sin embargo, aunque actualmente son muy empleadas y acogidas por la sociedad en general, debido a que estas interfaces aun carecen de amigabilidad frente al usuario, se han buscado formas alternativas que permitan que dicha interacción se realice de forma más humana. Debido a que se considera el computador como una herramienta muy útil para llevar a cabo las actividades diarias, como el trabajo, el aprendizaje y la recreación, se necesita mejorar los mecanismos de interacción con él. Por tal motivo, existen investigaciones en todo el mundo [2, 3] dedicadas a crear nuevos dispositivos y algoritmos de computador capaces de interpretar órdenes de diferentes tipos, tales como reconocimiento de voz, generación de comandos a partir de gestos corporales y sistemas basados en tecnologías táctiles, entre otros. Cada uno de ellos ofrece ventajas y desventajas respecto a los demás. Sin embargo, para la mayor parte de los usuarios sin limitaciones físicas, la tendencia natural para la manipulación de objetos, herramientas y la realización de actividades cotidianas

implica utilizar las manos y los dedos, lo que conduce a dar gran importancia al perfeccionamiento de las tecnologías multitoque.

Existen en el mercado diversos sistemas manipulados mediante pantallas táctiles, que en su mayoría constan de sensores eléctricos [4]. Muchos de estos sistemas poseen inconvenientes, principalmente porque su uso requiere de un solo toque, convirtiéndose tan solo en una mejora a los dispositivos de señalamiento. Entonces se debe satisfacer la necesidad tecnológica del multitoque empleando diferentes principios físicos de detección, por ejemplo, los métodos ópticos y visión por computador porque permiten obtener las coordenadas de los puntos de toque con precisión, confiabilidad y ofrecen rapidez en los algoritmos de procesamiento. Esto permite integrar los dispositivos de entrada y despliegue en un único periférico capaz de mostrar la información de manera gráfica y al mismo tiempo sensor los comandos que provienen de gestos hechos por el usuario mediante múltiples toques.

La construcción de dispositivos de captura y detección de coordenadas de toque ofrece algunos beneficios a la sociedad descritos a continuación. Desde el punto de vista de la necesidad del acercamiento amistoso de la tecnología a las personas, se tiene que el empleo de múltiples toques, junto con algoritmos adecuados de seguimiento de objetos e identificación de gestos permite que el usuario manipule el computador para dar órdenes de forma muy cómoda. A esto se le añade la posibilidad de utilización de una misma plataforma por múltiples usuarios, lo que implica un hecho importante en el fomento del trabajo en equipo, en comparación con los esquemas tradicionales que pretenden el uso de un computador por persona.

En el ámbito tecnológico, se tiene que estos mecanismos presentan muchas características novedosas puesto que aún son un campo de investigaciones recientes en muchos países. La detección de los puntos de toque se puede hacer con gran confiabilidad

porque existen diversos tipos de esquemas ópticos que se pueden emplear en distintas condiciones de iluminación. La precisión y la escalabilidad son dos cualidades que también están presentes porque se trabaja con imágenes de buena resolución y se pueden crear pantallas o superficies de distintos tamaños dependiendo de la aplicación requerida. Adicionalmente, se emplea procesamiento de video en tiempo real, algoritmos rápidos y robustos, adaptables de forma automática e inteligente a cambios de iluminación, rutinas de seguimiento de objetos, interpretación de gestos, entre otros. Se sabe que algunos productos comerciales ya soportan características multitoque, por ejemplo, computadores con sistemas operativos como Ubuntu 9.10 o Windows 7 [2, 3]. Lo que se ofrece aquí, es la posibilidad de trabajar tanto con estos sistemas como también con sus antecesores, los cuales no contaban con dichas cualidades.

Finalmente, en el aspecto económico regional, se sabe que el sur occidente colombiano no cuenta con una tecnificación empresarial en comparación con las principales ciudades del país, luego se esta brindando una facilidad para que las personas puedan adquirir sistemas ópticos multitoque debido a que estos no requieren materiales de alto costo para su fabricación, lo que puede traducirse en ahorro de recursos, en contraste con la importación de sistemas similares. La introducción de este tipo de tecnologías en la región puede ser una fuente de desarrollo ya que las empresas locales alcanzarían, entre otras metas, una mayor captación de clientela mediante publicidad, entretenimiento o educación, utilizando aplicaciones gráficas interactivas.

CAPITULO I. INTERACCION HUMANO-COMPUTADOR Y MULTITOQUE OPTICO

1.1. INTERACCION HUMANO-COMPUTADOR (IHC)

Es el diseño de sistemas computacionales que apoyan a las personas para que realicen sus actividades de manera eficiente y segura. Sin embargo, es necesario aclarar qué papel desempeña el computador y qué papel desempeña el usuario para que las tareas se lleven a cabo adecuadamente. Diferentes tipos de aplicaciones se requieren para diferentes propósitos y se necesita cuidado para dividir las tareas entre hombres y máquinas, asegurando que las operaciones numéricas o repetitivas se asignen a las máquinas mientras que las actividades creativas (diseño de dibujos o gráficos, redacción de textos, estudio de la disposición de componentes en una interfaz gráfica, etc.) o subjetivas (selección de colores, objetos o componentes según el gusto y la personalidad) sean asignadas a las personas [5]. Es por eso, que este tema no solo es un campo de las ciencias computacionales, sino que involucra varias disciplinas como: Psicología (qué tipo de elementos gráficos son agradables al usuario, permiten la operación normal de los sistemas computacionales incluso durante largos periodos de trabajo y cuáles son adecuados o no de acuerdo al contexto o ambiente en el que se trabaja), Ergonomía (diseño de elementos hardware que no alteren la postura o fisiología de las personas), Lingüística (estudio de las formas más eficientes de transmitir información del computador al usuario y recibir órdenes provenientes de él), Inteligencia Artificial (interpretación automática de los comandos recibidos), Antropología (diseño de sistemas teniendo en cuenta las posibles repercusiones o alteraciones culturales) y Diseño (elaboración estética de los elementos hardware y software).

1.1.1. Propósitos de la interacción humano computador

El rol de IHC en el de diseño de sistemas es mejorar la calidad de la interacción entre humanos y computadores. Para ello se requiere de la aplicación sistemática de conocimiento sobre las metas humanas, capacidades y limitaciones junto con el conocimiento sobre las capacidades y limitaciones de la tecnología. Este conocimiento debe relacionarse al entendimiento de los aspectos sociales, organizacionales y físicos del entorno de trabajo del usuario.

El reto para los que desean diseñar computadoras o sistemas que la gente pueda usar consiste en saber cómo hacer la transición de lo que se puede hacer (funcionalidad) a cómo debe hacerse para cubrir los requerimientos del usuario (usabilidad) en el ambiente de trabajo. En el nivel físico esto significa seleccionar los dispositivos adecuados de entrada o salida y determinar el mejor esquema de interacción (formas, lenguaje natural, interfaces gráficas de usuario, etc.) [5]. Todo esto con el fin de hacer la interacción entre el hombre y el computador lo mas natural posible.

1.1.2. Criterios para diseñar un buen sistema de IHC

Un buen diseño de la IHC permite al usuario comprender, utilizar y recordar la información más rápido y con mayor facilidad. Los criterios básicos para diseñar una buena interfaz [6] son: usabilidad, funcionalidad y comunicación visual y estética. En lo referente a la comunicación visual y estética, se debe manejar un ambiente amigable e interactivo, cuidando la ubicación correcta de los elementos y el uso adecuado de textos, gráficos, color y sonidos. Por lo que respecta a la usabilidad, hay que hacer un diseño adecuado al tipo de usuario así como manejar retroalimentación y un ambiente configurable.

Finalmente para la funcionalidad, es importante el control de navegación y el manejo adecuado de elementos de ejercitación y diagnóstico.

1.2. DISPOSITIVOS DE ENTRADA COMUNES

Entre los dispositivos que aprovechan el tacto está el teclado. Aunque existen varias configuraciones o variantes, su manejo requiere cierto nivel de entrenamiento y experiencia. La ventaja es que se pueden dar muchas órdenes o comandos. Su contraparte son las pantallas táctiles que permiten un aprendizaje inmediato, requieren menos espacio y su interacción con el usuario es directa. Se sabe que las pantallas táctiles tradicionales son imprecisas, cuentan con una tasa alta de errores e inducen rápido cansancio y obstrucción de datos con los dedos. Sistemas basados en tecnologías táctiles son, por ejemplo, aquellos que hacen reconocimiento de escritura manual, pero manipulan pocos datos, vocabulario limitado y generan errores frecuentes.

También existen apuntadores como el ratón, guantes con sensores, *joystick*, *touchpads*, dispositivos inalámbricos, etc. El uso común de estos dispositivos es el control de un puntero para interfaces gráficas de usuario y generación de eventos simples de acuerdo al número de controles que se presenten en pantalla. También sirven para controlar el movimiento de objetos virtuales en dos y tres dimensiones. Existen variantes del ratón para utilizar con el pie, pero de poca precisión.

Hay actualmente algunos sistemas de entrada de voz cuyas aplicaciones principales son, por ejemplo, manos libres o asistencia a discapacitados. Algunos de estos sistemas requieren el movimiento de la persona para poderse utilizar [7].

La elección de uno o varios periféricos va a depender de la aplicación final, el usuario y el contexto en el que se va a utilizar, ya que para ciertas actividades un dispositivo puede ser más adecuado que otro por su facilidad de uso y funcionalidad. No obstante, este trabajo muestra como la utilización de plataformas multitáctiles, con métodos ópticos de detección, en tiempo real, y con despliegue gráfico incorporado en la misma zona de toque para obtener la retroalimentación necesaria, puede convertirse en el reemplazo de varios dispositivos mencionados anteriormente más el monitor habitual.

1.3. TECNOLOGIAS MULTITOQUE

Multitoque es un conjunto de técnicas interactivas que permiten a los usuarios de computador controlar aplicaciones gráficas con varios dedos. Los dispositivos multitoque constan de una plataforma táctil (pantalla de computador, mesa o pared), como también de software que reconoce múltiples toques simultáneos, de forma opuesta a las pantallas táctiles estándar usuales que reconocen un único punto de toque [8]. El fin es diseñar Interfaces Naturales de Usuario que impliquen la elaboración de plataformas multitáctiles basadas en gestos o movimientos usuales hechos por las personas con las manos.

1.3.1. Estado del arte en tecnologías multitoque

A continuación se menciona una breve reseña histórica sobre el nacimiento de las tecnologías multitoque, para posteriormente mostrar cuáles son las tendencias actuales en este campo de las ciencias y la ingeniería. Esto tiene como objetivo mostrar que la investigación a nivel regional en estas áreas tiene mucha importancia puesto que el conocimiento de los nuevos avances a nivel mundial permite la adaptación de éstos a la solución de problemas locales concretos, desde el punto de vista de la viabilidad

económica por la elaboración de sistemas similares con recursos accesibles en el medio, como también las altas potencialidades en el mejoramiento del aprendizaje, la forma de interacción de las personas con el computador o el reconocimiento del mismo como una herramienta de trabajo cada vez más útil.

Las tecnologías multitáctiles surgieron en 1982 cuando un grupo de la Universidad de Toronto desarrolló la primera interfaz humana multitoque. Esta interfaz usaba un panel de vidrio escarchado con una cámara ubicada detrás de él. Cuando uno o varios dedos presionaban el vidrio, la cámara detectaría una o más manchas oscuras sobre fondo blanco, permitiendo registrar este patrón como entrada. Desde entonces, se han desarrollado con el paso del tiempo diversas tecnologías paralelamente a las de sensado óptico basado en cámaras, pero con preferencia hacia las de tipo electrónico tales como las matrices capacitivas [9]. Esto se debe a que la arquitectura de los computadores aún no era capaz de procesar la información en forma de imágenes en tiempo real con resolución aceptable.

Cuando la capacidad de procesamiento mejoró notablemente durante la década de los años 90, varias compañías empezaron a desarrollar diferentes prototipos, permitiendo así que en los primeros años del siglo XXI se den grandes avances en estos campos. El primer dispositivo comercial que utilizaba tecnología multitáctil óptica basada en visión por computador fue el *Lemur Input Device*, un controlador multimedia profesional de una compañía francesa, que apareció en el mercado en 2005. En julio de 2007, Apple registró la palabra *multi-touch* junto con su *iPhone*. El nuevo *iPod touch* también incluye esta tecnología entre sus características.

Uno de los pioneros en el ámbito del multitoque es el científico investigador Jefferson Han de la Universidad de New York, quien introdujo el principio de la reflexión total interna frustrada para la fabricación de este tipo de plataformas. El presentó su trabajo de

sensado multitáctil en 2006 en Monterrey, California. Jeff Han fundó la compañía *Perceptive Pixel* para continuar desarrollando sus tecnologías y se conoce que sus grandes aportes van desde productos comerciales hasta dispositivos militares [10].

Desde el surgimiento masivo de diferentes esquemas de plataformas multitoque a principios del siglo, importantes compañías a nivel mundial han hecho grandes aportes tecnológicos con sus novedosos productos. Tal es el caso de Microsoft, iniciando con teléfonos de pantalla táctil en 2006 hasta llegar a la actualidad con uno de sus sistemas más populares que detecta múltiples toques y registra gestos, denominado *Microsoft Surface*. El sistema *Microsoft Surface* permite además la interacción entre múltiples usuarios, el reconocimiento de objetos reales y su funcionamiento se basa en iluminación directa utilizando luz LED infrarroja, una superficie difusora (iluminación difusa por reflexión) y un sistema de visión por computador, además de contener diversos puertos para comunicaciones o salidas de audio [11].

Actualmente, muchas otras empresas están creando sus propios prototipos multitoque tanto ópticos como electrónicos. Ejemplos de compañías e instituciones que diseñan productos con gran impacto tecnológico son: La Universidad de Tromsø en Noruega junto con la Universidad de Princeton, desarrolladoras de un sistema multicámaras tipo 1394 conectadas en red para detectar eventos en 2D y 3D no solo en el plano de despliegue sino también en diferentes planos cercanos a él [12]; como también la empresa PQ Labs desarrolladora de la *iTable* con pantalla de 42 pulgadas, soporte de hasta 32 dedos simultáneos [13].

Finalmente, cabe resaltar que ninguno de estos sistemas tendría éxito comercial si no poseen su propio software encargado de manejarlo, el cual realiza operaciones de identificación y seguimiento constante de cada punto de toque u objetos, con el fin de correr distintas aplicaciones en ellos. Sin embargo, la diversificación de las plataformas

comerciales multitoque introduce la necesidad en las empresas, como también en los investigadores particulares de utilizar estándares que permitan ejecutar programas independientemente del hardware diseñado. Es por esta razón que muchas organizaciones como el grupo NUI (*Natural User Interfaces*) [8], los creadores del proyecto *ReactiVision* y diferentes desarrolladores de sistemas operativos como Linux actualmente trabajan en la implementación de algoritmos multiplataforma con el fin de hacer posible la compatibilidad requerida.

1.3.2. Plataformas ópticas para tecnologías multitoque

Los sistemas ópticos o de sensado de luz constituyen un gran porcentaje de los dispositivos multitoque conocidos actualmente. Cada técnica consiste básicamente de un sensor óptico (típicamente cámaras), fuentes de iluminación, y realimentación visual en forma de proyección sobre la plataforma o utilización de pantallas planas LCD, plasma o LED [8].

Cuando el elemento sensor es un sistema formador de imagen, como es el caso de las cámaras, es necesario adecuar tanto al hardware como al software un sistema de visión artificial o visión por computador que incluye tanto la ubicación más apropiada para los componentes ópticos y la cámara, como también el diseño de algoritmos eficientes, rápidos y robustos de procesamiento digital de imágenes o señales en general. La figura 1.1 muestra el funcionamiento típico de cualquier sistema multitáctil que utiliza visión por computador.

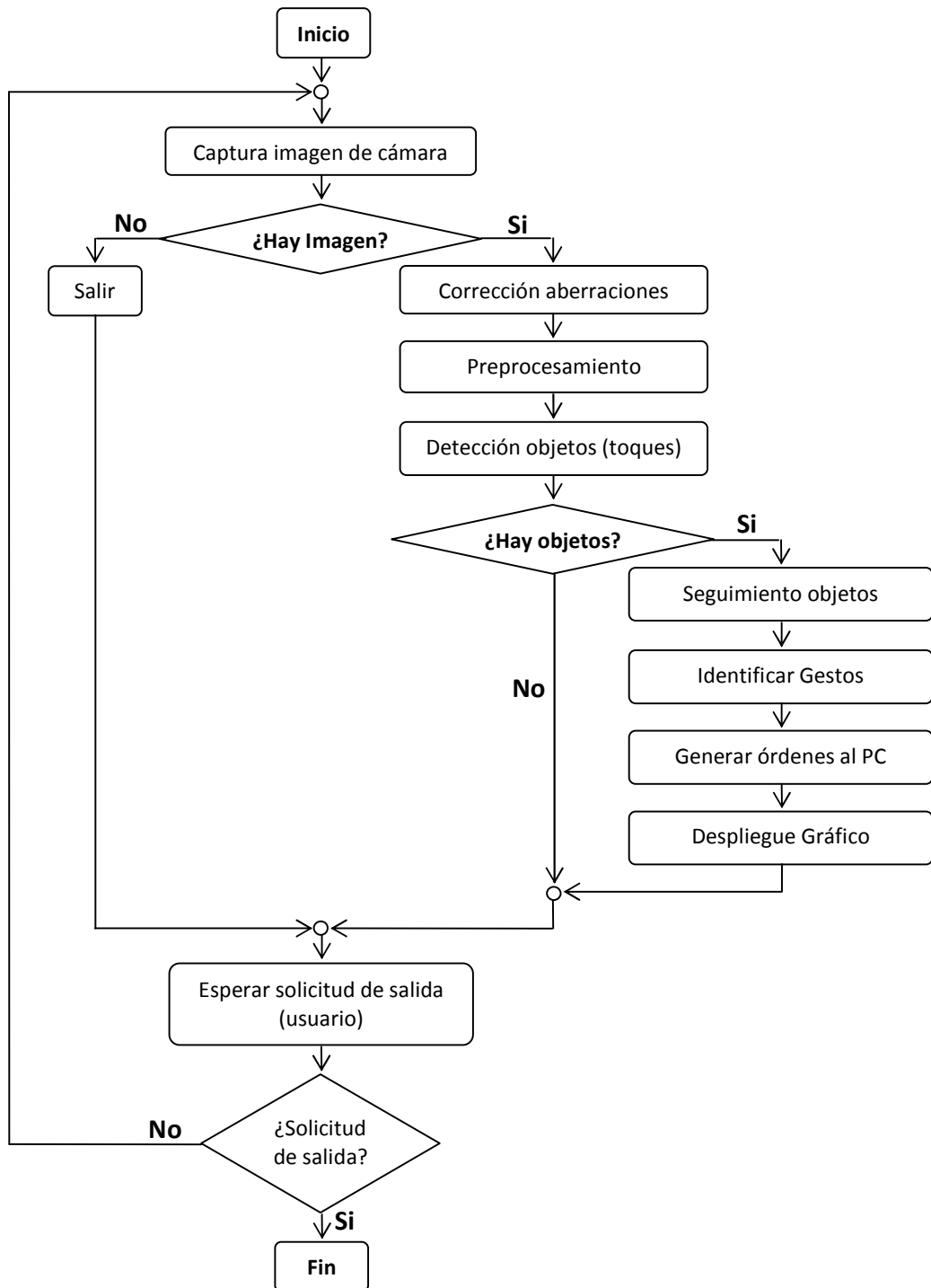


Figura 1.1. Diagrama de flujo del funcionamiento de un sistema multitoque usando visión artificial.

CAPITULO II. CONCEPTOS SOBRE VISION ARTIFICIAL

2.1. EL PROCESO DE FORMACION DE LAS IMÁGENES

La formación de imágenes es un proceso mediante el cual una escena 3D del mundo real es proyectada en un plano 2D, denominado imagen [14]. Una de las formas más comunes para adquirir imágenes es el uso de cámaras. A continuación se muestra el modelo ideal de la cámara, el cual permite obtener imágenes sin distorsiones.

2.1.1. Modelo ideal de la cámara (modelo *pin-hole*)

Este modelo permite representar el sistema óptico de una cámara como se muestra en la figura 2.1.

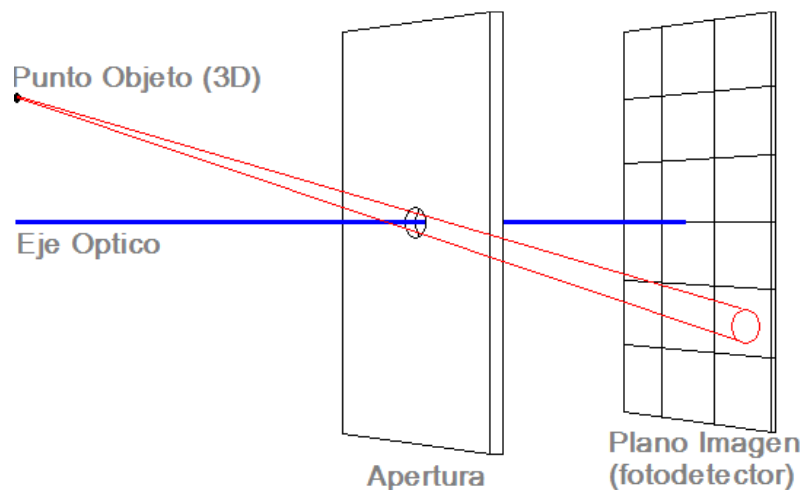


Figura 2.1. Modelo *pin-hole* de la cámara.

De acuerdo a la figura anterior, dependiendo del tamaño de la apertura, el cono de luz que se proyecta desde el punto origen en el objeto forma una mancha en el plano imagen que puede ser de mayor o menor tamaño que un fotodetector. Cuando la apertura es muy grande, la manchas imagen son mas grandes que un fotodetector, se superponen y como consecuencia la imagen se ve difuminada o desenfocada. Por esta razón se establece que la apertura es suficientemente pequeña para considerarse puntual y que a partir de cierta distancia de la cámara los objetos se vean enfocados, dentro de un intervalo de profundidades denominado profundidad de campo.

Se puede determinar la resolución angular de la cámara en la dirección horizontal, es decir, conociendo el ángulo de apertura total se puede encontrar el ángulo equivalente de cada uno de los elementos fotodetectores de la matriz (o pixel en la imagen digital) de acuerdo a las relaciones geométricas que se obtienen de la figura 2.2. Sea Θ la apertura angular, w el ancho de la matriz (número de columnas), entonces el ángulo discreto mínimo (equivalente a $\Delta x=1$) es Θ/w .

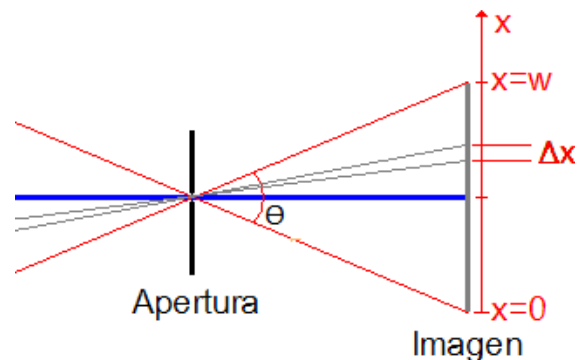


Figura 2.2. Relación geométrica entre el ancho de la matriz de fotodetectores y el ángulo de apertura.

De forma similar se puede obtener la resolución angular en la dirección vertical.

2.2. DIGITALIZACIÓN DE LAS IMÁGENES

La superposición de distintas combinaciones de rojo, verde y azul es suficiente para permitir a las personas observar los demás colores. Por tal razón, el arreglo de fotodetectores de una cámara es una matriz de sensores optoelectrónicos sensibles a estos tres tipos de radiación, capturando cada uno de estos valores por separado.

Cuando se codifica una imagen, se transfieren los valores existentes en el arreglo a un sistema digital encargado de transformar las señales analógicas producidas por las intensidades luminosas en algún formato numérico (generalmente de 8 bits). Con los valores discretos de iluminación codificados en niveles de gris digitales, se puede representar la imagen computacionalmente como una matriz bidimensional con cierto número de filas y columnas, además de tres planos de profundidad correspondientes a las componentes roja, verde y azul. A este modelo de representación de los colores se le denomina espacio RGB por sus siglas en inglés (*Red, Green, Blue*). Se denomina pixel a cada celda de la imagen a quien le corresponde un vector de valores RGB y se identifica por un índice para la fila y un índice para la columna a la que pertenece [15].

2.3. DISTORSIONES DEBIDAS A LAS LENTES Y CALIBRACION DE CAMARA

El modelo *pin-hole* dista del verdadero funcionamiento del sistema óptico de la cámara, debido a que posee al menos una lente que introduce distorsiones a las imágenes. Las distorsiones más comunes son de dos tipos: radial y tangencial. Las radiales se deben a que la fabricación de una lente esférica es más fácil que elaborar una lente parabólica cuya superficie es matemáticamente “ideal”. Por tal razón, si el objetivo de una cámara posee una apertura muy grande (radio de curvatura muy pequeño de la lente) las distorsiones radiales son mayores. Las distorsiones tangenciales surgen principalmente del

proceso de ensamble mecánico de los componentes ópticos. En la figura 2.3 se puede observar un ejemplo común de formación de la imagen con distorsiones de un objeto rectangular mediante una lente.

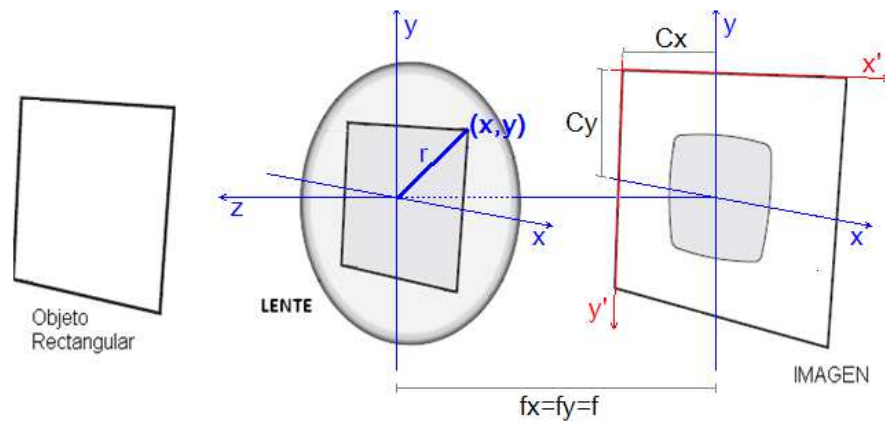


Figura 2.3. Formación de una imagen con distorsiones o aberraciones geométricas comunes.

Para eliminar las distorsiones que se presentan se necesita conocer los coeficientes de distorsión radiales, tangenciales y la matriz de parámetros intrínsecos de la cámara, que contiene la distancia focal en x (f_x), la distancia focal en y (f_y) y el desplazamiento del punto central (c_x, c_y) respecto al eje óptico para tomar como origen el pixel (0,0) en el sistema de coordenadas de la imagen (x', y').

Las correcciones a un punto (x, y) ubicado a una distancia r desde el centro de la imagen, cuando se presentan distorsiones radiales con coeficientes k_1, k_2 y k_3 , son:

$$x_{correg} = x(1 + k_1r^2 + k_2r^4 + k_3r^6) \quad (2.1)$$

$$y_{correg} = y(1 + k_1r^2 + k_2r^4 + k_3r^6) \quad (2.2)$$

Las correcciones cuando se presentan distorsiones tangenciales con coeficientes p_1 y p_2 son:

$$x_{correg} = x + (2p_1y + p_2(r^2 + 2x^2)) \quad (2.3)$$

$$y_{correg} = y + (p_1(r^2 + 2y^2) + 2p_2x) \quad (2.4)$$

Esta es la matriz de transferencia de la cámara (parámetros intrínsecos) que permite proyectar un punto en 3D sobre un plano imagen (x, y) :

$$M = \begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix} \quad (2.5)$$

La calibración de cámara es el proceso mediante el cual se determinan sus parámetros intrínsecos y coeficientes de distorsión. Para esto, se toman varios puntos de la escena con coordenadas conocidas (x, y, z) , se calcula las posiciones donde deberían aparecer (x, y) y se miden las posiciones donde se ubican realmente en la imagen. Se plantean las ecuaciones necesarias para el número de incógnitas requerido, se solucionan y se almacenan los resultados en un archivo para poderse reutilizar [16]. Actualmente existen diversos algoritmos computacionales que realizan estos cálculos de forma automática.

La figura 2.4 muestra una fotografía tomada con una de las cámaras empleadas en el trabajo, antes de corregir distorsiones y después de hacerlo. Se señalan con líneas rectas de color verde los límites de una estructura recta y se observa cómo en la imagen original las distorsiones no permiten hacer esta correspondencia, mientras que en la imagen corregida las rectas trazadas se ajustan correctamente a las reales.

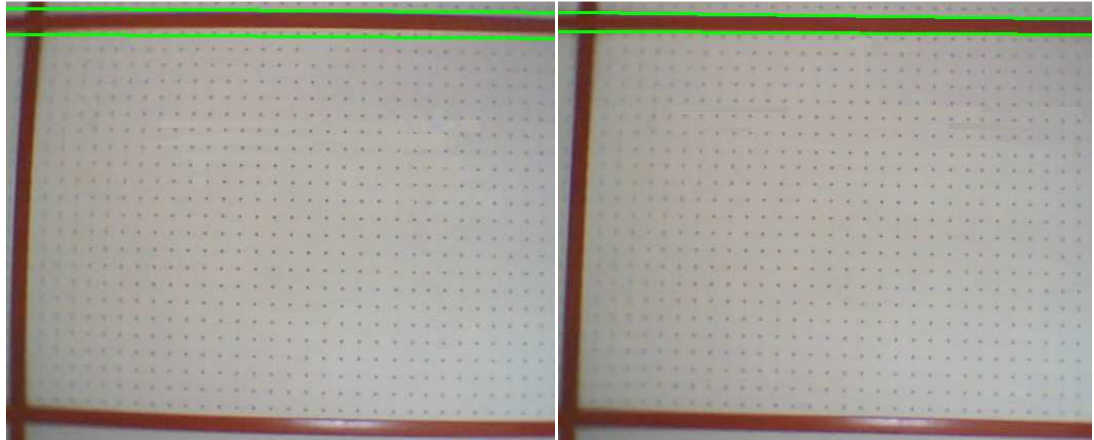


Figura 2.4. Corrección de distorsiones como resultado de la calibración de una cámara cuyo ángulo de apertura es de 30 grados. A la izquierda, imagen original. A la derecha, imagen corregida.

En el capítulo III, cuando se muestran imágenes de entrada para los montajes experimentales, es un hecho que previamente se calibró la cámara, se guardaron los coeficientes de distorsión y se hacen las correcciones antes del pre-procesamiento.

2.4. PROCESAMIENTO DIGITAL Y ANÁLISIS DE IMÁGENES

Se denomina procesamiento de imágenes a todas aquellas operaciones se tiene como entrada una imagen y como salida otra imagen, tales como: operaciones de punto, transformaciones geométricas, filtrado, conversión a escala de grises, binarización, entre otras; mientras que el análisis de imágenes es la extracción de descriptores o valores numéricos a partir de una imagen, por ejemplo el numero de objetos o la posición de cada uno.

Con el fin de dar una explicación formal de las operaciones que se utilizan en el capítulo III, a continuación se darán algunas definiciones matemáticas relacionadas con el tratamiento de las imágenes como funciones discretas [17]. Junto con la definición dada, se muestran ejemplos de las operaciones a realizar con imágenes tomadas de los montajes experimentales que se mencionan en este trabajo.

2.4.1. Imagen

Una imagen I de n canales es una función discreta: $[0, \dots, w-1] \times [0, \dots, h-1] \rightarrow \mathbb{R}^n$, de dos variables, donde w es su número de columnas y h es su número de filas. Usando la notación de funciones, se obtiene el valor de un pixel (x, y) mediante $I(x, y)$.

Se denota al conjunto de todas las imágenes de n canales con I^n . Normalmente se trabaja con imágenes en escala de grises, es decir de I^1 , o con el espacio RGB, esto es I^3 . Se utiliza la notación $I(x, y).r$, $I(x, y).g$, $I(x, y).b$, para referirse a las componentes del pixel.

2.4.2. Región de una imagen

Una región R , en una imagen I , es el conjunto $\{I(x, y): x_{\min} \leq x \leq x_{\max}, y_{\min} \leq y \leq y_{\max}\}$, donde x_{\min} y x_{\max} pertenecen a $[0, w-1]$, y_{\min} y y_{\max} pertenecen a $[0, h-1]$, $x_{\min} \leq x_{\max}$ y $y_{\min} \leq y_{\max}$.

2.4.3. Señal unidimensional

Es una función discreta en una variable, es decir, $s: [s_{\min}, \dots, s_{\max}] \rightarrow \mathbb{R}$. Cabe mencionar que las mismas operaciones que se aplican a una imagen de h filas y w columnas se

pueden aplicar a una señal unidimensional, ya que esta es un caso particular de las imágenes, donde $h=1$ o $w=1$;

2.4.4. Perfil de línea

Es una señal unidimensional obtenida al muestrear una imagen de I^1 a lo largo de una línea. Generalmente se utilizan perfiles horizontales o verticales. Los perfiles de línea horizontal y vertical se definen de la siguiente forma, donde y_i y x_i son un valor constante de fila y columna, respectivamente, dentro de la imagen:

$$Lh(x) = I(x, y_i) \quad (2.6)$$

$$Lv(y) = I(x_i, y) \quad (2.7)$$

La figura 2.5 muestra un ejemplo de perfil de línea horizontal de una imagen, donde la gráfica inferior representa los niveles de gris respecto a la coordenada x .

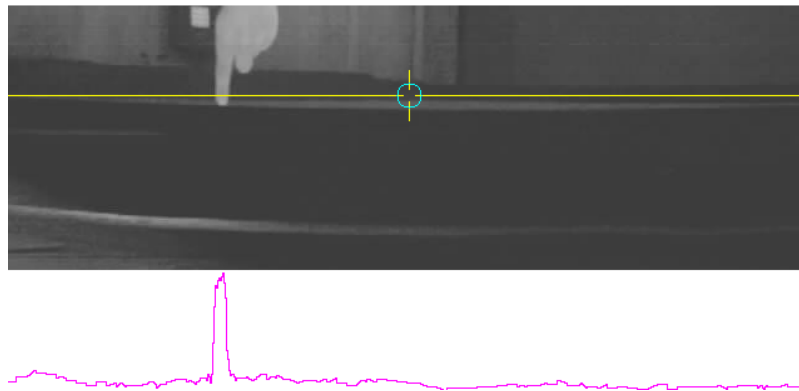


Figura 2.5. Ejemplo de un perfil de línea de una imagen a través de la línea horizontal señalada.

2.4.5. Integrales proyectivas

La proyección vertical de una región R definida por $[x_{\min}, x_{\max}]$ y $[y_{\min}, y_{\max}]$, en una imagen de I^1 , es una señal unidimensional de variable $y' = y - y_{\min}$ ($0 \leq y' \leq y_{\max} - y_{\min}$), definida por:

$$Pv_R(y') = \frac{1}{x_{\max} - x_{\min} + 1} \sum_{x=x_{\min}}^{x_{\max}} I(x, y' + y_{\min}) \quad (2.8)$$

Similarmente, para la misma región R, se define la integral proyectiva horizontal con variable $x' = x - x_{\min}$ ($0 \leq x' \leq x_{\max} - x_{\min}$) como:

$$Ph_R(x') = \frac{1}{y_{\max} - y_{\min} + 1} \sum_{y=y_{\min}}^{y_{\max}} I(x' + x_{\min}, y) \quad (2.9)$$

A continuación se muestra un ejemplo de proyección vertical en la región seleccionada mediante el rectángulo de color cian. La imagen corresponde al patrón de iluminación de dos lámparas en el montaje de proyección de sombras descrito en el capítulo III.

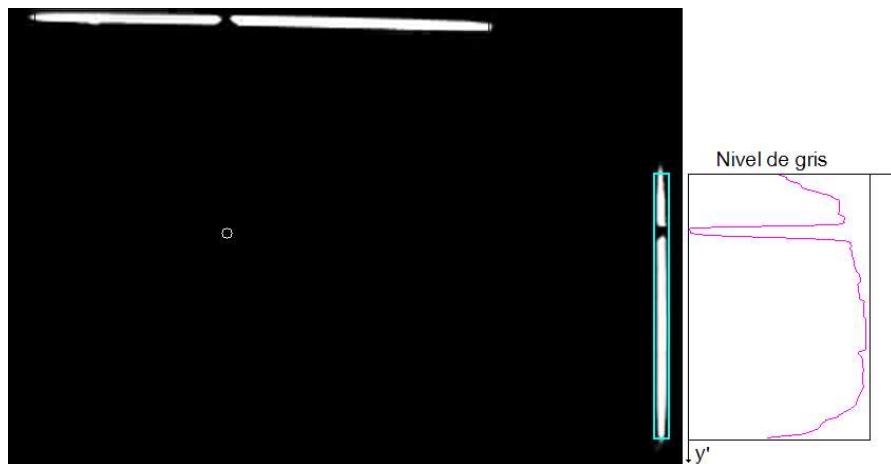


Figura 2.6. Ejemplo de una imagen en escala de grises y la proyección vertical de la región señalada.

2.4.6. Filtrado de imágenes

Son muchas las fuentes de ruido y efectos no deseados de iluminación que se pueden presentar durante la captura de las imágenes desde una cámara. Entre estas se encuentran: ruido electrónico en la cámara, presencia de polvo, impurezas o pequeñas partículas en la escena, cambios no controlados en la iluminación del ambiente (luz solar u oscilaciones debidas a lámparas, sombras por objetos en movimiento, etc.), no uniformidad en los colores y tonalidades de los fondos requeridos, entre otras. Es por esta razón que antes de hacer alguna operación sobre la imagen, es necesario tratar de eliminar o atenuar hasta un nivel adecuado estos disturbios. Dependiendo la aplicación, para atenuar eliminar lo que no necesitamos de la imagen se puede aplicar el enfoque de filtrado espacial, que ofrece las dos posibilidades de filtros más utilizadas en procesamiento digital de imágenes: pasa bajas o pasa altas.

En resumen, un filtro pasa bajas tiene el efecto de reducir los cambios fuertes en la diferencia de intensidad entre pixeles cercanos dando como resultado una imagen difuminada, mientras que un filtro pasa altas resalta los cambios grandes de intensidad entre ellos debidos a los detalles finos de los objetos, como son los bordes. Ejemplos de filtros comunes son: el filtro de media (promedio entre un pixel y sus vecinos) es pasa bajas y los filtros derivativos (derivadas de Prewitt, Sobel, Gradiente y Laplaciano) son pasa altas [15]. La figura 2.7 muestra el efecto de los dos tipos de filtros mencionados: el filtro pasa bajas de media y filtro derivativo de Sobel aplicados a la misma imagen. La imagen original corresponde a la sombra de una mano, iluminada con luz visible vista a través de una superficie difusora. Este montaje se describe más detalladamente en el capítulo III.



Figura 2.7. Ejemplo de filtros pasa bajas y pasa altas aplicados a una misma imagen, respectivamente.

2.4.7. Conversión a escala de grises

En general, se puede definir esta operación como una transformación cuya imagen de entrada $I(x, y)$ es de I^3 y su salida $G(x, y)$ de I^1 . Entre las transformaciones más utilizadas están las siguientes:

- Operación de promedio ponderado: $G(x,y)=A I(x, y).r+ B I(x, y).g+ C I(x, y).b$, donde $A+B+C=1$;
- Operación de promedio normal: la operación anterior con $A=B=C=1/3$;
- Operación de máximo: $G(x, y)=\text{Max} (I(x, y).r, I(x, y).g, I(x, y).b)$.

Por su bajo consumo de recursos computacionales, la operación de máximo es la que se emplea con más frecuencia en numerosas funciones de bibliotecas de procesamiento de imágenes conocidas, por ejemplo la biblioteca OpenCV de Intel [18].

2.4.8. Binarización

Este concepto hace referencia a cualquier operación que tome como entrada una imagen y genere como salida otra imagen que solo tenga dos posibles colores. Generalmente la nueva imagen I pertenece a I^1 y los elementos $I(x, y)$ son los valores de saturación, por ejemplo 0 y 255 para una imagen de números de 8 bits sin signo.

Como ejemplo se muestra la imagen en escala de grises (formato entero de 8 bits) de la figura 2.8 (parte izquierda), correspondiente a la vista de una mano iluminada con luz infrarroja vista a través de una superficie difusora. La condición de binarización es poner en 0 los valores de un pixel que superen un umbral fijo y en 255 los que no. Con esta operación se tiene que los pixeles cuyo valor sea inferior al umbral se establecen en blanco, mientras que el resto se establece en negro. La imagen central de la figura 2.8 corresponde a una binarización con umbral de 180 y la imagen de la derecha con umbral de 200. Esta es una aproximación a un método de detección de toques, pero como se verá en el próximo capítulo, la umbralización simple no es confiable debido a la distribución no uniforme de la iluminación.



Figura 2.8. Ejemplo de umbralización de una imagen en escala de grises, utilizando un umbral de 180 y otro de 200.

2.4.9. Operaciones con imágenes binarias

Existen varias operaciones que se pueden realizar con imágenes binarias, por ejemplo: filtrado morfológico, conteo de objetos, detección y análisis de contornos etc. En el siguiente capítulo, se mostrará que en algunos sistemas se necesita obtener la silueta de las regiones de toque para calcular su centro de masa (c_x, c_y) , el cual está definido por las siguientes expresiones [16]:

$$c_x = \frac{1}{n} \sum_{i=1}^n x_i \quad (2.10)$$

$$c_y = \frac{1}{n} \sum_{i=1}^n y_i \quad (2.11)$$

Donde n es el número de píxeles del objeto en la imagen binaria y (x_i, y_i) representa cada punto del objeto.

CAPITULO III. CONSTRUCCIÓN DE PLATAFORMAS TÁCTILES MEDIANTE VISION POR COMPUTADOR

En la actualidad existen muchas topologías de detección de múltiples toques mediante técnicas que utilizan luz [8]. En este trabajo se mencionan algunos métodos ópticos puestos a prueba como son: reflexión total interna frustrada, iluminación difusa, proyecciones de sombras y triangulación con dos cámaras.

Para todos los métodos de detección se elabora su respectivo sistema de visión artificial que consta de una o dos cámaras y una etapa de detección de coordenadas de toque mediante procesamiento digital de imágenes. Para tal fin, se emplea un compilador libre de C++ junto con la biblioteca de código abierto de visión por computador OpenCV (*Open Source Computer Vision Library*) de Intel, con el fin de realizar todas las operaciones requeridas en tiempo real.

3.1. REFLEXIÓN TOTAL INTERNA FRUSTRADA

3.1.1. Principio físico

La reflexión total interna es una condición presente en ciertos sistemas cuando la luz incide en la interfaz entre dos materiales, proviniendo del material que tiene mayor índice de refracción (n_i) e incidiendo con un ángulo mayor que un ángulo crítico. El ángulo crítico o mínimo al cual ocurre este fenómeno se puede calcular usando la ley de Snell [19], dando como resultado que el seno del ángulo crítico es la razón entre los índices de

refracción de los dos medios (ecuación 3.1). Cuando esto sucede, no se da refracción en el material y el haz de luz es totalmente reflejado.

$$\sin(\theta_c) = \frac{n_t}{n_i} \quad (3.1)$$

Cuando un objeto dispersor se pone en contacto con la interfaz, en aquel punto se rompe la condición de reflexión total interna y se dice que los rayos son frustrados y la reflexión ya no es total en ese punto. A este fenómeno se le denomina Reflexión Total Interna Frustrada (FTIR, *Frustrated Total Internal Reflection*).

3.1.2. Esquema experimental propuesto

El esquema que se utilizó es similar al propuesto por Jeff Han, quien introdujo el fenómeno FTIR a las tecnologías multitoque [8] (Ver figura 3.1).

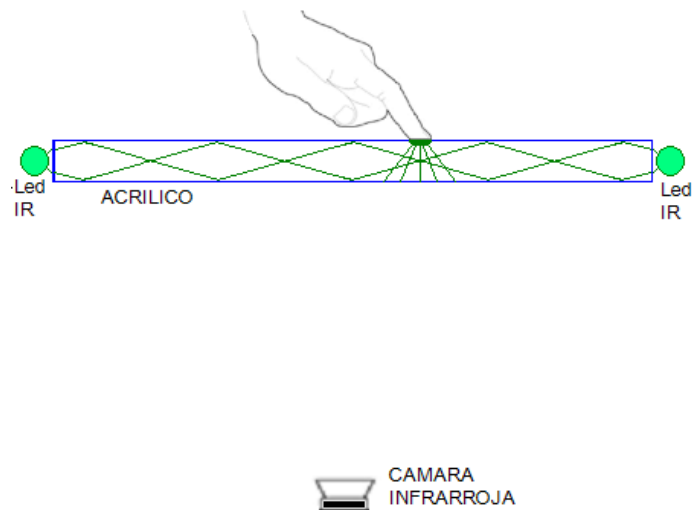


Figura 3.1. Componentes mínimos empleados en un sistema FTIR para tecnologías multitoque.

Este método utiliza una fuente de luz infrarroja (arreglo de IR LED de luz infrarroja cercana comprendida entre 700 nm y 1000 nm de longitud de onda) incidiendo por los extremos de una lámina de acrílico transparente, cuyo grosor es superior a 6cm. La luz frustrada es esparcida por la piel y capturada por una cámara infrarroja. La imagen obtenida se codifica como un patrón de manchas brillantes provenientes de las zonas de contacto.

3.1.3. Método de detección

La figura 3.2 muestra las imágenes obtenidas mediante el método de detección propuesto. Se señaló en color verde la región de interés a analizar dentro de la imagen, correspondiente a la superficie de acrílico. Se puede observar que utilizando luz infrarroja, el contraste de las manchas brillantes en las regiones de contacto respecto al fondo es alto, siempre y cuando no incidan directamente las fuentes de luz parásita como la luz solar o diferentes tipos de lámparas. Por lo tanto el procesamiento previo para la imagen no requiere muchas operaciones de mejoramiento y filtrado.

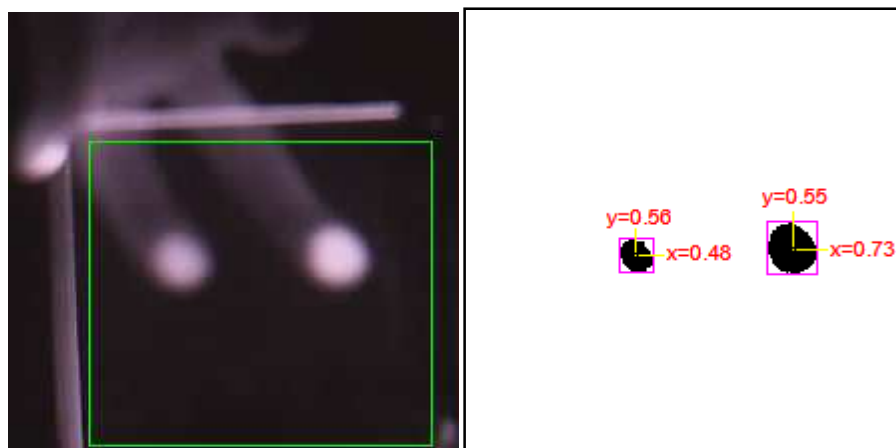


Figura 3.2. A la izquierda: fenómeno de FTIR en infrarrojo debido a dos dedos tocando una superficie de acrílico cerca de los bordes del mismo. A la derecha: detección de las coordenadas de toque en la imagen binaria.

Para la detección de los puntos de toque se tomó en cuenta únicamente la región de interés R seleccionada y se utilizaron dos transformaciones, T_1 (transformación a escala de grises) y T_2 (umbralización), definidas de la siguiente manera:

T_1 es la transformación de una imagen I (3 canales, formato de 8 bits sin signo) a escala de grises, es decir, $T_1: I^3 \rightarrow I^1$, donde la imagen resultante G (1 canal, 8 bits sin signo) tiene el mismo tamaño de la imagen I y se obtiene mediante la operación de máximo: $G(x, y) = T_1\{I(x, y)\} = \text{Max}(I(x, y).r, I(x, y).g, I(x, y).b)$ para todos los puntos (x, y) en R y $G(x, y) = 0$ para (x, y) fuera de R.

T_2 es la transformación de G a una imagen binaria, es decir, $T_2: I^1 \rightarrow I^1$, donde la imagen resultante B (1 canal, 8 bits sin signo) tiene el mismo tamaño de la imagen G y se obtiene mediante la operación de umbralización uniforme: sea u el umbral (u está entre 0 y 255), entonces $B(x, y) = T_2\{G(x, y)\} = 255$ si $G(x, y) \geq u$ o 0 en caso contrario.

Finalmente, se calculan los centros de masa de cada objeto binario para obtener sus coordenadas en forma numérica utilizando las ecuaciones 2.10 y 2.11.

3.2. ILUMINACION DIFUSA

3.2.1. Principio físico

Cuando la luz incide sobre una superficie no especular, es decir, posee rugosidades o relieve irregular, sus rayos no cumplen con las leyes de Snell para la reflexión o refracción, sino que salen desviados del objeto en todas las direcciones [19]. Este efecto se puede aprovechar para la elaboración de plataformas de detección de toques si la superficie es

plana, lo suficientemente delgada para dejar pasar luz y sus rugosidades son pequeñas. Como resultado, la imagen de un objeto visto a través de la superficie difusora, pierde los detalles finos y revela únicamente la silueta o forma del mismo.

Se pueden emplear dos esquemas de iluminación difusa para detección de toques: por reflexión (figura 3.3) o por transmisión [8]. Los dos utilizan la superficie de contacto en posición horizontal. Se utiliza un material difusor para crear un patrón de sombras sobre un fondo brillante o un patrón de manchas más brillantes que el fondo. Para que exista realimentación visual, la iluminación y la detección se realiza empleando luz infrarroja, ya que la proyección sobre la superficie se hace con luz visible.

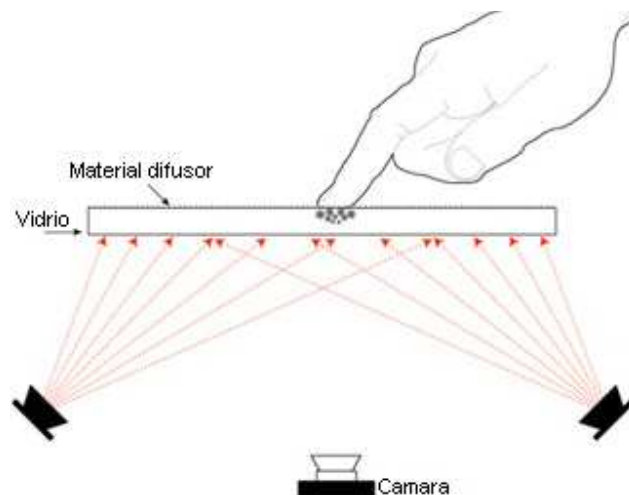


Figura 3.3. Esquema básico de Iluminación Difusa por reflexión. Imagen tomada de [8].

Si el esquema es por reflexión, se ilumina el área de detección desde la parte inferior tratando de crear una distribución de intensidad de luz uniforme. Cuando ocurren los toques, la cámara detecta un patrón de manchas más brillantes que el fondo en las regiones donde se está haciendo contacto. Si el esquema es por transmisión, la luz

proveniente de lámparas o del ambiente brilla en la pantalla desde arriba de la superficie de contacto. Cuando un objeto toca la superficie, se crea una sombra en esa región.

3.2.2. Esquema experimental propuesto

Se propuso trabajar con el esquema de reflexión mostrado en la figura 3.3 porque es el que ofrece mayor facilidad para la detección de toques debido a que presenta la menor cantidad de restricciones en la iluminación, es decir, no importan las condiciones de luminosidad del ambiente siempre y cuando no llegue a la superficie difusora incidencia directa de radiación solar, de lámparas incandescentes o de cualquier fuente de luz infrarroja externa al sistema. La restricción anterior se tiene en cuenta con el fin de evitar sombras indeseadas en las imágenes adquiridas. El montaje básico utilizado se muestra en la figura 3.4.

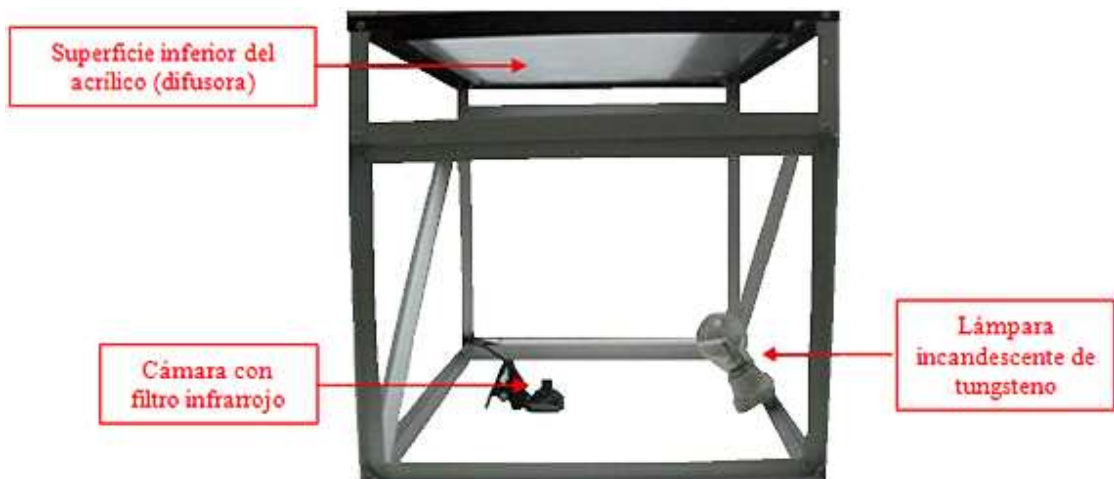


Figura 3.4. Esquema propuesto para detección de toques mediante Iluminación difusa por reflexión.

Para la elaboración del sistema hardware se utilizó una estructura en aluminio para soporte de los distintos elementos, una lámina de acrílico con una cara pulida, la cual desempeña la función de superficie táctil, una cámara de video con filtro pasa bandas en infrarrojo cercano, y una lámpara incandescente como fuente de luz infrarroja.

La figura 3.5 muestra la superficie táctil del sistema. En la superficie de detección no se cuenta con una distribución uniforme de iluminación de fondo producida por la fuente infrarroja. Es difícil tener tal condición deseada de uniformidad; sin embargo, debido a que la lámpara no es la única fuente presente de luz infrarroja es de esperar que las condiciones luminosas variables y cambiantes del ambiente impidan que los puntos de contacto se puedan detectar mediante umbralización simple. Posteriormente se explicará el módulo software diseñado para hacer dicha detección.

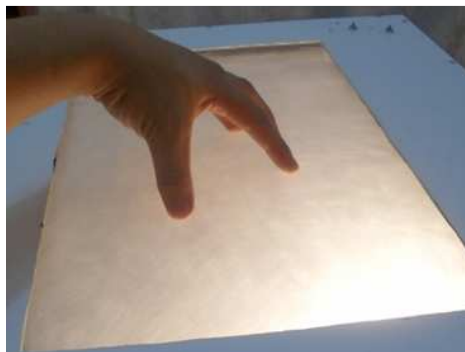


Figura 3.5. Vista de la superficie de contacto. Se puede observar que la distribución de iluminación no es uniforme.

Se hizo una comparación entre el uso del sistema con luz infrarroja y luz visible. Esto se resume en la figura 3.6.

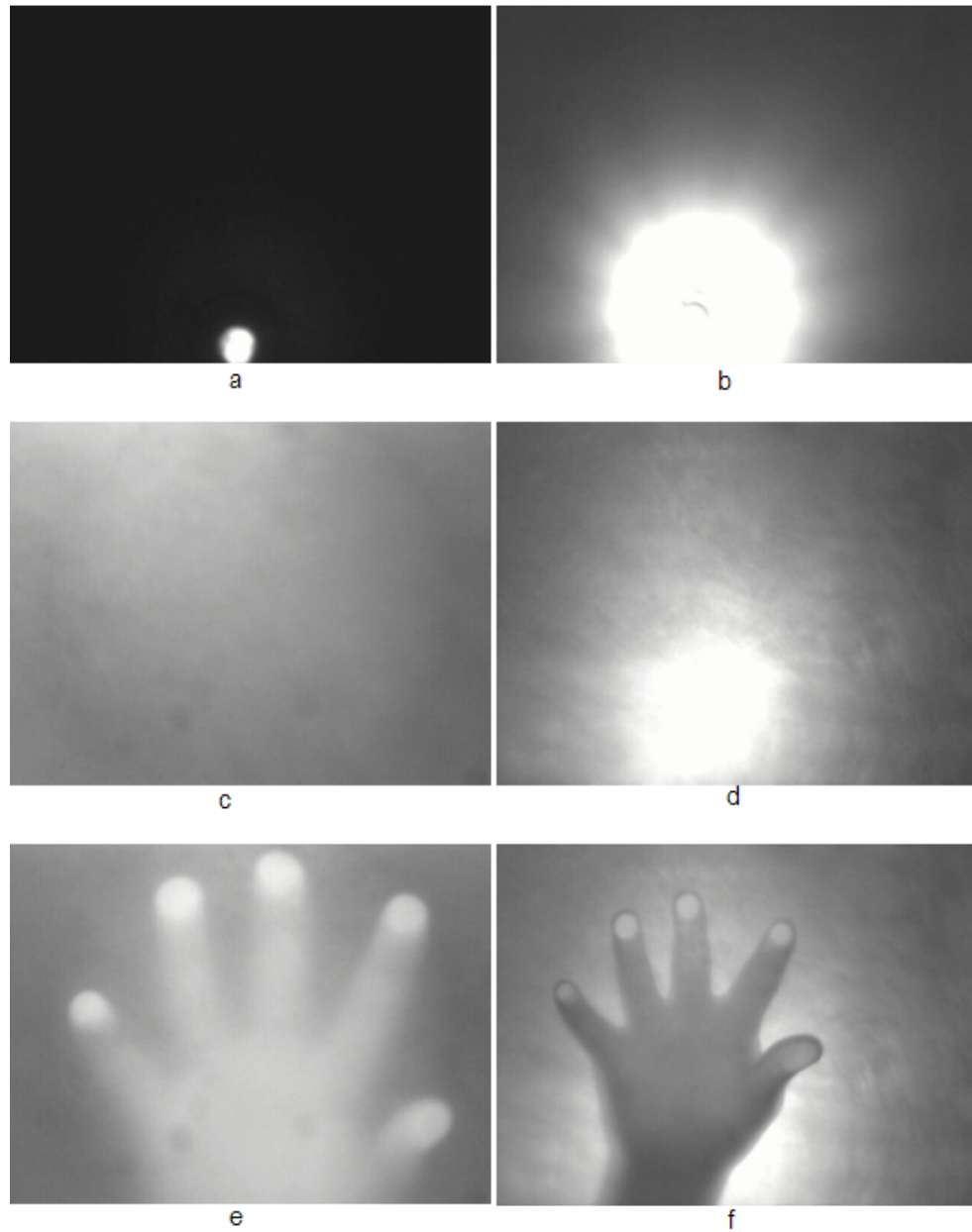


Figura 3.6. Comparación entre el uso de luz infrarroja (columna izquierda) y luz visible (columna derecha) para reflexión difusa: a) y b) vista directa del bombillo de luz blanca, c) y d) vista a través del difusor y con la lámpara incandescente encendida, e) y f) tocando la pantalla.

Primero se tomaron fotografías en escala de grises de un bombillo de luz blanca para determinar de forma cualitativa la cantidad de radiación infrarroja y visible emitida. Luego se observó la cantidad de radiación del mismo bombillo, en los dos espectros, que atraviesa por la lámina de acrílico, así como el efecto de dispersión (filtro pasa bajas) que ofrece la cara difusora. Finalmente, se encendió la lámpara incandescente para determinar visualmente el cambio de contraste y el efecto total de la iluminación parásita producida por el bombillo de luz blanca. El resultado fue muy favorable para el espectro infrarrojo, debido a que prácticamente es nula la cantidad de sombras producidas por la mano antes de tocar la superficie, mientras que con luz visible este efecto no deseado es muy notorio.

3.2.3. Método de detección

Como se puede observar en la figura 3.6 (parte e), las zonas de toque son más brillantes que su entorno. Sin embargo, para poderse detectar no basta simplemente con aplicar un umbral simple a toda la imagen, ya que la iluminación no es uniforme en toda la superficie, como se mencionó anteriormente.

Como una alternativa eficiente para la detección de las coordenadas de toque que utiliza un filtrado, en vista de la presencia de cambios muy grandes de niveles de gris en las fronteras de las manchas de contacto. Se tiene que a diferencia de estas zonas, las regiones de no contacto presentan bordes poco definidos por el efecto de la superficie difusora.

El filtro pasa altas propuesto requiere calcular la magnitud del gradiente de la imagen, vista como una función bidimensional discreta. Para una función continua $I(x, y)$, el gradiente se puede expresar de la siguiente forma:

$$\nabla I(x, y) = \left(\frac{\partial I(x, y)}{\partial x}, \frac{\partial I(x, y)}{\partial y} \right) \quad (3.2)$$

Para aproximar las derivadas requeridas de la imagen de entrada se utilizaron operadores diferenciales:

$$\frac{\partial I(x, y)}{\partial x} \approx \begin{pmatrix} 0 & 0 & 0 \\ -1 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix} \quad (3.3)$$

$$\frac{\partial I(x, y)}{\partial y} \approx \begin{pmatrix} 0 & -1 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \quad (3.4)$$

Con cada operador diferencial aplicado a la imagen en niveles de gris se obtiene una imagen correspondiente a la derivada direccional indicada, de la siguiente forma: Sea $G(x, y)$ la imagen original transformada a escala de grises (formato de 8 bits sin signo, w columnas y h filas), con un filtrado previo pasa bajas para atenuar el ruido. La derivada en x es una nueva imagen en punto flotante definida por la transformación: $D_1(x, y) = G(x+1, y) - G(x-1, y)$ para (x, y) en el interior de la imagen y $D_1(x, y) = 0$ en los bordes de la imagen. La derivada en y , D_2 , es una imagen del mismo tamaño y formato que D_1 y se define de forma similar: $D_2(x, y) = G(x, y+1) - G(x, y-1)$ para (x, y) en el interior de la imagen y $D_2(x, y) = 0$ en los bordes de la imagen.

Luego, se calcula el cuadrado de cada una, punto a punto, y se suman las dos imágenes. La magnitud del gradiente M_{grad} se obtiene extrayendo la raíz cuadrada de la imagen final. Matemáticamente, los pasos finales mencionados se pueden expresar de la siguiente forma: $M_{grad}(x, y) = \text{raíz} (D_1(x, y)^2 + D_2(x, y)^2)$. Como resultado se obtiene una imagen con manchas brillantes en las cercanías a los bordes de las regiones de contacto.

Se puede apreciar un ejemplo del método propuesto en la figura 3.7, en la cual se muestra el negativo o complemento del gradiente. Para obtener las coordenadas, se calcula el centro de masa de cada mancha binarizada (ecuaciones 2.10 y 2.11).

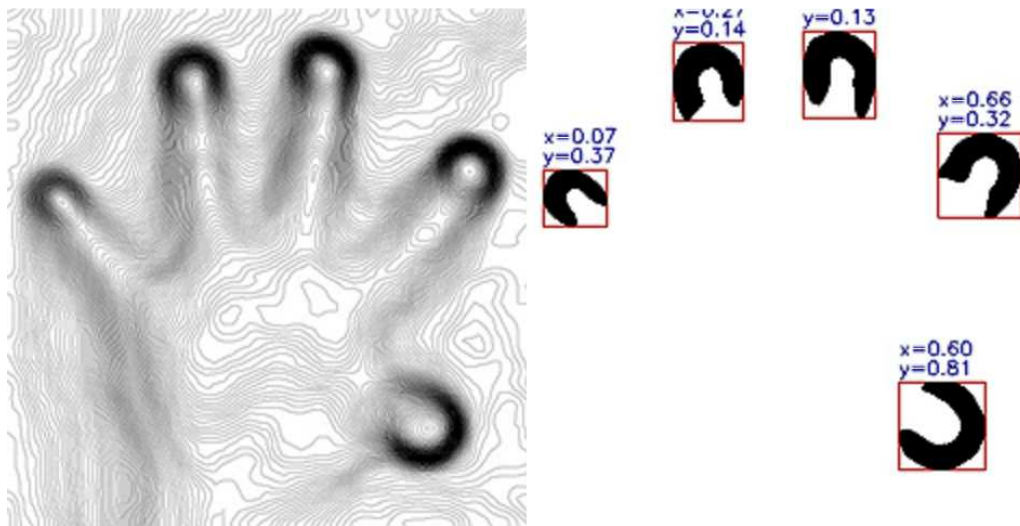


Figura 3.7. A la izquierda está el gradiente de la imagen original. A la derecha, la imagen binaria y se identifica cada región de toque con sus coordenadas.

3.3. PROYECCION DE SOMBRAS

3.3.1. Esquema experimental propuesto

La figura 3.8 muestra de forma simplificada el principio de funcionamiento. El montaje real se muestra en la figura 3.9. Cuando no hay toques sobre la pantalla de acrílico, la cámara observa una línea luminosa horizontal y una línea luminosa vertical, formadas por la luz que emiten las lámparas y que se reflejan en los espejos ubicados en los lados opuestos a ellas.

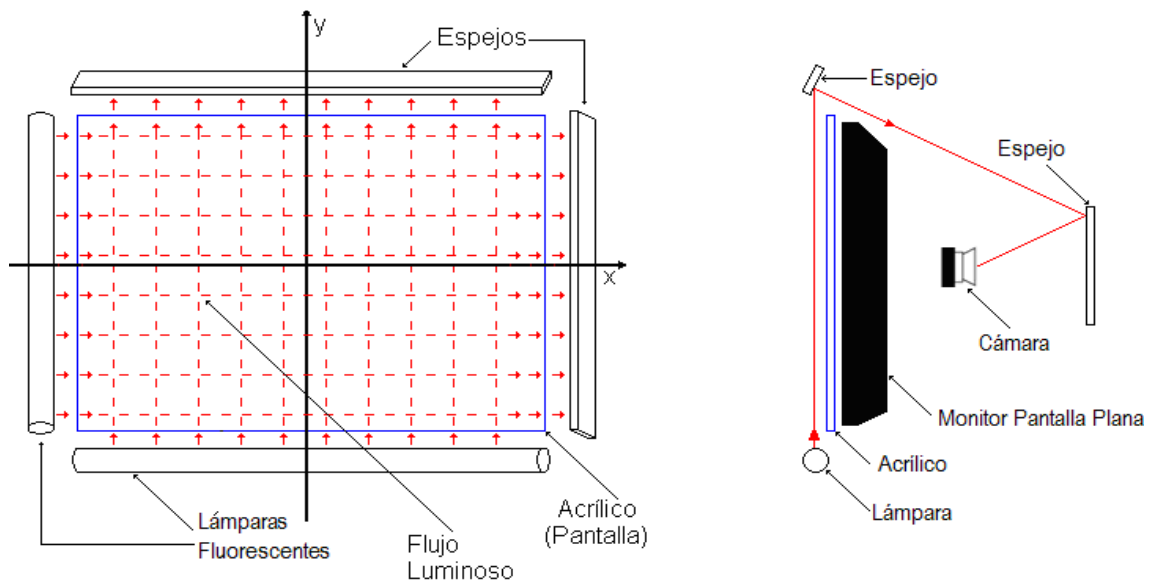


Figura 3.8. Funcionamiento simplificado del sistema visto de frente y de perfil.



Figura 3.9. Vista frontal y de perfil del sistema completo en funcionamiento.

El sistema elaborado consta de una estructura de soporte en aluminio y elementos ópticos: una lámina de acrílico transparente (superficie de contacto), dos lámparas cilíndricas de luz blanca, dos espejos planos (encargados de generar las imágenes de los patrones de sombra), una cámara de video y un espejo plano encargado de obtener la imagen con las dos proyecciones para que pueda ser visible a la cámara.

3.3.2. Método de detección

Cuando se presentan toques sobre la lámina de acrílico, las líneas luminosas pierden continuidad en uno o varios puntos debido a que un dedo produce sombra sobre los patrones de iluminación, tanto en el horizontal como en el vertical. A continuación se describe el método para detectar los puntos que producen tales discontinuidades.

Antes de utilizar el sistema es necesario hacer una calibración previa del mismo. Esta consta de dos partes: seleccionar las regiones de interés de la imagen, correspondientes a los lugares donde se visualizan las lámparas con el fin de determinar los umbrales de la detección; y posteriormente ejecutar un algoritmo que determina la matriz de correspondencia entre la posición real del objeto medida en milímetros y las coordenadas en la imagen, medidas en píxeles.

Detección de las coordenadas dentro de la imagen

El método que se describe a continuación se representa en la figura 3.10. En la etapa de determinación del umbral inicial para cada patrón de luz se hace uso de integrales proyectivas sobre las dos regiones seleccionadas, como se definió en el capítulo II

(ecuaciones 2.8 y 2.9), con el fin de obtener la distribución de iluminación en cada eje cuando no hay toques (parte a y b).

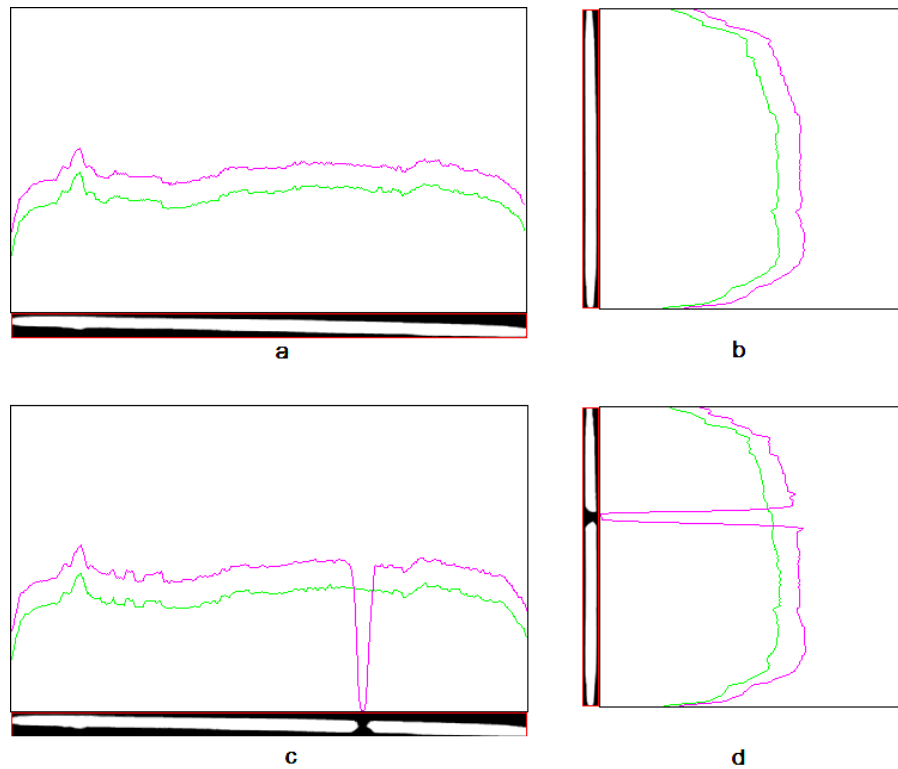


Figura 3.10. Método de integrales proyectivas para detectar toques en el esquema de proyección de sombras. Las zonas blancas corresponden a la imagen de las lámparas, en verde se muestran los umbrales y en violeta las proyecciones en un instante dado.

Lo mencionado anteriormente se realiza de la siguiente forma: sean (x_0, y_0) , (x_1, y_1) las coordenadas que definen el rectángulo de la región horizontal R_h (esquina superior izquierda y esquina inferior derecha, respectivamente) y de la misma forma, (x_2, y_2) , (x_3, y_3) son las que definen la región vertical R_v . Para la región R_h se calcula la proyección horizontal U_x , y para la región R_v se calcula la proyección vertical U_y , como se muestra en las ecuaciones 3.5 a 3.8.

$$U_x = (u_{x_0}, u_{x_0+1}, \dots, u_{x_1}) \quad (3.5)$$

$$U_y = (u_{y_2}, u_{y_2+1}, \dots, u_{y_3}) \quad (3.6)$$

$$u_{x_k} = \frac{1}{y_1 - y_0 + 1} \sum_{y=y_0}^{y_1} I(x_k, y) \quad (3.7)$$

$$u_{y_r} = \frac{1}{x_3 - x_2 + 1} \sum_{x=x_2}^{x_3} I(x, y_r) \quad (3.8)$$

Para evitar la influencia de las fluctuaciones temporales debidas a los cambios en la luz emitida por las lámparas y la luz del ambiente, a cada componente de U_x y U_y se resta un valor de ajuste uniforme, lo suficientemente grande para estar fuera del alcance del ruido y lo suficientemente pequeño comparado con los cambios de iluminación en las zonas de toque. Se encontró experimentalmente que este valor de ajuste debe estar comprendido entre 10 y 20 niveles de gris.

Después de la calibración, cuando el sistema está en uso, se hace un muestreo permanente de imágenes a través del tiempo para determinar cómo cambia la iluminación de las regiones y verificar si se ha tocado la pantalla. Para tal fin, se obtienen las integrales proyectivas de los dos patrones P_x , P_y , de la misma forma como se mostró para los vectores iniciales U_x y U_y en las ecuaciones 3.5 a 3.8.

Posteriormente se comparan los vectores P_x y P_y , con el modelo almacenado del fondo U_x y U_y componente a componente, y se decide que hay punto de toque donde los valores de la proyección actual sean menores que el umbral tanto para el patrón horizontal como para el vertical, es decir, en las regiones donde se producen discontinuidades (parte c y d de la figura 3.10). Si no las hay significa que se tiene un fondo reciente y se descartan los umbrales almacenados para actualizarlos con las nuevas distribuciones.

Corrección de las desviaciones debidas al sistema óptico

Se ha mostrado cómo calcular las coordenadas de los puntos de toque dentro de la imagen. Sin embargo, esto no es suficiente para obtener las coordenadas reales, ya que el sistema óptico utilizado hace que las coordenadas calculadas contengan desviaciones de tipo trapezoidal, como se muestra en la figura 3.11:

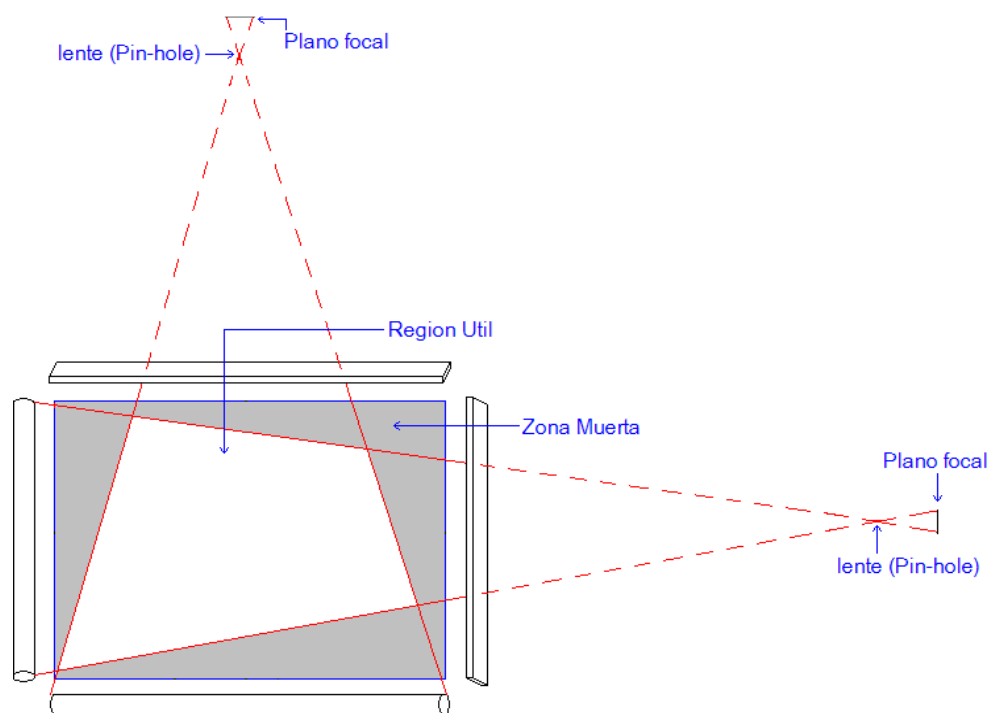


Figura 3.11. Modelo óptico geométrico del sistema de proyección de sombras.

Como se puede apreciar en la anterior figura, las desviaciones en las coordenadas detectadas se deben a que las líneas de proyección de los objetos que tocan la pantalla no son paralelas sino que convergen a un mismo punto, el cual corresponde al objetivo de la cámara. Como resultado se tiene, por ejemplo, si se traza un trapecioide con el dedo como el que determina la región útil en la imagen se detectará un rectángulo.

Para hacer las correcciones debidas a las desviaciones producidas por el sistema óptico en las coordenadas medidas y además aprovechar la necesidad de crear una correspondencia entre la posición real de los puntos de toque en milímetros y la posición medida en pixeles, se propone hacer una transformación geométrica de perspectiva.

Se escogió esta transformación debido a que no solamente se puede utilizar para proyectar objetos 3D en un plano sino también para transformar un cuadrilátero con sus cuatro puntos conocidos en otro cuadrilátero, en este caso un rectángulo en trapezoide. La transformación perspectiva está determinada por una matriz Q con la siguiente estructura [14]:

$$Q = \begin{pmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & 1 \end{pmatrix} \quad (3.9)$$

El cálculo de los coeficientes de la matriz Q se realiza en la etapa de calibración, mostrando en pantalla cuatro círculos con coordenadas conocidas en milímetros. Para determinar las coordenadas de los centros de los círculos en pixeles se pide al usuario tocar en ellos con los dedos en un orden específico. Con los cuatro pares de puntos se pueden calcular los coeficientes teniendo en cuenta que dado un par de puntos correspondientes origen y destino, por ejemplo (x_o, y_o) y (x_d, y_d) , se pueden plantear las siguientes ecuaciones dadas por la transformación:

$$x_d = \frac{c_{11}x_o + c_{12}y_o + c_{13}}{c_{31}x_o + c_{32}y_o + 1} \quad (3.10)$$

$$y_d = \frac{c_{21}x_o + c_{22}y_o + c_{23}}{c_{31}x_o + c_{32}y_o + 1} \quad (3.11)$$

Como se cuenta con cuatro pares de puntos se pueden plantear 8 ecuaciones en total para encontrar las 8 incógnitas c_{ij} de la matriz.

Finalmente, cuando el sistema está en uso después de la calibración, se utiliza la matriz ya calculada para corregir las desviaciones presentes, de la misma forma como se hizo para calcular los coeficientes de la matriz con las ecuaciones 3.10 y 3.11, pero ahora utilizando estas ecuaciones para calcular x_d y y_d .

3.4. TRIANGULACIÓN CON DOS CAMARAS

3.4.1. Esquema experimental propuesto

Este método utiliza el montaje experimental mostrado en la figura 3.12.

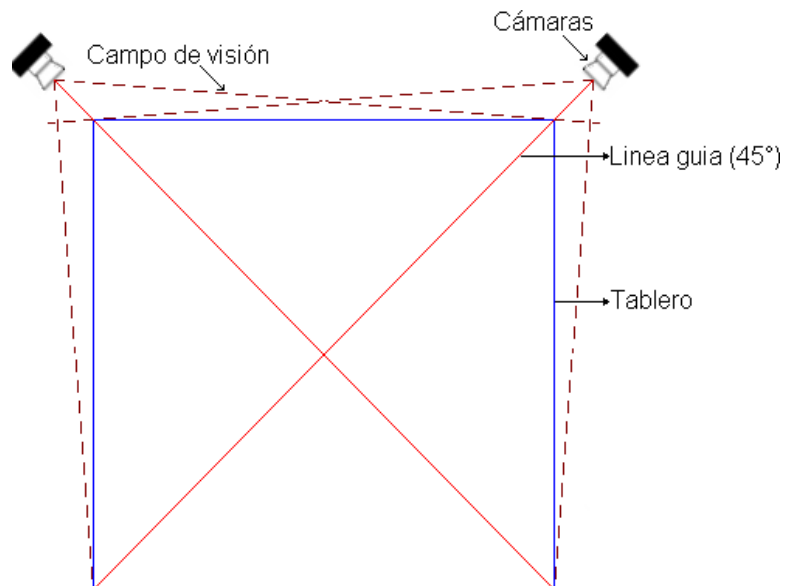


Figura 3.12. Disposición de los componentes para el método de triangulación.

En el montaje se toma en cuenta que cada cámara tiene un ángulo de inclinación de 45 grados respecto a la horizontal, la superficie del tablero es plana y las dos cámaras ven su superficie en ángulo rasante, es decir, la superficie debe coincidir con la fila central de la imagen.

3.4.2. Método de detección

Contando con la condición mencionada anteriormente, es decir que la superficie plana del tablero coincida con la fila central de las imágenes $I(x, y)$ adquiridas por las cámaras, la operación inicial para detectar las coordenadas de toque es la extracción de un perfil de línea de cada imagen $L_h(x) = I(x, h/2)$, como se define en la ecuación 2.6, ubicando la fila $y=h/2$ en el plano más cercano posible al tablero. Aquí, $h/2$ representa la fila central de las imágenes de altura h .

Para facilitar la detección del dedo, se cubrió el fondo de las imágenes en el plano de detección con una superficie de color negro opaco (figura 3.13), debido a que esta es una tonalidad que casi no varía con los cambios de iluminación.

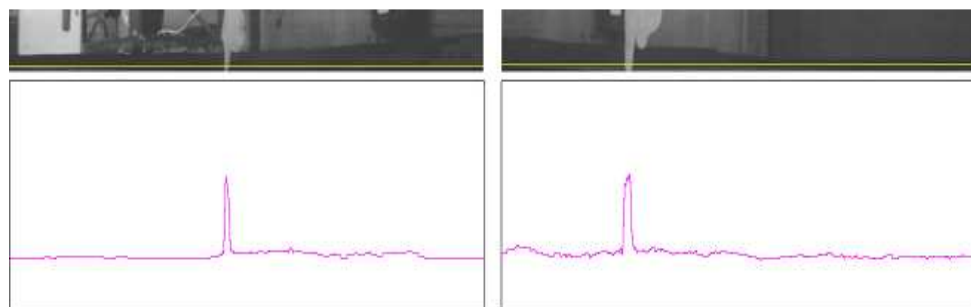


Figura 3.13. Obtención del perfil de línea para la imagen de cada cámara en el esquema de triangulación. La línea amarilla muestra la fila a escanear y las graficas en la parte inferior representan los perfiles de línea.

Como se puede observar, el nivel de gris no es completamente cero en el fondo y varía espacial y temporalmente con la iluminación del entorno, lo que sugiere hacer un pre-procesamiento a las imágenes antes de detectar las coordenadas de toque.

El siguiente paso incluye un suavizado de la imagen con un filtro de media para eliminar las variaciones bruscas entre regiones cercanas y posteriormente un filtro derivativo horizontal como el descrito en la ecuación 3.3. Las derivadas obtenidas se asemejan a las gráficas mostradas en la figura 3.14. Además, se establece un umbral uniforme igual para los dos perfiles de línea, de tal forma que las regiones de contacto corresponderán a aquellos puntos donde el perfil de línea calculado para las nuevas imágenes supere el umbral.

Finalmente, se permite que el umbral varíe de forma dinámica dependiendo de las condiciones de iluminación, evaluando el valor en nivel de gris del perfil de línea en las regiones donde no hay toques.

En la figura 3.14 se muestra el método expuesto, donde x_1 y x_2 son las coordenadas en píxeles de un punto de toque para la cámara izquierda y la cámara derecha respectivamente.

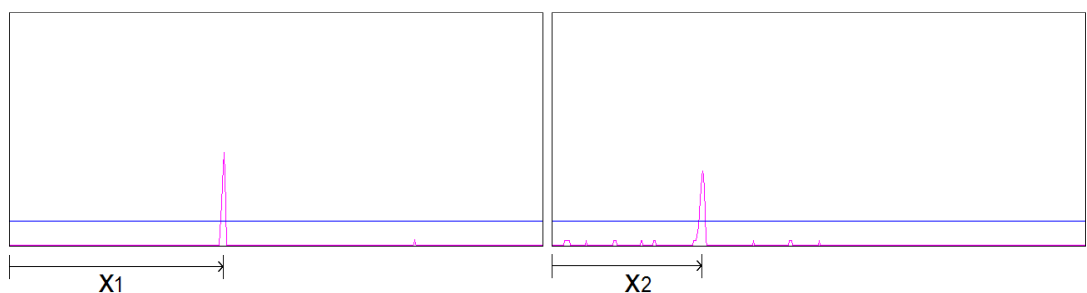


Figura 3.14. Cálculo de x_1 y x_2 en píxeles. La línea azul es el umbral y la curva violeta es la derivada del perfil de línea suavizado.

Cálculo de las coordenadas en el plano del tablero

Se establece un modelo geométrico, como el de la figura 3.15 y se considera cada cámara forma imágenes siguiendo el modelo *pin-hole*. Sean x_1 y x_2 las coordenadas ya calculadas en pixeles de un punto de toque en el perfil de línea de las dos imágenes, w es el ancho de las mismas en pixeles, L es el ancho deseado del tablero (cuadrado), y θ el ángulo de apertura de las cámaras (el campo de visión de la cámara 2 se muestra en la figura mediante líneas a trazos), entonces, las coordenadas del punto (x, y) en el tablero se determinan de la siguiente forma:

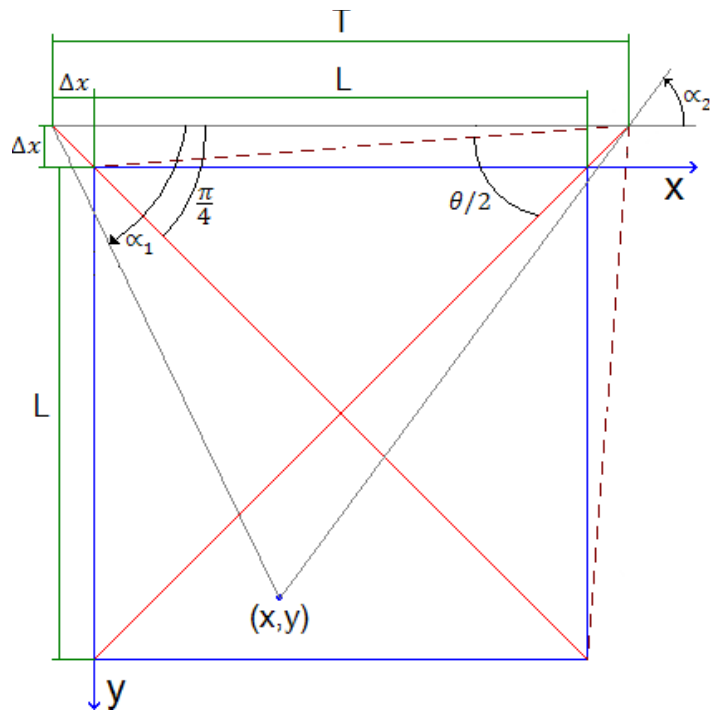


Figura 3.15. Modelo geométrico para determinar las coordenadas (x, y) en el tablero.

Teniendo en cuenta la figura, se ubican las cámaras en los puntos cuyas coordenadas son $(-\Delta x, -\Delta x)$ y $(L+\Delta x, -\Delta x)$ y con sus ejes ópticos alineados con las rectas de 45° (color rojo),

respectivamente. De acuerdo a estas suposiciones se puede calcular las coordenadas del punto de toque (x, y), como también la separación a la que deben estar las cámaras, T, como se muestra a continuación:

- Del triángulo formado por los puntos (0, 0), (L/2, L/2) y (L+Δx, - Δx) se obtiene:

$$\Delta x = \frac{L}{2} \left(\frac{1}{\tan(\theta/2)} - 1 \right) \quad (3.12)$$

- Se puede verificar en la figura que $T=L+2\Delta x$, lo que conduce a la ecuación 3.13:

$$T = \frac{L}{\tan(\theta/2)} \quad (3.13)$$

- De la figura se observa que la cámara 1, ubicada en el punto (0,0) tiene su eje óptico inclinado 45 grados respecto al eje x positivo, mientras que la cámara 2, ubicada en (L+Δx, - Δx), lo tiene inclinado -45 grados respecto al mismo eje.
- Del modelo *Pin-Hole* se tiene que la cámara 1 ve el dedo en la coordenada x_1 de la imagen, correspondiente a un ángulo de visión de $\theta(x_1-w/2)/w$ respecto a su eje óptico y la cámara 2 lo ve con un ángulo de $\theta(x_2-w/2)/w$ respecto al suyo.
- De los resultados de c) y d) se tiene que:

$$\alpha_1 = \frac{\pi}{4} + \frac{\theta(x_1 - \frac{w}{2})}{w} \quad (3.14)$$

$$\alpha_2 = \frac{\theta(x_2 - \frac{w}{2})}{w} - \frac{\pi}{4} \quad (3.15)$$

- Posteriormente se establece un sistema de ecuaciones, donde cada una corresponde a la recta que pasa por el punto de intersección (x, y) y una de las cámaras (líneas de color gris). Estas ecuaciones se muestran a continuación:

$$y + \Delta x = (x + \Delta x)\tan(\alpha_1) \quad (3.16)$$

$$y + \Delta x = (x - T + \Delta x)\tan(\alpha_2) \quad (3.17)$$

- Al solucionar el sistema anterior, finalmente se llega a las expresiones necesarias para calcular x e y:

$$x = \frac{T \tan \alpha_2}{(\tan \alpha_2 - \tan \alpha_1)} - \Delta x \quad (3.18)$$

$$y = \frac{T \tan \alpha_2 \tan \alpha_1}{(\tan \alpha_2 - \tan \alpha_1)} - \Delta x \quad (3.19)$$

De acuerdo a las ecuaciones anteriores se puede deducir que éstas se expresan en unidades de longitud como los centímetros o milímetros. Por esta razón, los valores de x e y se dividen entre la longitud del cuadrado, L, para normalizarse. Algunos parámetros con los cuales se puso a prueba el sistema son los siguientes:

Tabla 3.1. Parámetros reales para la prueba del sistema de triangulación.

Angulo de apertura de las cámaras (Θ)	60°
Ancho deseado del tablero (L)	80 cm
Separación necesaria entre las cámaras (T)	138.6 cm
Ancho de la captura o numero de columnas (w)	640 pixeles
Alto de la captura o numero de filas (h)	480 pixeles

CAPÍTULO IV. TÉCNICAS DE ANÁLISIS DE PUNTOS DE TOQUE Y GENERACIÓN DE ÓRDENES

En el capítulo anterior se mostró como obtener coordenadas bidimensionales de puntos de toque en distintos esquemas ópticos. A continuación se muestra cómo se puede crear una plataforma software común a los montajes, encargada de procesar los datos numéricos y convertirlos en órdenes para manipular diferentes aplicaciones.

4.1. SEGUIMIENTO DE OBJETOS EN DOS DIMENSIONES

El problema general que intentan solucionar las diferentes técnicas relacionadas con seguimiento de objetos es identificar y determinar la trayectoria de varios puntos o partículas indistinguibles (círculos de igual radio, por ejemplo) moviéndose en una imagen [20]. Para solucionar este problema, los distintos enfoques utilizan frecuentemente todas o algunas de las siguientes suposiciones, dependiendo los requerimientos de la aplicación específica:

- La ubicación de un objeto cambia suavemente en el tiempo.
- La velocidad de un objeto cambia suavemente en el tiempo: esto incluye tanto la magnitud como la dirección.
- Un objeto puede estar en una única posición en la imagen en un instante de tiempo dado.
- Dos objetos no pueden ocupar la misma posición al mismo tiempo.

Algunos enfoques más utilizados para realizar de forma eficiente el seguimiento son el criterio de la mínima distancia, proveniente del análisis de los primeros vecinos de un objeto entre imágenes consecutivas; el criterio de la trayectoria más suave, que requiere minimizar los cambios en la magnitud y dirección de las velocidades de cada uno y el modelamiento de la dinámica de los objetos, con el fin de predecir mediante estimadores las posiciones futuras si ya se conoce matemáticamente el comportamiento que estos tienen.

Para las plataformas multitoque implementadas, se encontró experimentalmente que en los sistemas de vista directa de la superficie cumplen con la condición d) puesto que físicamente un dedo no puede ocupar el lugar de otro y por simplicidad el seguimiento se puede realizar aplicando el criterio de la mínima distancia, en contraste con los sistemas de proyección para los cuales en ocasiones se superponen los objetos y se necesita del análisis de velocidades para determinar su trayectoria al separarse. No se utilizó modelamiento matemático para la predicción de las posiciones futuras porque el movimiento de los objetos depende de las decisiones instantáneas y subjetivas de las personas.

A menudo, para identificar un objeto y diferenciarlo de los demás se utiliza una etiqueta numérica que se conserva y lo acompaña durante todo su tiempo de existencia y su trayectoria.

4.1.1. Primeros Vecinos y Criterio de la mínima distancia

Para determinar la posición de cada objeto en el cuadro siguiente se puede hacer una asociación simple mediante el método de los primeros vecinos. Esta aproximación

compara las coordenadas de los puntos de un conjunto con los de otro conjunto y los asocia creando correspondencias, a menudo midiendo distancias Euclidianas [8].

Cuando se hayan calculado todas las posibles distancias entre un punto ya identificado y etiquetado en el instante $t-\Delta t$ (imagen anterior) y sus primeros vecinos en t (imagen actual) se determina que dicho objeto corresponde a un objeto en el instante actual si la distancia entre ellos es la mínima de todas las que se midieron [21]. A esto se le denomina criterio de la mínima distancia y se explica con la figura 4.1.

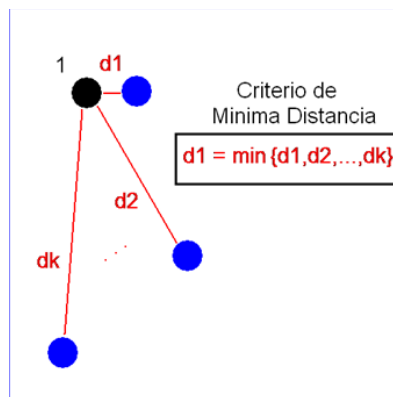


Figura 4.1. Explicación del criterio de la mínima distancia. El punto con etiqueta 1 aparece en una imagen; los puntos azules aparecen en la siguiente imagen. La etiqueta 1 será cedida al objeto que se encuentra a la distancia $d1$ porque tal distancia es la más pequeña.

Imagen tomada de [17].

4.1.2. Criterio de las trayectorias más suaves

El algoritmo general utiliza las suposiciones mencionadas anteriormente para calcular el conjunto más suave de caminos a través de una secuencia de puntos. A continuación se

define el concepto de trayectoria, suavidad de una trayectoria, y se muestra el criterio que permite determinar la ruta que sigue cada objeto [20].

Trayectoria

Si un objeto con etiqueta i es observado en instantes de tiempo $t= 0, 1, \dots, n$; entonces se define trayectoria como la señal unidimensional (Ver definición en el capítulo II), de variable t , que representa la secuencia de puntos $p_{i,t}$ definida por:

$$T_i = (p_{i,0}, p_{i,1}, \dots, p_{i,n}) \quad (4.1)$$

Suavidad de una trayectoria:

Sea $V_{i,t}$ la velocidad vectorial del objeto i en un instante t (suponiendo una tasa de muestreo de imágenes constante), definida por:

$$V_{i,t} = p_{i,t+1} - p_{i,t} \quad (4.2)$$

Entonces, si w es un valor entre 0 y 1, se define la suavidad de la trayectoria del objeto i en un punto $p_{i,t}$ como:

$$S_{i,t} = w \frac{V_{i,t-1} * V_{i,t}}{\|V_{i,t-1}\| \|V_{i,t}\|} + 2(1 - w) \frac{\sqrt{\|V_{i,t-1}\| \|V_{i,t}\|}}{\|V_{i,t-1}\| + \|V_{i,t}\|} \quad (4.3)$$

Es fácil verificar que el primer miembro (* representa producto punto) hace referencia a la suavidad en el cambio de la dirección del vector velocidad y el segundo miembro se refiere a la suavidad en el cambio de la magnitud (media geométrica normalizada).

Criterio de Selección

Se asume que se extraen m puntos a seguir en n imágenes. Entonces, la suavidad total de un conjunto de trayectorias se define de la siguiente forma:

$$S_T = \sum_{i=1}^m \sum_{t=1}^{n-1} S_{i,t} \quad (4.4)$$

Con el conjunto de todas las combinaciones de trayectorias posibles, se escoge el conjunto de ellas que presenten la máxima suavidad total S_T .

4.2. IDENTIFICACIÓN DE GESTOS BASADOS EN POSICIÓN Y TRAYECTORIAS

Se denomina gesto, en forma general, a una configuración específica cambiante en el tiempo de un conjunto de objetos moviéndose, la cual se puede reconocer mediante su disposición geométrica en el plano, su trayectoria, su velocidad, entre otras características cinemáticas.

En este trabajo, como un aporte al desarrollo de Interfaces Naturales de Usuario y a mejorar la interacción entre el Hombre y el Computador en el contexto tecnológico regional, se muestran dos métodos de detección de gestos hechos con los dedos sobre una plataforma táctil: el primero se refiere a la identificación de trayectorias especiales de un solo toque, denominadas íconos gestuales; mientras que el segundo muestra las

operaciones matemáticas necesarias para realizar las acciones de redimensionamiento y rotación de objetos en aplicaciones gráficas utilizando dos toques.

4.2.1. Iconos gestuales con un toque

Este método de identificación de gestos implica un entrenamiento previo del sistema para reconocer de forma inteligente algunas trayectorias fácilmente repetibles y de clara asociación a las actividades o tareas cotidianas de los usuarios. En este trabajo, se elaboraron algoritmos para reconocer formas simples de una sola trayectoria mediante técnicas de aprendizaje de máquina que se explican a continuación.

Redes Neuronales Artificiales

Una red neuronal es un conjunto de neuronas interconectadas. Una neurona es un módulo que consta de varias entradas x_i que se suman por medio de ponderaciones o pesos generando una señal u , posteriormente sufren una transformación matemática definida por una función f , denominada **función de activación**, para producir una única salida $y = f(u)$. El esquema de funcionamiento de una neurona se muestra en la figura 4.2.

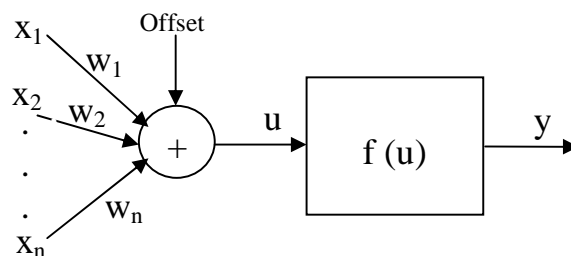


Figura 4.2. Arquitectura de una Red Neuronal Artificial.

La ecuación 4.5 explica la forma como se genera la señal de entrada u :

$$u = Offset + \sum_{i=1}^n x_i w_i \quad (4.5)$$

La arquitectura más común de red neuronal es el perceptrón de múltiples capas, que consta de una capa de neuronas en la entrada, una o varias capas ocultas y una capa en la salida. La figura 4.3 muestra un ejemplo de un perceptrón con dos neuronas en la entrada, tres neuronas en la capa oculta y una en la capa de salida:

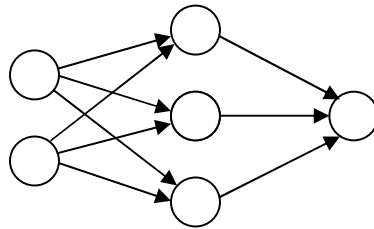


Figura 4.3. Ejemplo de un perceptrón de tres capas.

Para poner a funcionar la red, es necesario realizar un entrenamiento previo, el cual implica el cálculo de los pesos adecuados, de tal forma que para entradas conocidas se obtengan valores conocidos de salida.

Una red neuronal puede utilizarse como un clasificador. En este caso se van a clasificar distintos tipos de trayectorias. La arquitectura del perceptrón será de la siguiente forma: el número de neuronas de entrada será el número de características extraídas a cada trayectoria; el número de neuronas de salida será el número de tipos de trayectorias a clasificar y existirá una o más capas ocultas. El entrenamiento de la red se hace extrayendo características de varias muestras conocidas pertenecientes a cada uno de los tipos de iconos gestuales a identificar.

En la etapa de prueba posterior al entrenamiento, se realizan diversos íconos gestuales sobre la superficie de detección, se extraen las características requeridas de la trayectoria, se introducen a la entrada de la red neuronal entrenada y la salida será un vector de valores en un intervalo entre 0 y 1, correspondientes a la probabilidad de correspondencia entre la muestra y cada clase de gestos. Finalmente, se aplica un umbral al máximo valor de salida para determinar si la trayectoria de prueba pertenece o no a la clase más probable.

4.2.2. Gestos de traslación, rotación y redimensionamiento con dos toques

El objetivo es interpretar el movimiento de dos puntos de toque y convertirlo en transformaciones de traslación, rotación y redimensionamiento sobre objetos en aplicaciones gráficas.

Un objeto cualquiera en una secuencia de imágenes en un instante de tiempo discreto $t=t_i$, es señalado mediante dos puntos de coordenadas $(x_{i1}, y_{i1}), (x_{i2}, y_{i2})$, tocándolo con los dedos. En el siguiente instante en $t=t_f$ los dedos se han movido de sus coordenadas iniciales a sus nuevos valores $(x_{f1}, y_{f1}), (x_{f2}, y_{f2})$ y se muestra el efecto generado.

Para lograr el objetivo propuesto, es importante tener en cuenta el orden en el cual se realizan las transformaciones. En este caso se propone realizar primero la traslación del objeto, ubicando el punto medio de los dos pares coordenados iniciales en la posición del punto medio de los pares coordenados finales; luego realizar la rotación cuyo eje se ubica en el punto medio de los puntos de toque de la segunda imagen y finalmente la operación de redimensionamiento. A continuación se muestra de forma gráfica y analítica cada una de estas operaciones.

Traslación

Esta operación implica mover el objeto mediante un vector de traslación T , como se muestra en la figura 4.4, aplicado al punto medio de los puntos iniciales de toque, manteniendo su orientación. En la figura, los puntos azules representan los toques en el instante anterior y los rojos en el instante actual. La misma representación se utilizará para las operaciones de rotación y redimensionamiento.

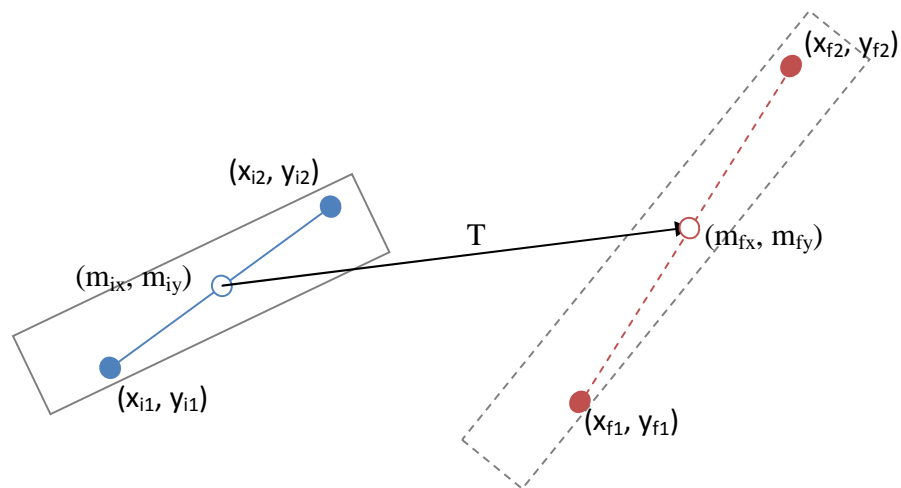


Figura 4.4. Operación de traslación con dos toques.

Las coordenadas del punto medio entre los puntos iniciales y las del punto medio entre los puntos finales se calculan de la siguiente forma:

$$(m_{ix}, m_{iy}) = \left(\frac{x_{i1} + x_{i2}}{2}, \frac{y_{i1} + y_{i2}}{2} \right) \quad (4.6)$$

$$(m_{fx}, m_{fy}) = \left(\frac{x_{f1} + x_{f2}}{2}, \frac{y_{f1} + y_{f2}}{2} \right) \quad (4.7)$$

El vector de desplazamiento T se puede expresar de la siguiente forma:

$$(T_x, T_y) = (m_{fx} - m_{ix}, m_{fy} - m_{iy}) \quad (4.8)$$

Finalmente, las coordenadas de los puntos resultantes de la operación de traslación (x_{t1} , y_{t1}) y (x_{t2} , y_{t2}) se calculan sumando el vector T a cada punto origen:

$$(x_{t1}, y_{t1}) = (x_{i1} + T_x, y_{i1} + T_y) \quad (4.9)$$

$$(x_{t2}, y_{t2}) = (x_{i2} + T_x, y_{i2} + T_y) \quad (4.10)$$

Rotación

El paso a seguir es cambiar la orientación de los puntos calculados a la orientación deseada. Esto se muestra en la figura 4.5.

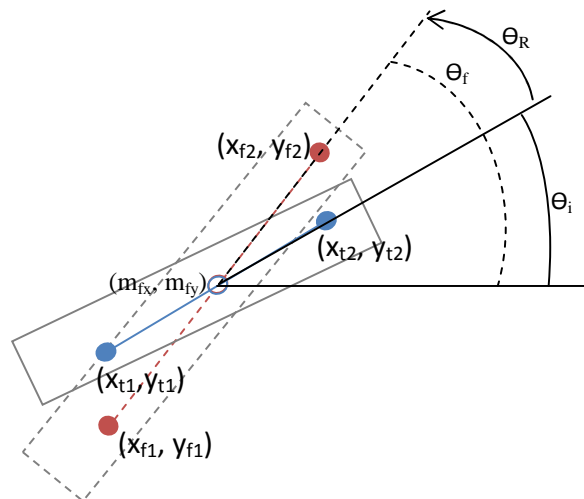


Figura 4.5. Operación de rotación con dos toques.

El ángulo de inclinación inicial y el ángulo de inclinación final están determinados por las siguientes ecuaciones:

$$\theta_i = \arctan \left(\frac{y_{t2} - y_{t1}}{x_{t2} - x_{t1}} \right) \quad (4.11)$$

$$\theta_f = \arctan \left(\frac{y_{f2} - y_{f1}}{x_{f2} - x_{f1}} \right) \quad (4.12)$$

El ángulo de rotación necesario es igual a la resta entre el ángulo final y el ángulo inicial, es decir:

$$\theta_R = \theta_f - \theta_i \quad (4.13)$$

Finalmente, las coordenadas de los nuevos puntos obtenidos mediante la transformación de rotación, respecto al punto de giro (m_{fx}, m_{fy}) son los siguientes [16]:

$$\begin{pmatrix} x_r \\ y_r \end{pmatrix} = \begin{pmatrix} \alpha & \beta & (1 - \alpha)m_{fx} - \beta m_{fy} \\ -\beta & \alpha & \beta m_{fx} + (1 - \alpha)m_{fy} \end{pmatrix} \begin{pmatrix} x_t \\ y_t \\ 1 \end{pmatrix} \quad (4.14)$$

, donde $\alpha = \cos \theta_R$ y $\beta = \sin \theta_R$.

Redimensionamiento

Esta operación finaliza la etapa del procesamiento ya mencionado de los dos toques, como se muestra en la figura 4.6.

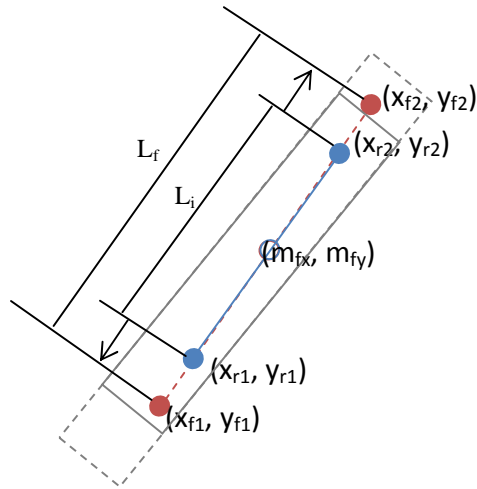


Figura 4.6. Operación de redimensionamiento con dos toques.

En la figura, las distancias entre los puntos iniciales, L_i , y la distancia entre los puntos finales, L_f , se determinan mediante las siguientes ecuaciones:

$$L_i = \sqrt{(x_{r2} - x_{r1})^2 + (y_{r2} - y_{r1})^2} \quad (4.15)$$

$$L_f = \sqrt{(x_{f2} - x_{f1})^2 + (y_{f2} - y_{f1})^2} \quad (4.16)$$

El factor de escala será $E=L_f/L_i$. A continuación se muestra la operación final requerida, que consta de dos partes: la primera es establecer el punto de origen de la operación en (m_{fx}, m_{fy}) y la segunda el redimensionamiento como tal, de la siguiente forma:

$$(x_{e1}, y_{e1}) = (m_{fx}, m_{fy}) + E(x_{r1} - m_{fx}, y_{r1} - m_{fy}) \quad (4.17)$$

$$(x_{e2}, y_{e2}) = (m_{fx}, m_{fy}) + E(x_{r2} - m_{fx}, y_{r2} - m_{fy}) \quad (4.18)$$

De lo anterior, se puede comprobar mediante las transformaciones anteriores que los puntos calculados (x_{e1}, y_{e1}) y (x_{e2}, y_{e2}) ocupan el lugar de (x_{f1}, y_{f1}) y (x_{f2}, y_{f2}) , respectivamente.

4.3. GENERACIÓN DE EVENTOS Y ÓRDENES AL COMPUTADOR

Esta sección muestra como se aprovechan los distintos gestos detectados, tanto simples (un solo toque o múltiples toques individuales sin una configuración programada para detección) como complejos (los vistos en la sección 4.2), con el fin de dar órdenes de control al computador en el momento que el usuario interactúa con él.

En este trabajo se han utilizado dos formas de enviar órdenes al computador: utilizando funciones del sistema operativo y mediante comunicación de datos entre aplicaciones, sin intervenir directamente con el funcionamiento del sistema.

4.3.1. Funciones de control del sistema operativo

En el sistema operativo existen diversas características, que van desde variables de entorno, registros de ocurrencia de eventos (por ejemplo el movimiento del ratón, un clic o una tecla presionada en el teclado) o direcciones de memoria de las entradas y salidas de los periféricos. Mediante estas características se puede controlar el funcionamiento del computador dependiendo el ambiente de desarrollo.

Uno de los propósitos de un sistema multitoque es la emulación de diversas órdenes que provienen de periféricos de entrada de texto o dispositivos señaladores. Para tal fin, en este trabajo se aprovechan las funciones que ofrece el sistema operativo Windows para

detección y emulación de eventos relacionados con el teclado y el ratón, utilizando el lenguaje de programación C++.

Para controlar a bajo nivel estas funciones, los compiladores de C++ ofrecen la biblioteca de cabecera "windows.h" que permite generar eventos simulados del ratón y teclado como presionar o liberar cada botón, mover el ratón, presionar o girar la rueda, presionar o liberar una tecla, emular combinaciones de teclas, entre otras.

Se pusieron a prueba las plataformas multitoque elaboradas para generar órdenes al sistema operativo en los siguientes casos:

a) Con retroalimentación visual:

- Con un dedo: al tocar se emula la acción de presionar el botón izquierdo del ratón y al dejar de hacerlo se emula la acción de liberar. Esto permite hacer clic simple (presionar y liberar en el mismo sitio), doble clic (evento especial que requiere hacer clic dos veces en un intervalo menor en milisegundos al establecido en las variables del sistema operativo), sostener y mover objetos en las ventanas del explorador de carpetas (presionar, mover y liberar en un sitio diferente).
- Con dos dedos: dependiendo la aplicación en la que se trabaja, se realizan las operaciones de zoom, giro y movimiento de objetos mediante combinaciones del teclado y el ratón. Por ejemplo: *Acrobat Reader* de *Adobe* realiza zoom con la combinación CTRL+RUEDA DEL RATON; *Google Earth* realiza zoom con el movimiento de la rueda y realiza giros con CTRL+RUEDA DEL RATON.

- b) Sin retroalimentación visual: se pueden utilizar los esquemas experimentales mencionados para generar órdenes similares a los touchpads.
- Con un dedo: mover el cursor del ratón.
 - Con dos dedos: dependiendo la configuración de la aparición y los movimientos de los mismos se emulan diferentes tipos de eventos, por ejemplo: se introduce un dedo y posteriormente, al introducir el segundo se emula la acción de presionar el botón izquierdo del ratón en la posición del primer toque con el fin de obtener clic, doble clic o clic sostenido.
- c) Eventos independientes de la realimentación visual: se utilizaron los íconos gestuales programados para que al identificar la aparición de uno de ellos, el sistema operativo recibe la orden de ejecutar un programa específico o tarea especial.

4.3.2. Métodos de comunicación entre aplicaciones

Existen varias formas de hacer que un programa envíe datos a otro, ejecutándose en paralelo continuamente, sin necesidad de convertirse en un módulo o subrutina, con el fin de independizar los algoritmos de detección de toques de los algoritmos de generación de eventos, puesto que cada aplicación puede poseer su propio funcionamiento, dependiendo el contexto en el cual se desarrolle: aplicaciones gráficas, de edición de texto, de control del sistema operativo y periféricos, etc.

No obstante, no todos estos métodos cumplen con el componente de usabilidad multiplataforma, importante en muchos campos, sobretodo en el diseño de hardware y

software. Por tal razón, uno de las técnicas más utilizadas en el ambiente de las tecnologías multitáctiles es el Protocolo TUIO [22], el cual se describe a continuación:

Protocolo TUIO

Las siglas TUIO hacen referencia al protocolo diseñado para satisfacer los requerimientos de interfaces de usuario que utilizan plataformas tipo mesa táctil (*Tangible User Interface protocol*), creado por los programadores del proyecto *ReactiVision*. Este protocolo define propiedades comunes de objetos controladores de la superficie (fiducias) como también de gestos de los dedos y manos, para transmitirse desde una aplicación servidor hacia aplicaciones clientes. La actual implementación del protocolo permite que las aplicaciones servidor o cliente se desarrollen independientemente en alguno de los siguientes lenguajes: Java, C++, PureData, Max/MSP, SuperCollider y Flash.

TUIO define dos clases principales de mensajes: *Set* y *Alive*. Los mensajes *Set* se utilizan para informar sobre el estado del objeto, como su posición u orientación, mientras que los mensajes *Alive* indican el conjunto actual de objetos presentes tocando la superficie, usando una lista de indicadores numéricos. El protocolo utiliza transporte de datos UDP (protocolo para comunicaciones locales y en red) para hacer eficiente la comunicación. Se empaquetan varios mensajes *Set* de forma redundante para completar la capacidad de transporte UDP, con el fin de evitar pérdidas de información y en cada uno se incluye un mensaje *Alive*.

TUIO fue implementado utilizando un protocolo más general denominado *Open Sound Control*, razón por la cual debe seguir su sintaxis. El formato general para el envío y recepción de los mensajes es el siguiente para mensajes *Set* y *Alive*, respectivamente:

/tuio/[Nombre del Perfil] set [numero de identificador del objeto] [Parámetros]

/tuio/[Nombre del Perfil] alive [lista de identificadores de objetos activos]

En el anterior formato, dependiendo que tipo de objetos se están identificando en la superficie táctil se tienen distintos Nombres de Perfil que se pueden utilizar en el protocolo, por ejemplo un elemento sobre la superficie puede ser de dos tipos: objeto (fiducia o marcador) o cursor (un solo punto), y estos a su vez pueden funcionar en dos o mas “dimensiones” dependiendo de los requerimientos de la aplicación, es decir, si en el campo de la lista de Parámetros se necesitan coordenadas, ángulos, velocidades o aceleraciones de los elementos en 2D o 3D.

En este trabajo, se utilizaron las librerías de código abierto del protocolo TUIO en C++ que distribuye *ReactiVision* de forma gratuita, con el fin de transferir coordenadas de puntos simples en dos dimensiones hacia otras aplicaciones en C++ y Flash de forma local, es decir, en el mismo computador donde se lleva a cabo la detección de los puntos de toque.

CAPITULO V. ANALISIS DE RESULTADOS.

5.1. DESEMPEÑO Y LIMITACIONES DE LOS MÉTODOS DE DETECCIÓN

5.1.1. Montaje de FTIR

Como se pudo observar en el capítulo III, el sistema de visión implementado permite obtener directamente en la imagen el patrón de manchas formadas por los puntos de toque (figura 3.2) gracias a su alto contraste respecto al fondo, por lo tanto el algoritmo de detección de coordenadas únicamente necesita realizar operaciones de cálculo del centro de masa, por lo cual el sistema funciona con alta rapidez. Una ventaja adicional de la vista directa de la superficie es que no hay limitaciones en cuanto al número de objetos que pueden tocar la pantalla.

En cuanto al sistema de iluminación y visión por computador se puede mencionar que no es necesario construir una caja cerrada. Únicamente se necesita evitar la incidencia directa en la cámara o mediante reflexión en el acrílico de radiación parásita intensa de luz infrarroja.

Respecto al hardware, ya que el sistema es iluminado completamente por LED infrarrojos es necesario un gran número de ellos, como se muestra en la figura 5.1, tanto para dar la suficiente intensidad luminosa, es decir, para obtener un buen contraste, como también para que la distribución de la iluminación sea uniforme en toda la superficie. Esto es una desventaja respecto a otros sistemas, puesto que requiere también el diseño del sistema electrónico adecuado.

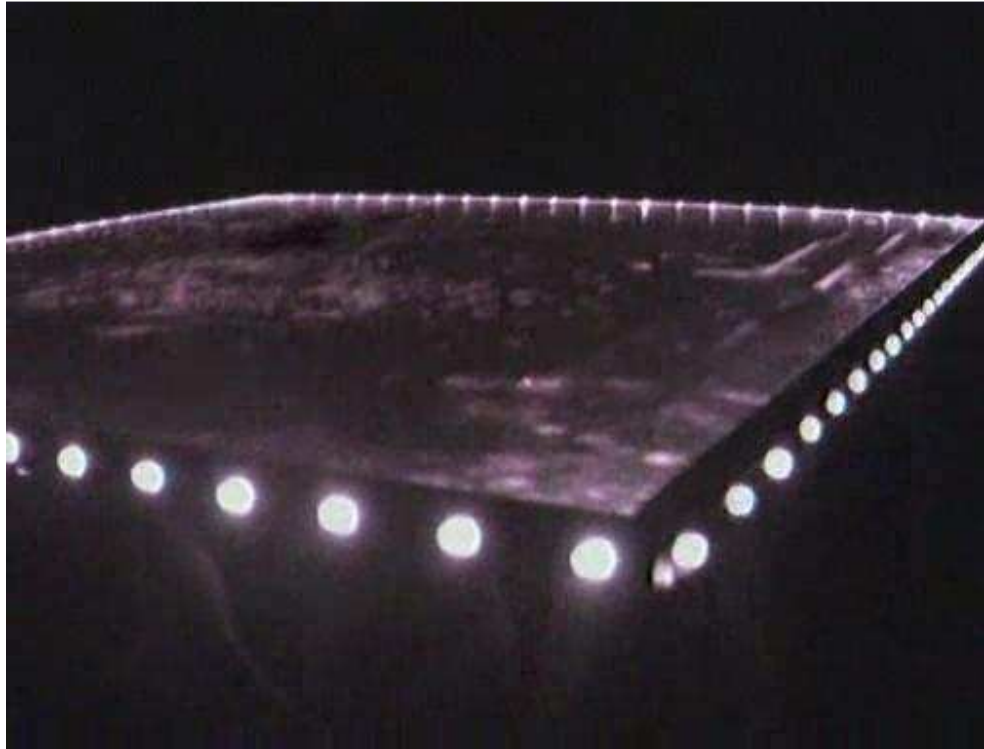


Figura 5.1. Numerosos LED infrarrojos para iluminar la superficie de detección con FTIR.

Teniendo en cuenta que el fenómeno de FTIR depende de las propiedades de los materiales en contacto, el esquema empleado es adecuado para la detección de toques de los dedos directamente sobre el acrílico (debido a la humedad que posee la piel) y además requiere de cierta presión para que la iluminación obtenida en estas zonas sea suficiente. Si se utilizan materiales no húmedos, el efecto no se observa. Esto tiene como consecuencia que únicamente se puedan detectar toques con los dedos y no se puedan reconocer marcadores o fiducias.

Para reconocer algunos objetos (por ejemplo la punta de un lápiz), el grupo NUI y la literatura en general recomienda utilizar una película de algún material de silicona sobre

el acrílico [8], lo cual permite además identificar las zonas de toque haciendo menos presión.

5.1.2. Montaje de Iluminación Difusa

El sistema hardware es mucho más fácil para construir que el de FTIR, debido a que no requiere un sistema electrónico adicional. Como la iluminación se hace desde debajo de la superficie, es posible identificar algunos objetos o fiducias además de los dedos o las manos. Sin embargo, es más difícil obtener iluminación uniforme sobre la superficie y los toques presentan menor contraste.

Aunque la etapa de detección requiere algunas operaciones de procesamiento previo (conversión a escala de grises, filtrado pasa bajas para eliminación de ruido y suavizado, filtrado pasa altas con gradiente para detectar cambios entre el fondo y las zonas de contacto, operaciones aritméticas punto a punto para obtener la magnitud y binarización), el costo computacional es muy bajo, incluso en computadores cuyas especificaciones de hardware no son comparables con las de los computadores nuevos, referentes a la cantidad de memoria o velocidad del procesador. Se determinó que el sistema funciona adecuadamente en cualquier computador de características aceptables utilizando un tamaño de captura de 640 por 480 pixeles para obtener un tiempo aproximado de muestreo y procesamiento de imágenes de 200 milisegundos.

Al igual que como ocurre con el método de Reflexión Total Interna Frustrada, la detección de las coordenadas se hace rápidamente mediante el cálculo de los centros de masa de los objetos detectados en la imagen binaria y la vista directa de las regiones de toque permite que no hayan restricciones asociadas al número de objetos que pueden tocar la pantalla.

5.1.3. Montaje de proyección de sombras

Como se pudo observar, la etapa de detección requiere de varios procedimientos previos a la determinación de los puntos de toque. Sin embargo, cualquier computador adquirido en los últimos 5 a 10 años puede realizar las operaciones requeridas (cálculo constante de integrales proyectivas, comparación con el umbral almacenado, actualización dinámica del perfil de iluminación del fondo y corrección de las desviaciones mediante transformación de perspectiva) en corto tiempo. Esto se debe a que las operaciones mencionadas solamente se realizan en las regiones de interés correspondientes a las zonas donde se produce la imagen de las lámparas cilíndricas, desechando la mayor parte de los datos de las imágenes. De lo mencionado cabe resaltar la importancia que tiene el diseño del sistema de actualización dinámica del umbral, ya que esta técnica muestra un ejemplo muy básico de aprendizaje de máquina porque el sistema cada instante está adaptándose a las nuevas condiciones de iluminación.

Las ventajas que ofrece el uso del lenguaje C++ y OpenCV (la cual está optimizada para operaciones de visión por computador) se notan de forma clara en la velocidad de procesamiento. Además por el hecho de utilizar integrales proyectivas, los algoritmos funcionan con gran rapidez gracias al procesamiento de señales unidimensionales y no las imágenes completas.

Adicionalmente a la rapidez de los algoritmos de detección, existe una característica del sistema que influye mucho en la disminución del tiempo de muestreo de imágenes: debido a que solamente es de interés la detección de las sombras proyectadas sobre la imagen de las lámparas, las cuales son de alta luminosidad, durante la etapa de calibración del sistema se configura el controlador de la cámara, dependiendo sus especificaciones técnicas y el sistema operativo empleado, para que esta utilice el menor tiempo posible de exposición o tiempo de integración. Con la reducción del tiempo de

integración de la cámara se logra que únicamente se vea en la imagen las regiones más iluminadas, evitando así algunas fluctuaciones de la luz del ambiente.

Limitaciones en el número de toques

Debido a que la cámara no visualiza directamente la zona táctil, el sistema está limitado en el número posible de toques que se pueden detectar, dependiendo de cómo se introducen en la pantalla. Por ejemplo, si se introducen los toques únicamente en la región útil:

- Se introduce un solo dedo: se detecta e identifica sin problemas.
- Se introduce primero 1 dedo y después otro: se detectan e identifican sin problemas y se pueden seguir en las posteriores imágenes.
- Se introducen dos dedos a la vez: si están alineados de tal forma que producen una sombra en un eje y dos sombras en el otro entonces se pueden detectar e identificar en la imagen actual y seguir en las imágenes posteriores. Si no es así, es decir, si hay dos sombras horizontales y dos verticales en la imagen actual, se presentan dos posibles pares de puntos (un par falso y uno verdadero) que producen el mismo patrón, lo cual corresponde a un caso de ambigüedad.
- Más de dos dedos: entre mayor sea el número de toques sobre la pantalla, mayor es la probabilidad de presencia de ambigüedades y son cada vez menos los casos en que éstas se resuelven.

También existe un caso de ambigüedad cuando se ubica un dedo en la zona útil y otro dedo en zona muerta, de tal forma que en un eje se detectan dos toques y en el otro se detecta uno. Esto conduce a generar un total de tres posibles combinaciones de pares de puntos que producen el mismo patrón de sombras.

La figura 5.2 resume los principales casos de ambigüedad, los cuales se presentan con dos dedos. Para más de dos toques, se presentan casos similares a los que se muestran.

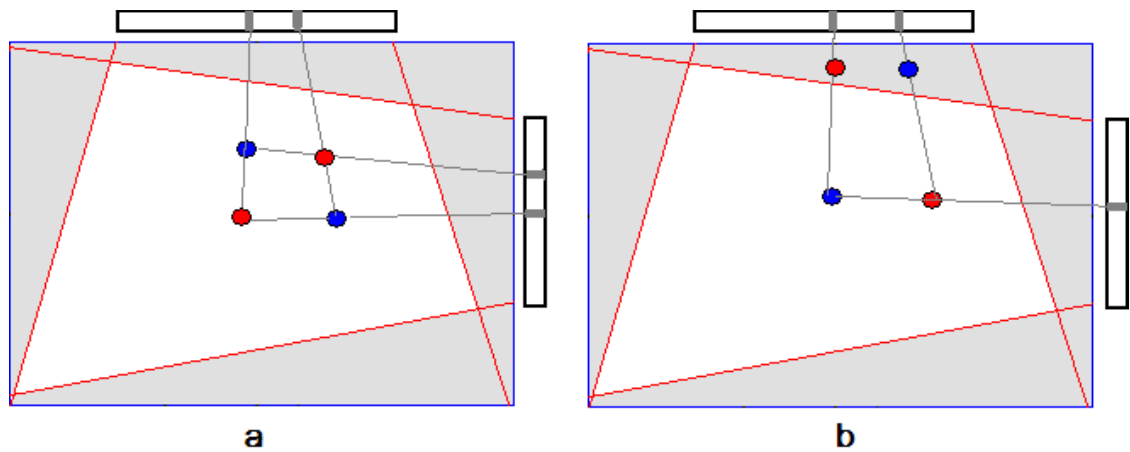


Figura 5.2. Casos de ambigüedad con dos toques para un sistema de proyecciones. En color azul se representan los toques reales y en rojo los toques falsos posibles. Parte a: dos toques que se introducen simultáneamente en la región útil. Parte b: dos toques en cualquier instante, con uno de ellos en zona muerta.

Debido a los anteriores resultados, se limitó el sistema a funcionar únicamente con un dedo o posteriormente introduciendo un segundo dedo en la zona útil en regiones donde éstos no queden alineados a lo largo del eje horizontal o vertical. Para todos los demás casos (dos dedos alineados, dos dedos que se introducen al mismo tiempo o más de dos toques) el sistema no responde.

Se puede dar solución al problema de las ambigüedades en las zonas muertas diseñando un sistema en el cual las lámparas sean lo suficientemente largas para que su proyección en los lados opuestos de la pantalla cubra todo el ancho y altura del área de despliegue.

5.1.4. Montaje de triangulación con dos cámaras

En cuanto al diseño de los componentes mecánicos, se comprobó que se necesita mucha precisión en el momento de calibrar el sistema, puesto que, teóricamente la superficie plana del tablero debe corresponder exactamente a la fila central de las imágenes. En la realidad, contando con los materiales y recursos del laboratorio del Grupo de Óptica y Láser de la Universidad del Cauca, se utilizó como superficie de detección y proyección un tablero acrílico que tiene una leve tendencia a curvarse hacia el centro del mismo. Sin embargo, esto no fue limitante para poner a funcionar este prototipo ya que la distancia máxima desde la superficie al dedo para detectar toques en el tablero es de 1.5 centímetros en las zonas de mayor curvatura y aproximadamente 0.5 centímetros para las de menor curvatura.

Como sucede en los montajes experimentales, las operaciones de procesamiento y análisis para todas las imágenes se pueden hacer en cada ciclo de muestreo invirtiendo alrededor de 200 ms para un computador de bajas capacidades en cuanto a su procesador y su memoria RAM, aunque se puede disminuir a menos de 50 ms en equipos actuales. Este es el método que más hace uso de los recursos del computador, puesto que debe adquirir y procesar un número dos veces mayor de imágenes que el que se requiere para los sistemas de 1 cámara. No obstante, el funcionamiento es lo suficientemente rápido para satisfacer los requerimientos de la detección de toques y la generación de órdenes al computador. Adicionalmente, para poder conseguir buenas detecciones, se debe controlar la luz del ambiente, de tal forma que se conserve el alto contraste entre el dedo

y el fondo negro, ya que no se trabaja con el perfil de línea directo de la imagen, sino con su derivada.

Cabe resaltar que un aspecto positivo del esquema propuesto es la actualización o adaptación dinámica del umbral, de forma similar al sistema de proyección de sombras.

Limitaciones en el número de toques

Al igual que en el sistema de proyección de sombras, en el método de triangulación se proyecta la posición angular de un dedo en el perfil de línea detectado por cada cámara. Como no se está observando directamente la zona de detección, se presentan las mismas ambigüedades que las mencionadas en el esquema anterior, excepto aquellas relacionadas con zonas muertas (figura 5.2, parte a), puesto que en este montaje se aprovecha todo el cuadrado destinado a ser zona útil. Utilizando las opciones para las cuales no se presentan ambigüedades, se tiene que el sistema solo responde cuando detecta e identifica uno o dos dedos.

5.1.5. Comparación entre los distintos métodos.

La tabla 5.1 muestra algunas características propias de cada sistema de detección de toques con el fin de comparar los cuatro esquemas elaborados. Todos los sistemas de visión utilizan un tamaño de captura de 640x480 píxeles. Los datos referentes al tiempo de procesamiento se midieron teniendo en cuenta las siguientes especificaciones del computador utilizado: 512 MB de memoria RAM, Procesador AMD Sempron de 1.76 GHz.

Tabla 5.1. Comparación entre las características de detección de toques de los cuatro esquemas elaborados en sus condiciones de funcionamiento.

Característica	ESQUEMA EXPERIMENTAL			
	FTIR	ILUMINACION DIFUSA	PROYECCION SOMBRAS	TRIANGULACIÓN.
Posee algoritmo(s) para calibración previa al uso	NO	NO	SI	NO
Tiempo aproximado de detección por ciclo (ms)	50	90	80	200
Actualización dinámica de fondo y umbral	NO	NO	SI	SI
Soporta varios toques o usuarios	SI	SI	SI	SI
Presencia de ambigüedades para resolver en la detección	NO	NO	SI	SI

5.2. EFICIENCIA Y ROBUSTEZ DE LOS ALGORITMOS DE PROCESAMIENTO

De forma general, se puede decir que se alcanzaron satisfactoriamente los objetivos planteados en cuanto al software requerido para interpretar los movimientos de los puntos de toque y convertirlos en órdenes al computador, ya que cada ciclo completo del funcionamiento de las plataformas ópticas multitáctiles diseñadas y elaboradas, representado en la figura 1.1, se realiza en tiempo real y de forma eficiente. A continuación se explican estos resultados para cada etapa de diseño de algoritmos:

5.2.1. Seguimiento

Para todas las plataformas construidas, incluso para el sistema con dos cámaras, el funcionamiento en tiempo real del ciclo propuesto permitió en gran parte simplificar la complejidad de las rutinas de seguimiento implementadas.

Para los dos montajes de vista directa de la superficie de detección por la cámara (FTIR e Iluminación Difusa) se determinó que únicamente se necesita utilizar el criterio de mínima distancia para la identificación de cada objeto de una imagen en la siguiente, puesto que para la tasa de adquisición de video utilizada (cerca a 30 cuadros por segundo), la longitud de la trayectoria recorrida por un dedo entre dos imágenes consecutivas siempre es menor que la separación entre ellos, mientras el usuario no realice movimientos bruscos. Incluso cuando los dos dedos se mueven muy juntos, si el algoritmo de detección los puede resolver como objetos diferentes, el desplazamiento de cada uno de los centros de masa casi siempre es menor que la separación entre los centros de masa en la misma imagen. La tendencia estadística de la utilización de los sistemas elaborados por parte de los usuarios muestra que es poco común e incómodo realizar acciones que requieran juntar mucho dos dedos, por tal razón se diseñaron aplicaciones en las cuales los objetos son de gran tamaño en comparación con la yema del dedo. Por las razones mencionadas, no fue necesario implementar validaciones adicionales o algoritmos más eficientes para estos dos sistemas.

En el caso de los dos sistemas que utilizan proyecciones, existe el problema de la presencia de ambigüedades, lo que hizo necesario implementar primero el criterio de la mínima distancia, con el fin de discriminar los puntos menos probables o aquellos que en teoría puedan aparecer fuera de las zonas de detección útiles. Posteriormente, se añadieron validaciones que utilizan acercamientos al método de las trayectorias más suaves, verificando que los vectores de velocidad en cada instante no cambien de

dirección bruscamente. Aunque este segundo criterio no se aplicó por completo, se pudo lograr un seguimiento básico y adecuado de hasta dos toques con la condición de introducir primero un dedo y después el otro.

5.2.2. Identificación de gestos y generación de eventos

El sistema se probó utilizando aplicaciones elaboradas y aplicaciones existentes para verificar el rendimiento del computador. Se determinó que aunque los algoritmos diseñados funcionan rápidamente, el consumo de recursos computacionales es más elevado entre mayor es la calidad de los gráficos de las aplicaciones clientes, es decir, aunque los algoritmos de detección, seguimiento, identificación de gestos y generación de eventos funcionan en tiempo real en la mayoría de los computadores probados, el desempeño de la aplicación gráfica en realidad depende de la capacidad que tenga el hardware para procesar video.

Iconos gestuales

Para la identificación de íconos gestuales se utilizaron algunos ejemplos de trayectorias conocidas de fácil repetición. Como se mencionó en el capítulo IV se utilizaron técnicas de Inteligencia Artificial basadas en Redes Neuronales Artificiales para entrenar al computador en el reconocimiento de las formas propuestas. Para esto, se tomaron algunas muestras para el entrenamiento elaboradas por un usuario y posteriormente el mismo usuario repitió sobre las pantallas los gestos repetidamente para verificar la confiabilidad del aprendizaje. La tabla 5.2 muestra las muestras utilizadas para el entrenamiento del perceptrón.

Tabla 5.2. Diversas trayectorias utilizadas para el aprendizaje de íconos gestuales con un dedo.

Espiral	Flecha	Letra S	Omega	Triángulo	Letra Z

Como se puede observar en la tabla 5.2, se han escogido gestos de tal forma que éstos se puedan reproducir sin importar la orientación o el tamaño. Por esta razón, el espacio de características a escoger para el entrenamiento de la máquina y la identificación debe constar de descriptores numéricos invariantes a las transformaciones de rotación y escala.

Las características escogidas para el entrenamiento de la red y la identificación de los gestos son las siguientes:

- Numero de puntos de inflexión: se hallan todos los puntos para los cuales se obtiene un cambio en el signo de la curvatura de la trayectoria vista como una función vectorial cuyas componentes x , y , dependen de la variable t , correspondiente al instante en que ocurre o aparece cada punto. Se espera que las espirales y omegas no posean puntos de inflexión, las letras s y las zetas posean un solo punto, las flechas y los triángulos un número indeterminado de puntos.
- Factor de forma: equivalente al área de la región que forma la trayectoria con sus extremos unidos por una recta, dividida sobre el cuadrado del perímetro que encierra dicha área. Para cada figura se tienen intervalos distintos de factor de forma, aunque en algunos casos, estos intervalos se intersectan en algunos valores.
- Relación ancho-altura: se obtiene la elipse inscrita a la trayectoria, es decir la mínima elipse que contiene todos los puntos en su interior. Posteriormente se hace la división entre la longitud de su eje mayor y la longitud de su eje menor.
- Parámetro especial de pendiente: se calcula el centro de masa equivalente para los puntos de la trayectoria. Luego se calculan las distancias entre el centro de masa y cada punto. Finalmente en la gráfica de distancia contra tiempo de ocurrencia

discreto (determinado por un índice entero) se calcula la pendiente de la recta que une el punto que presenta mayor distancia con el que presenta menor distancia. Para que esta pendiente sea invariante al tamaño de la figura se normaliza dividiendo entre la máxima distancia.

- Distancia entre los extremos: es la resta entre la distancia del punto final al centro de masa y la distancia del punto inicial al centro de masa. Se normaliza dividiendo el resultado entre la distancia máxima de todos los puntos. Esto sirve para diferenciar las figuras abiertas de las cerradas, específicamente para identificar los triángulos.
- Intervalo de curvatura normalizada: es la resta entre la máxima curvatura de la trayectoria menos la mínima curvatura y el resultado se divide entre el mayor valor absoluto de las curvaturas.
- Intervalo de ángulos: es la resta entre el mayor y el menor ángulo de giro de toda la trayectoria. Sirve para diferenciar las curvas con picos más agudos de las curvas suaves, por ejemplo, para diferenciar las letras s de las zetas.

La red neuronal diseñada cuenta con una capa de entrada igual al número de características extraídas de cada trayectoria, dos capas ocultas de cien neuronas cada una y una capa de salida de seis neuronas, equivalente al número de clases a identificar. Las neuronas cuentan con una función de activación sigmoidea determinada por la expresión: $f(x) = (1 - e^{-x}) / (1 + e^{-x})$.

Posteriormente al entrenamiento con las muestras de la tabla 5.2, se hizo una prueba con varias trayectorias de cada clase, utilizando como umbral de identificación el valor de 0.7, es decir, se necesita un 70% mínimo de correspondencia de un gesto con una de las 6

clases para poder considerarse como un gesto conocido, y se determinaron los resultados que se muestran en la tabla 5.3.

Tabla 5.3. Resultados obtenidos de la prueba del perceptrón utilizado con muestras distintas a las del entrenamiento.








Tipo de gesto	Número de muestras	Identificaciones correctas (%)	Identificaciones Incorrectas (%)	Gestos desconocidos (%)
Espirales	22	63.63	0.00	36.37
Flechas	20	90.00	5.00	5.00
Letras s	19	100.00	0.00	0.00
Omegas	27	96.30	0.00	3.70
Triángulos	16	75.00	12.50	12.50
Zetas	25	56.00	20.00	24.00
Total	129	79.84	6.20	13.96


Como se puede observar en los resultados de la tabla 5.3, se tiene que la red neuronal implementada cuenta con 79.84% de confiabilidad, aunque se pueden hacer muchas mejoras, escogiendo más y mejores características. De los porcentajes mostrados en la columna de gestos desconocidos, cabe mencionar que estos se obtuvieron de aquellos puntos máximos del vector de salida, de los cuales muchos de ellos corresponden al tipo correcto de trayectoria y los otros a tipos incorrectos. Para aquellos gestos que fueron identificados como desconocidos y el valor más probable era el correcto, se tiene que la causa principal de error fue el establecimiento de un alto umbral de aceptación. El valor de umbral se escogió de esta forma porque en muchos casos es preferible no enviar una

orden correcta de poca confiabilidad o seguridad al computador en lugar de enviar una orden equivocada.

A continuación se muestran algunos ejemplos de identificaciones incorrectas y de gestos desconocidos junto a las posibles causas de falla:

Tabla 5.4. Algunos fallos del perceptrón en el reconocimiento de íconos gestuales.

Forma	Resultado	Posible causa del error
	Desconocido	Umbral muy alto, al igual que sucedió con las demás espirales no reconocidas.
	Zeta	Similitud en el comportamiento temporal de la curvatura.
	Desconocido	Comportamiento de los ángulos y de gran parte de las distancias, similares a una omega.
	Desconocido	Comportamiento de la curvatura, ángulos, y factor de forma similar a una espiral.
	Espiral	Trazado de mala calidad, distanciamiento entre los extremos. Todas las detecciones incorrectas corresponden a falsas espirales.
	Letra s	Cambio suave de la curvatura en uno de los vértices.
	Triángulo	Similitud en el factor de forma, debida a la falta de longitud en uno de sus trazos.

	<p>Omega</p>	<p>Se conserva el signo de la curvatura (cero puntos de inflexión), debido a que en un vértice el giro se hace suave y en sentido contrario al habitual, aunque la curvatura es alta en el mismo.</p>
---	--------------	---

Gestos con dos toques

Se implementaron las operaciones matemáticas descritas en el Capítulo IV para las transformaciones de traslación, rotación y redimensionamiento. Los algoritmos se pusieron a prueba en una aplicación diseñada y la aplicación existente *Google Earth*. La aplicación diseñada consta de una imagen cualquiera, desplegada en pantalla completa. Cuando se detectan los dos toques se marcan sus centros de masa mediante círculos de color azul. Después del movimiento de los dedos se muestran nuevamente los puntos de toque con color azul y se superponen los puntos calculados por las transformaciones matemáticas mediante círculos de color rojo. El resultado fue de alta coincidencia para los puntos calculados respecto a los puntos de toque finales. En la mayoría de los casos, la desviación de los centros de masa es nula. En *Google Earth* se pudo observar los efectos de zoom y giro del globo terráqueo de forma adecuada, aunque con retardos mayores o menores dependiendo del computador utilizado.

Generación de eventos

Se hicieron pruebas simulando mediante software las órdenes provenientes del ratón y el teclado con el fin de manipular el sistema operativo. Se verificaron los eventos de clic, clic sostenido o doble clic en el explorador de carpetas de Windows, apertura de archivos y programas, entre otras acciones, como también se puso a prueba los comandos

generados mediante íconos gestuales (abrir un programa específico ante la detección de una forma conocida) y dos toques.

La generación de órdenes hacia el sistema operativo resulta ser adecuada solo para los comandos más simples como los de un solo toque, puesto que los complejos como zoom o rotación en aplicaciones gráficas tienen su propio mecanismo o combinación especial de teclas y botones que establece la compañía productora de dichos programas.

Como una alternativa para la solución del inconveniente de compatibilidad de los programas con las plataformas multitoque, actualmente cada vez más empresas desarrollan nuevas aplicaciones similares o mejores que las existentes mediante el uso de protocolos estándar de comunicación de datos como TUIO.

En este trabajo, se comprobó que los algoritmos para las plataformas multitáctiles desarrolladas se hicieron altamente compatibles, transmitiendo las coordenadas utilizando el protocolo TUIO, las cuales son recibidas en el mismo computador por aplicaciones clientes gratuitas desarrolladas por personas de otros lugares o países.

5.3. FUNCIONAMIENTO CONJUNTO DE HARDWARE Y SOFTWARE

5.3.1. Realimentación visual en la superficie táctil

Para el esquema de FTIR no se implementó despliegue gráfico, puesto que esto requiere un diseño más elaborado del sistema total que implica la búsqueda de un material para ser ubicado debajo de la superficie de acrílico y que sea lo suficientemente delgado (semitransparente), para permitir a la cámara observar las regiones de dispersión de luz,

como también lo suficientemente opaco (difusor) para que se pueda visualizar la proyección. Sin embargo, este es un sistema reproducido de forma similar a como se ha elaborado en otros lugares (siguiendo el montaje propuesto por Jeff Han), por lo cual existe suficiente información [8] que permite verificar que si es posible hacer realimentación visual.

Para el sistema de iluminación difusa se construyó un prototipo similar al descrito en el capítulo 3, con la diferencia que la lámpara está constituida con LED que únicamente emiten en infrarrojo, esto con el fin de no mezclar la luz amarilla de la lámpara incandescente con el despliegue y se proyectó la imagen de la pantalla mediante *Video Beam*. En la figura 5.3 se muestran los cambios mencionados. El sistema de detección mediante cámara infrarroja es el mismo, al igual que los algoritmos, excepto que la superficie de detección se reemplazó por vidrio y una lámina de papel, como se muestra en la figura 5.4.

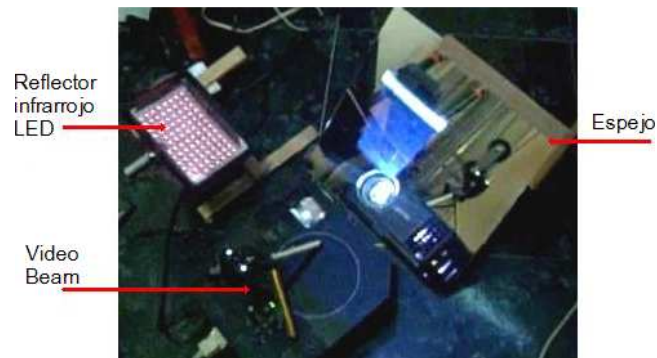


Figura 5.3. Sistema de iluminación y de proyección en el esquema de reflexión difusa.

Para el sistema de proyección de sombras, el despliegue se hace directamente sobre la pantalla del monitor, debido a que esta es la zona de detección de toques que incluye tanto la zona útil como las zonas muertas.



Figura 5.4. Superficie difusora siendo proyectada.

Finalmente, el sistema de triangulación se utilizó proyectando la salida del monitor directamente sobre el tablero haciendo coincidir el ancho de la proyección con el ancho del cuadrado de detección. El sistema de proyección se muestra en la figura 5.5.



Figura 5.5. Componentes del sistema de proyección en el esquema con dos cámaras. En la parte de arriba se muestran los componentes empleados (*Video Beam* y tablero) y abajo está el sistema en funcionamiento.

A diferencia de los otros esquemas, este tiene la ventaja de poderse utilizar tanto con proyecciones como con pantallas directamente, si se sustituye el tablero por una superficie transparente, por ejemplo vidrio, y se ubica detrás de ella el elemento de despliegue como un televisor pantalla plana. Esto permite crear una disposición que aprovecha el espacio de forma muy eficiente puesto que la profundidad total del sistema no sería mucho mayor al grosor del vidrio más el grosor del televisor.

5.3.2. Ejemplos

Para terminar este capítulo, a continuación se muestra una galería de imágenes de los sistemas elaborados funcionando en distintas aplicaciones:

Esquema de iluminación difusa

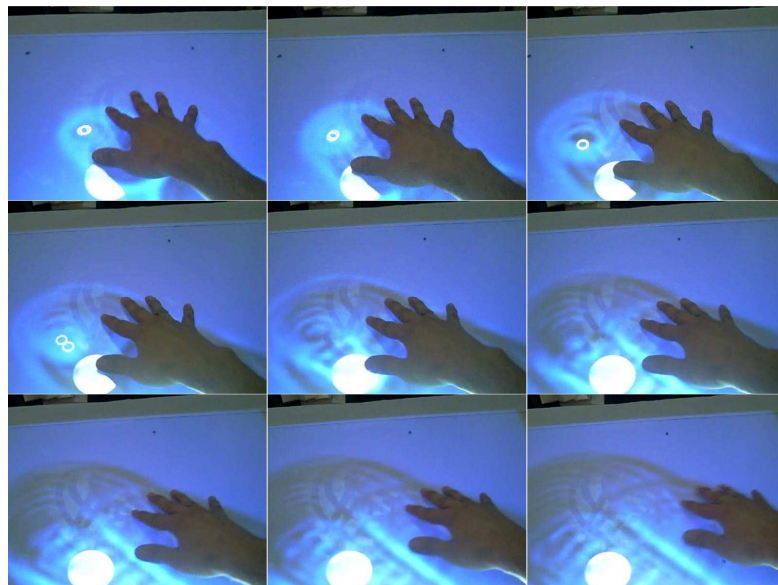


Figura 5.6. Simulación de ondas de agua sobre la superficie.

Esquema de triangulación con dos cámaras



Figura 5.7. Esquema de triangulación con dos cámaras siendo utilizado en Windows.

Esquema de proyección de sombras



Figura 5.8. Sombras proyectadas cuando se presenta uno, dos o más de dos toques.



Figura 5.9. Uno y dos usuarios utilizando el esquema de proyección de sombras.

CONCLUSIONES

De la realización de este trabajo se pueden concluir los siguientes aspectos:

- Se construyeron cuatro plataformas multitoque de tipo óptico, dos de ellas reproducidas a partir de experimentos mencionados en la literatura (esquemas de FTIR y de Iluminación Difusa) y las otras dos diseñadas a partir de conocimientos básicos sobre los fenómenos involucrados (basadas en proyecciones).
- Se diseñaron sistemas de visión por computador utilizando cámaras de video para cada plataforma, mediante los cuales se detectan los puntos de toque de forma eficiente y en tiempo real, logrando periodos de muestreo inferiores a 200 milisegundos entre imágenes consecutivas.
- Se dio un soporte apropiado a los procedimientos empleados, teniendo en cuenta los conceptos físicos y matemáticos involucrados en el diseño de los algoritmos de detección para las cuatro plataformas. Esto con el fin de evitar la mayor cantidad posible de decisiones de programación basados en métodos de ensayo y error.
- Para la plataforma de proyección de sombras y la de triangulación con dos cámaras, se elaboraron algoritmos con alto grado de robustez gracias a la implementación de rutinas de actualización automática de las condiciones de iluminación y los umbrales para la detección.
- Se desarrollaron algoritmos de seguimiento eficientes para la identificación de cualquier número de toques utilizando el criterio de la mínima distancia para las

plataformas en las cuales la cámara ve directamente la superficie de detección, y lo suficientemente adecuados para identificar hasta dos toques en los sistemas que utilizan proyecciones con ayuda de algunas condiciones del método de suavizado de trayectorias.

- Se diseñaron algoritmos de reconocimiento de gestos con uno y con dos toques. Para un toque se aplicaron conocimientos de aprendizaje de basados en redes neuronales artificiales, mediante los cuales se entrenó el computador para reconocer formas básicas constituidas de una sola trayectoria. Para dos toques se desarrollaron algoritmos que interpreten los movimientos de los dedos y los conviertan en transformaciones geométricas de traslación, rotación y redimensionamiento de objetos en aplicaciones gráficas.
- Se implementaron rutinas de generación de órdenes al computador de dos tipos: el primero de emulación de eventos como los producidos por el ratón o el teclado, los cuales son interpretados de forma diferente por cada programa, y el segundo tipo de transmisión de las coordenadas de toque a diversas aplicaciones cliente mediante protocolo TUIO para que éstas generen sus propias acciones dependiendo el contexto en el que funcionan.
- Para 3 de los sistemas elaborados se implementó realimentación visual: para iluminación difusa se utilizó proyección, para proyección de sombras se utilizó directamente el monitor del computador y para el sistema de triangulación se utilizó proyección sobre el tablero.
- Si se toma cada punto de toque como un generador independiente de eventos, cualquiera de estas plataformas se puede considerar multiusuario, o de dos usuarios en el caso de los sistemas de proyección de sombras y triangulación.

TRABAJOS FUTUROS

Se puede explorar otros métodos de detección de coordenadas para plataformas multitoque, tales como planos de luz láser, sistemas de múltiples cámaras, iluminación difusa por transmisión, entre otras; como también mejorar los métodos de detección empleados: adecuación de una lámina de silicona y una superficie difusora para el esquema FTIR, reconocimiento de objetos, marcadores y la disposición de los dedos de una mano para el esquema de iluminación difusa por reflexión, eliminación de las zonas muertas en el esquema de proyección de sombras y optimización de las operaciones de procesamiento y detección en el esquema de las dos cámaras.

Se propone aumentar la eficiencia de las rutinas de seguimiento mediante la implementación completa del criterio de las trayectorias más suaves, sin dejar de lado el criterio de la mínima distancia.

También es posible estudiar gestos multitáctiles con más de dos dedos. Sería muy útil la identificación de grupos de puntos de toque para diferenciar los dedos de una mano de los dedos de otra o de un usuario diferente. Adicionalmente, en sistemas como los que utilizan más de una cámara se pueden identificar gestos en tres dimensiones determinando los movimientos de la persona tanto en el plano de detección como fuera de él, con el fin de crear interfaces de usuario más interactivas.

Como un campo adicional se puede trabajar en diseño de aplicaciones con algún fin específico: investigación, enseñanza, publicidad, entretenimiento, entre otros.

BIBLIOGRAFÍA

[1] Interfaz gráfica de usuario. Wikipedia, la enciclopedia libre. Página de internet: http://es.wikipedia.org/wiki/Interfaz_gráfica_de_usuario. Fecha de acceso: 17 de Enero de 2010.

[2] Proyectos de investigación y desarrollo relacionados con la accesibilidad. Página de internet: <http://www.microsoft.com/spain/accesibilidad/microsoft/research.aspx>. Fecha de acceso: 17 de Enero de 2010. Microsoft Corporation.

[3] Página principal de la compañía Apple. <http://www.apple.com>. Fecha de acceso: 17 de Enero de 2010.

[4] Planet Analog. The art of capacitive touch Sensing. Página de internet: <http://www.planetanalog.com/features/showArticle.jhtml>. Fecha de acceso: 31 de mayo de 2010.

[5] Interacción Humano-Computadora (IS-438). Universidad de las Américas-Puebla. México. <http://ict.udlap.mx/people/ingrid/Clases/IS438/IHC.html>. Fecha de acceso: 17 de Enero de 2010.

[6] UPIICSA. Artículos de los números 41 y 42, Mayo-Diciembre 2006. Página de internet: <http://www.revistaupiicsa.20m.com/Emilia/RevMayDic06/popmaydic06.htm>. Fecha de acceso: 17 de Enero de 2010.

[7] Interacción Humano-Computadora (IS-438). Aspectos tecnológicos. Página: <http://ict.udlap.mx/people/ingrid/Clases/IS438/aspectos-tecnologicos.html>. Fecha de acceso: 17 de Enero de 2010.

[8] NUI Group Authors. Multitouch Technologies. First Edition. Mayo de 2.009.

[9] Multi-touch. From Wikipedia, the free encyclopedia. Página de internet: <http://en.wikipedia.org/w/index.php?title=Multi-touch&action=edit§ion=2>. Fecha de acceso: 05 de Mayo de 2010.

[10] Biografía de Jefferson Hann. Página: http://en.wikipedia.org/wiki/Jefferson_Hann. Fecha de acceso 05 de Mayo de 2010.

[11] Microsoft Surface Data Sheet. Microsoft Corporation. 2008. www.surface.com. Fecha de Acceso: 05 de Mayo de 2010.

[12] Tech Note: Device-Free Interaction Spaces. Página de internet: <http://www.cs.uit.no/~daniels/interaction-spaces-3dui.pdf>. Interaction Spaces: A camera-based system for scalable multi-touch 2D and 3D interaction on wall-sized, high-resolution displays. Fecha de acceso: 05 de Mayo de 2010.

[13] Multi-Touch G³ Basic Specifications. Página: <http://multi-touch-screen.com>. Fecha de acceso: 05 de Mayo de 2010.

[14] García M., Ginés. Notas de clase sobre Procesamiento Audiovisual. Unidad 1. Departamento de informática y sistemas. Universidad de Murcia. <http://dis.um.es/profesores/ginesgm>. Fecha de acceso: 27 de Abril de 2010.

- [15] Gonzales, R. Woods, E. Digital Image Processing. Prentice Hall. 2002.
- [16] Bradski, G. Kaebler, A. Learning OpenCV. Computer Vision with the OpenCV library. First Edition. O'Reilly Media, Inc. 2008.
- [17] García M, Ginés. Tesis Doctoral: Procesamiento de Caras Humanas Mediante Integrales Proyectivas. Universidad de Murcia. Facultad de Informática. Departamento de Informática y Sistemas. 2007.
- [18] OpenCV Documentation. May 7, 2009.
- [19] Hecht, E. Optics. International edition. Fourth edition. Adelphi University. 2002.
- [20] Shapiro, L. Stockman, G. Computer Vision. Washington University and Department of Computer Science of Michigan State University. March 2000.
- [21] Buchelly I, F. Bravo M, D. Pencue F, L. Ferrín B, D. Sistema de Seguimiento Bidimensional Mediante Técnicas de Control Digital y Visión Computacional. Revista Colombiana de Física. Artículo aceptado para publicación. 2009.
- [22] Kaltenbrunner, M. Bovermann, T. Bencina, R. Costanza, E. TUIO: A Protocol for Table-Top Tangible User Interfaces.