

TESIS
II 2005
F43
v. 8.



UNIVERSIDAD CATÓLICA ANDRÉS BELLO
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA INFORMÁTICA

Diseño y Construcción de un set de Realidad Virtual

TRABAJO ESPECIAL DE GRADO

presentado ante la

UNIVERSIDAD CATÓLICA ANDRÉS BELLO

como parte de los requisitos para optar al título de

INGENIERO EN INFORMÁTICA

REALIZADO POR Dennis Federico

PROFESOR GUIA Carlos Magurno

FECHA 20 de Noviembre del 2005



UNIVERSIDAD CATÓLICA ANDRÉS BELLO
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA INFORMÁTICA

Diseño y Construcción de un set de Realidad Virtual

Este Jurado; una vez realizado el examen del presente trabajo ha evaluado los desarrolladores con el resultado: 20 puntos - Mención Honorífica y Mención Publicación

JURADO EXAMINADOR

Firma: 
Nombre: ASSAF YAMIN

Firma: 
Nombre: Ramón Poma G.

Firma: 
Nombre: CARLO MAGURNO

REALIZADO POR

Dennis Federico

PROFESOR GUIA

Carlos Magurno

FECHA

20 de Noviembre del 2005

Prólogo.

Este trabajo de grado está inspirado en el deseo de obtener una interacción más natural entre los usuarios y las computadoras; así como también en la curiosidad de sentirse inmerso en las fantasías que se pueden simular con un computador.

Tiene como objetivo estudiar un conjunto de posibles medios de interacción con el computador, basados en el movimiento del cuerpo humano, específicamente el de las manos y la cabeza.

La razón que llevó a adoptar este tema es la posibilidad que brinda de romper con el mito de que estos dispositivos no están al alcance del usuario común y los desarrolladores de software.

5.2.7 Imágenes de vista **Tabla de contenido.**

Prólogo.....	i
Agradecimientos.....	ii
Dedicatoria.....	iii
Sinopsis.....	1
Introducción.....	3
1. Planteamiento del Problema.....	5
2. Objetivos.....	7
2.1. Objetivo General.....	7
2.2. Objetivos Específicos.....	7
3. Limitaciones y Alcances.....	9
4. Justificación.....	12
5. Marco Teórico.....	14
5.1. Efecto estereoscópico de la vista.....	14
5.2. Técnicas para lograr un efecto estereoscópico.....	17
5.2.1. Imágenes de visión libre paralela.....	17

5.2.2. Imágenes de visión libre cruzada.....	18
5.2.3. Anaglifos.....	18
5.2.4. Polarización.....	18
5.2.5. Imágenes Alternativas o Entrelazadas.....	19
5.3. Tipos de <i>Head Mounted Displays</i> (HMD's).....	19
5.3.1. HMD's Monoculares.....	20
5.3.2. HMD's Binoculares o Estéreo.....	20
5.3.3. <i>Shutter Glasses</i>	20
5.4. Tipos de Guantes Virtuales según su funcionamiento.....	20
5.4.1. Guantes Ópticos (<i>DataGlove</i>):.....	21
5.4.2. Guantes de Tinta Conductiva (<i>PowerGlove</i>):.....	21
5.4.3. Guantes de Tensión (<i>CyberGlove</i>):.....	22
5.4.4. Guante Electro-Mecánico (<i>EXOS Dextrous Hand Máster</i>):.....	23
5.5. Métodos de respuesta al usuario para los guantes virtuales.....	23
5.5.1. <i>Force Feedback</i> :.....	23
5.5.2. <i>Tactile Feedback</i> :.....	24

5.6. Tecnologías de rastreo de movimiento y posicionamiento.....	25
5.6.1. Rastreadores Mecánicos.....	25
5.6.2. Rastreadores Inerciales.....	27
5.6.2.1. Funcionamiento de los Giroscopios.....	29
5.6.2.2. Funcionamiento de los Acelerómetros.	29
5.6.3. Rastreadores Magnéticos.....	30
5.6.4. Rastreadores Ultrasónicos.	31
5.6.5. Rastreadores Ópticos.....	32
5.6.6. Rastreadores por Ondas de Radio y Microondas.....	33
5.6.7. Rastreadores Híbridos.....	35
5.6.8. Comparación entre los diferentes tipos de rastreadores.	35
5.7. Funcionamiento de un Convertidor Analógico Digital (ADC).....	36
5.8. Puerto USB (Universal Serial Bus).....	39
6. Metodología.....	45
7. Desarrollo.	47
7.1. Convertidor Analógico/Digital e Interfaz USB.....	47

7.1.1. Microcontrolador Pic18f4550.....	49
12.1.2. Configuración del ADC.....	51
7.1.3. Implementación y configuración del módulo USB.....	53
7.2. Construcción del guante virtual.....	56
7.3. Construcción del dispositivo de rastreo de posición y orientación para el guante.....	58
7.3.1. <i>Firmware</i> del Guante virtual.....	65
7.4. Construcción del dispositivo de rastreo de posición y orientación para la cabeza.....	65
7.5 Diseño y desarrollo del software de control y API.....	67
8. Resultados.....	70
9. Conclusiones y Recomendaciones.....	72
Índice de Tablas y Figuras.....	74
Bibliografía.....	76

Sinopsis.

El objetivo de este trabajo es construir un set de realidad virtual compuesto por un guante y un casco que permitan la interacción con el computador y que se puedan elaborar réplicas fácilmente. La función principal de estos dispositivos es proveer al computador información sobre la flexión de los dedos de la mano, la posición y orientación de la misma y de la cabeza del usuario.

Para detectar la flexión de los dedos de la mano se utilizó un arreglo compuesto de un diodo luminoso o diodo LED y una foto resistencia colocados en los extremos de un tubo flexible cubierto con un material semi reflectivo para cada uno de los dedos, de manera que la intensidad de luz emitida por el diodo que llega a la foto resistencia varía con la flexión del tubo.

El dispositivo de rastreo de la cabeza esta conformado por un arreglo de resistencias variables o potenciómetros sobre una estructura de madera que semeja un brazo mecánico, los potenciómetros ubicados en las uniones permiten calcular los ángulos según la variación en las resistencias.

Para el rastreo de la posición y orientación de la mano se utilizó un arreglo de tres giroscopios electromecánicos de *Analog Devices* modelo ADXR300 y un acelerómetro tri-axial de *Kionix* modelo KXM52. Estos dispositivos permiten calcular los movimientos de rotación de la mano, así como sus movimientos a lo largo de los ejes de coordenadas.

Los voltajes emitidos por los componentes deben ser convertidos en señales digitales para ser transmitidas y procesadas por el computador; para ello, el elemento clave y primer paso del desarrollo fue el diseño e implementación de un circuito que convirtiera las señales analógicas en digitales y las transmitiera al computador. Esto se logró utilizando un microprocesador Pic18f4550 de Microchip que tiene entre sus capacidades un convertidor Analógico Digital de 10-bits y módulo USB (*Universal Serial Bus*) para comunicarse con el computador.

Introducción.

En la ciencia de la computación, la realidad virtual esta definida como la simulación por computadora de un sistema real o imaginario que permita al usuario realizar operaciones, interactuar en el sistema simulado y mostrar el efecto en tiempo real.

Los sistemas de realidad virtual (VR) permiten al usuario moverse e interactuar en un ambiente simulado por computadora, utilizando diversos dispositivos a través de los cuales el usuario puede sentir y manipular los objetos virtuales como en el mundo real. Los mundos simulados son creados a partir de modelos matemáticos y programas de computadoras. La realidad virtual se diferencia de otras simulaciones, ya que estas requieren de dispositivos de interfaz que permitan una interacción más "natural" entre el usuario y el ambiente simulado, sistemas más complejos no solo permiten respuesta visual y auditiva por parte del sistema, también tienen la capacidad adicional de transmitir los movimientos del usuario al mundo simulado como parte de la interacción.

La inmersión del usuario en el mundo virtual es el factor clave en la realidad virtual. La percepción del mundo virtual esta conformada por la forma en que se estimulan los sentidos del usuario, la combinación de sentidos estimulados y la interacción con el mundo virtual; esto es directamente proporcional al grado de inmersión. Los dispositivos a desarrollar en este trabajo proponen una interacción cinética con el mundo virtual a través de una respuesta visual.

En la primera sección, **Planteamiento del problema**, se presenta los dispositivos de realidad virtual estudiados y construidos en este proyecto.

En la segunda sección, **Objetivos**, se enumeran los objetivos que se buscan alcanzar en el desarrollo.

En la tercera sección, **Limitaciones y alcances**, se plantean los alcances de este proyecto y algunos factores que pudieron limitar su desarrollo.

En la cuarta sección, **Justificación**, se consideran algunos factores causantes de la situación actual de los dispositivos de realidad virtual con el objetivo de justificar la elaboración de este proyecto.

En la quinta sección, **Marco teórico**, se presenta un resumen de la teoría que fundamenta este proyecto.

En la sexta sección, **Metodología**, se explica la metodología aplicada para desarrollar este proyecto y las razones por las cuales fue seleccionada.

En la séptima sección, **Desarrollo**, se plantea cada una de las fases de desarrollo del proyecto y se explica como se desarrollo cada dispositivo.

En la octava sección, **Resultados**, se enumeran los objetivos alcanzados y los resultados obtenidos.

En la novena sección, **Conclusiones y recomendaciones**, contiene algunas recomendaciones para desarrollos futuros y las conclusiones obtenidas.

1. Planteamiento del Problema.

A pesar de sus aplicaciones, el desarrollo de dispositivos de realidad virtual estuvo estancado para el usuario común desde sus orígenes entre la década de los '80 y '90, debido a su alto costo y la baja velocidad de procesamiento de las computadoras de la época que eran insuficientes y limitadas. Actualmente, el costo de los mismos dispositivos sigue siendo muy elevado.

Los dispositivos de realidad virtual más comunes en el mercado y disponibles para el usuario promedio según su precio, son un conjunto de *Head Mounted Displays* (HMD's) y lentes de realidad virtual para ver películas. Si bien pueden adaptarse para ser utilizados con un computador, en su mayoría estos dispositivos deben estar conectados a un convertidor que interpreta las señales de video de películas especialmente diseñadas para ellos.

Otros dispositivos que ofrezcan interacción por parte del usuario, como es el caso de los dispositivos de rastreo de posición y movimiento para el cuerpo humano, son más costosos y son utilizados casi exclusivamente en áreas específicas como la captura de movimientos del usuario para desarrollar modelos por computadora para el cine u otras simulaciones, para investigaciones científicas, entrenamiento militar, etc.

Los principios de funcionamiento de la mayoría de los dispositivos de realidad virtual son conocidos y los componentes necesarios para su implementación y construcción cada día cuestan menos y/o son mejores, como es el caso de las

pantallas de cristal liquido (LCD's), uno de los componentes esenciales de los HMD's, lo que hace difícil entender el alto costo de dichos dispositivos en el mercado.

Dispositivos y Construcción de Realidad Virtual

Sin embargo, durante la realización de este trabajo, *ESSENTIAL REALITY* redujo el precio de su guante de realidad virtual P5 (de 250\$ a 70\$) y colocó a disposición del público un API (*Application Programming Interface / Interfaz para programar aplicaciones*) que permite desarrollar programas para hacer uso del dispositivo.

2. Objetivos. Desarrollo del software de simulación para utilizar en los dispositivos

2.1. Objetivo General.

Desarrollo de un programa de simulación
Diseñar y Construir un set de realidad virtual.

2.2. Objetivos Específicos.

- Construcción de lentes 3D o *Head Mounted Display* (HMD)
 - Estudiar el funcionamiento del efecto estereoscópico de la vista.
 - Estudiar los diferentes tipos de HMD y las técnicas alternativas para lograr el efecto estereoscópico.
- Construcción del Guante Virtual.
 - Estudiar los diferentes tipos de guantes virtuales según su funcionamiento.
 - Estudiar los métodos de respuesta al usuario para el guante virtual.
- Adquisición o construcción de los dispositivos de rastreo de posición y orientación para la cabeza y las manos
 - Estudiar las tecnologías de rastreo y posicionamiento más comunes.
- Adquisición o construcción de un convertidor Analógico Digital que se adapte a los dispositivos del set y sirva de interfaz con el computador.

3. • Desarrollo del software de control y calibración para cada uno de los dispositivos.

Este trabajo especial de grado busca alcanzar el diseño y construcción

• Desarrollo de un conjunto de API's que permitan el desarrollo de software para utilizar el set de realidad virtual o sus dispositivos en forma individual.

control y calibración de dispositivos de realidad virtual.

Por último, se pretende desarrollar un

API que permita implementar

forma individual o en conjunto.

La falta de un estándar

dispositivos de realidad virtual

de estos objetivos. En primer lugar,

especificaciones generales para

la computadora. Para ello, se

para dispositivos de interfaz, como

del consorcio USB, se

a cada uno de los dispositivos.

La adquisición de datos

de los dispositivos de realidad

giroscopio, acelerómetro y

nacional de estándares

3. Limitaciones y Alcances.

Este trabajo especial de grado pretende alcanzar el diseño y construcción de un *set* (conjunto de dispositivos, hardware y software) de realidad virtual (VR). Para tal fin, es necesario estudiar y fabricar los diferentes componentes que constituirán el *set*, así como también desarrollar los programas de acceso, control y calibración de dichos dispositivos.

Por último, se pretende hacer uso de dichos controladores para desarrollar un API que permita implementar programas que hagan uso de los dispositivos en forma individual o en conjunto.

La falta de un estándar sobre las especificaciones de entrada y salida de los dispositivos de realidad virtual, constituye un factor limitante para el desarrollo de estos objetivos. Es posible que los dispositivos actuales no utilicen especificaciones genéricas que describan como debe ser la comunicación con la computadora. Para superar esta limitante, se utilizarán las especificaciones para dispositivos de interfaz humana (HID) del puerto USB, definidas en HID 1.1 del consorcio USB que además implementa un conjunto de etiquetas aplicables a cada uno de los datos, con la finalidad de describir su uso u origen.

La adquisición de los componentes especiales utilizados para la construcción de los dispositivos, como son los microcontroladores, los acelerómetros y los giroscopios, representó otra limitante, ya que no se consiguen en el mercado nacional, por lo cual fue necesario importarlos al país.

En cuanto a los acelerómetros y los giroscopios, por ser dispositivos relativamente innovadores en su arquitectura y estar disponibles recientemente al público general, la documentación existente en cuanto a sus aplicaciones es todavía limitada a ciertas aplicaciones experimentales conducidas por universidades extranjeras y otras iniciativas independientes. Entre dichas aplicaciones, podemos mencionar la detección de golpes, caídas y aceleraciones repentinas, datos utilizados en la industria automotriz para la activación de sistemas *airbag*, el bloqueo de los cabezales de disco duros para equipos portátiles y otras aplicaciones industriales.

El protocolo USB (*Universal Serial Bus*), aunque ampliamente adoptado desde hace varios años por los fabricantes de hardware, representó un reto al momento de desarrollar los controladores y el API (*Application Programming Interface / Interfaz para programar aplicaciones*) de los dispositivos. En primer lugar, la documentación, si bien es muy extensa, no ofrece información concreta para el desarrollo de dispositivos. Por otro lado, la comunicación a través del concentrador USB del computador, depende del sistema operativo, de manera que, para dispositivos específicos, los controladores deben ser desarrollados según el sistema operativo a utilizar, con excepción de las clases USB definidas en las especificaciones, entre ellas los dispositivos HID, que están implementadas en los sistemas operativos más comunes (*Windows* y *Linux*) en la mayoría de sus versiones. De igual forma, fue necesario implementar un controlador intermedio entre JAVA (el lenguaje de programación utilizado) y el

sistema operativo, ya que este no ofrece librerías para comunicación con el puerto USB; este *middleware* se desarrolló como un DLL (Librería de Enlace Dinámico) estático utilizando C y JNI (*Java Native Interface*).

4. Justificación.

Es por todo lo expuesto anteriormente, que este trabajo especial de grado plantea diseñar y construir un *set* de realidad virtual accesible en precio, basado en arquitecturas de computadoras actuales que facilite el acceso a esta tecnología para programadores y usuarios.

Es indudable la cantidad de aplicaciones que se le puede dar a la realidad virtual y sus dispositivos, pero estos últimos, parecen estar restringidos a científicos y empresas que ofrecen servicios de realidad virtual para entretenimiento, modelación de productos, asistencia para animaciones, etc. Inclusive en el área de entretenimiento doméstico, la realidad virtual se limita casi en su totalidad a aplicaciones netamente visuales con poca a ninguna interacción por parte del usuario.

A mediados de la década de los '80, los dispositivos de realidad virtual disponibles para el usuario común eran solo para consolas de video juegos. A pesar de esto, los entusiastas no se quedaron de brazos cruzados, así que idearon formás para utilizar estos dispositivos en sus computadores, como es el caso del *Nintendo Power Glove*. A través de foros y sitios *Web* intercambiaban ideas y experiencias, inclusive escribieron sus propios programas para experimentar con sus dispositivos, como el *REND386*, una herramienta para crear mundos virtuales con polígonos simples e interactuar usando guantes virtuales. Comenzando la década de los '90, se dieron cuenta de que estaban limitados por las capacidades de video y procesamiento de sus

computadoras, perdiendo así el interés y dejando relegadas estas tecnologías a los departamentos de investigaciones de empresas y universidades, quienes podían costear la adquisición de supercomputadoras y estaciones de trabajo gráfico.

Diez años más tarde, con procesadores 100 veces más potentes y tarjetas gráficas aceleradoras con procesadores extras para cómputos gráficos a la disposición de cualquier usuario, la actividad en el área sigue siendo casi nula. Así lo comentan los miembros del equipo de desarrollo de *DirectInput* para *DirectX 8.0* de *Microsoft*, en un foro abierto con la comunidad cibernauta al responder si darían soporte a dispositivos de realidad virtual, con estas palabras... *"We frequently look at these types of devices. Right now, the price points for the most compelling devices are too high for the consumer market. We will if they are becoming common."* (Hoy en día los precios son muy altos para los consumidores, daremos soporte sí estos dispositivos se hacen comunes).

¿Por qué sucede así?, nuevamente muchas personas se quejan del costo de los dispositivos, otras de los pocos programas disponibles para utilizarlos. ¿Por qué hay pocos programas?, según la revista *IEEE Computer Graphics and Applications* 2001, la industria de la realidad virtual no ha establecido un estándar para los diferentes aspectos involucrados en el diseño de los dispositivos.

5. Marco Teórico.

5.1. Efecto estereoscópico de la vista.

De manera natural, el mecanismo de visión humana es estéreo, es decir, somos capaces de apreciar, a través de la visión binocular, las diferentes distancias y volúmenes en el entorno que nos rodea. Los ojos, debido a su separación, obtienen dos imágenes con pequeñas diferencias entre ellas, lo que se denomina **disparidad**. El cerebro procesa las diferencias entre ambas imágenes y las interpreta de forma que se percibe la sensación de profundidad, lejanía o cercanía de los objetos. Este proceso se denomina **estereopsis**. [Gonzales, 2004].

En la estereopsis intervienen diversos mecanismos. Cuando se observan objetos muy lejanos, los ejes ópticos de los ojos son paralelos (Fig.1). Cuando se observa un objeto cercano, nuestros ojos giran para que los ejes ópticos estén alineados sobre él, es decir, **convergen**. A su vez se produce la **acomodación** o enfoque para ver nítidamente el objeto. Este proceso conjunto se llama **fusión**. No todas las personas en el mundo tienen la misma capacidad de fusionar un par de imágenes en una sola imagen tridimensional. Alrededor de un 5% de la población tiene problemas de fusión. La agudeza estereoscópica es la capacidad de discernir, mediante la estereopsis, detalles situados en planos diferentes y a una distancia mínima. Existe una distancia límite a partir de la cual el hombre es capaz de apreciar la separación de planos, dicha distancia varía de una persona a otra. Así, la distancia límite a

la que dejamos de percibir la sensación estereoscópica puede variar desde unos 60 metros hasta cientos de metros [Gonzales, 2004].

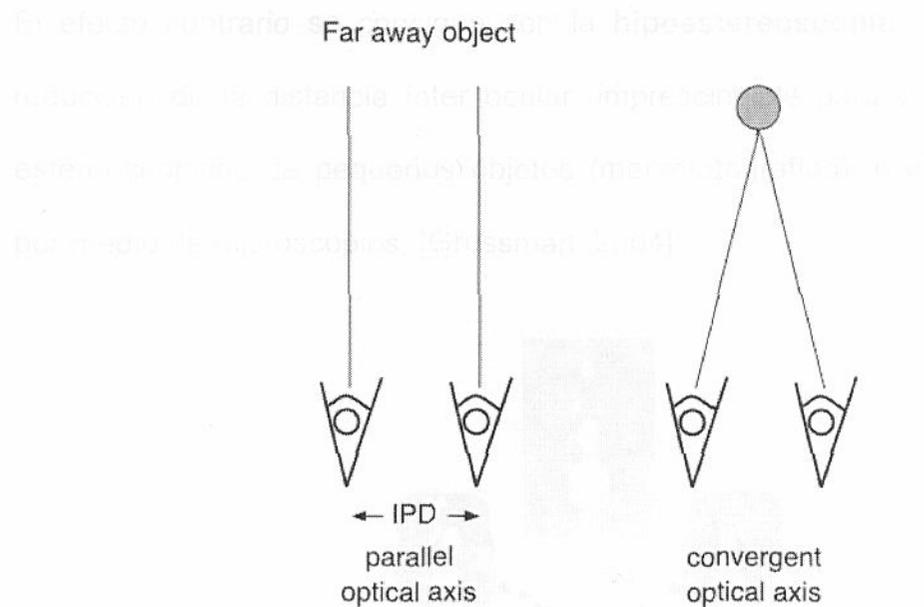


Fig. 1. Convergencia de los ejes ópticos de la vista.

Fuente: <http://www.users.red3i.es/~stereoweb/index.htm>

Un factor que interviene directamente en esta capacidad es la separación ínter ocular. A mayor separación entre los ojos, mayor es la distancia a la que apreciamos el efecto de relieve. Esto se aplica, por ejemplo, a los binoculares, en los cuales, mediante lentes, se consigue una separación ínter ocular efectiva mayor que la normal, consiguiendo apreciar en relieve objetos distantes que en condiciones normales no seríamos capaces de separar del entorno. También se aplica en la fotografía aérea, en la que se obtienen pares estereoscópicos con separaciones de cientos de metros, siendo posible apreciar claramente el relieve del terreno, lo que con la visión normal y desde gran altura sería imposible. El efecto obtenido con una separación ínter ocular mayor que la

habitual consiste en que los objetos parecen más pequeños de lo normal (liliputismo), y la técnica se denomina **hiperestereoscopia**.

El efecto contrario se consigue con la **hipoestereoscopia**, es decir, con la reducción de la distancia ínter ocular, imprescindible para obtener imágenes estereoscópicas de pequeños objetos (macrofotografías), o incluso obtenidas por medio de microscopios. [Grossman, 2004]

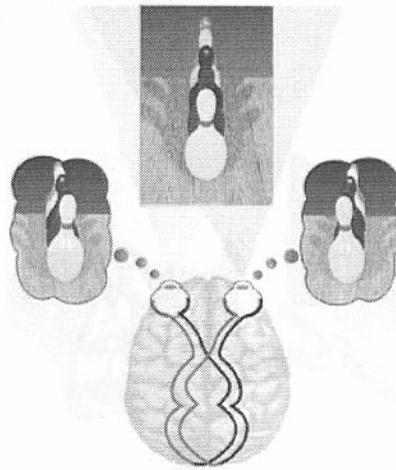


Fig. 2. Fusión de imágenes en el cerebro (estereoscopia).

Fuente: <http://www.vision3d.com/>

La percepción estereoscópica es comunmente confundida con los efectos 3D generados por computadora, lo que no es completamente correcto, ya que, la profundidad (tercera dimensión) en estos casos es solo “simulada” mediante sombras. Para simular la visión estereoscópica es necesario un mínimo de dos imágenes que representen la imagen captada por cada uno de los ojos.

5.2. Técnicas para lograr un efecto estereoscópico.

Al parecer, Euclides y el genial Leonardo DaVinci observaron y estudiaron el fenómeno de la visión binocular, siendo considerados los pioneros en este tema. Curiosamente la estereoscopia precedió a la fotografía. Fue un físico escocés, Sir Charles Wheatstone, quién en Junio de 1838 describió por primera vez y con cierto rigor el fenómeno de la visión tridimensional y construyó luego un aparato con el que se podían apreciar en relieve dibujos geométricos: el Estereoscopio (Fig. 3).

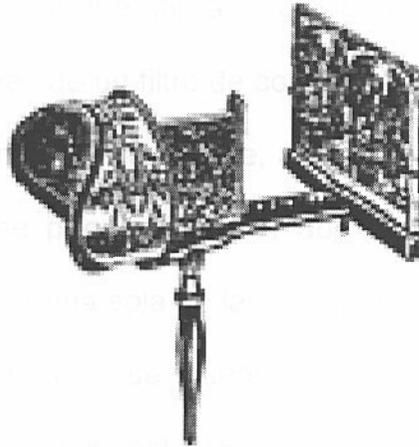


Fig. 3. Estereoscopio de Sir Charles Wheatstone.

Fuente: <http://www.users.red3i.es/~stereoweb/historia.htm>

5.2.1. Imágenes de visión libre paralela.

Comúnmente llamados estereogramás; en ellas se conjugan las imágenes destinadas para cada ojo en una sola, a simple vista, la imagen resulta plana, pero al enfocar la vista detrás de la imagen, con los ejes ópticos en paralelo o

mirando al infinito, se puede observar una forma o relieve que sale del plano.

5.2.2. Imágenes de visión libre cruzada.

También llamadas fotografías estereoscópicas, son precisamente dos imágenes independientes que se observan cruzando los ejes ópticos de los ojos. El par estéreo se presenta invertido, es decir, la imagen derecha está situada a la izquierda y viceversa.

5.2.3. Anaglifos.

Se denomina así a la técnica que utiliza filtros de colores para separar las dos imágenes. Mirando a través de un filtro de color rojo, los colores verde o azul se ven como negro y utilizando un filtro verde, azul o cian, el rojo parece negro. A partir de este principio se pueden mezclar dos imágenes y observarlas con filtros de color para separar una sola de las dos imágenes, finalmente al colocar un filtro distinto en cada ojo, se pueden proporcionar las dos imágenes necesarias para obtener el efecto estereoscopio.

5.2.4. Polarización.

Esta técnica se basa en un fenómeno físico llamado polarización de la luz. La luz se transmite por ondas que pueden ser verticales u horizontales, utilizando lentes o filtros capaces de eliminar las ondas en una dirección o polaridad. Se pueden proyectar dos imágenes, una polarizada en un sentido y la otra a 90 grados y utilizar dos filtros para que cada ojo vea una imagen distinta.

5.2.5. Imágenes Alternativas o Entrelazadas.

Con este sistema se presentan en secuencia y alternativamente las imágenes izquierda y derecha, en sincronía con unos lentes obturadores de cristal líquido (denominadas **LCS**, *Liquid Crystal Shutter glasses* o **LCD**, *Liquid Crystal Display glasses*), de forma que cada ojo ve solamente su imagen correspondiente. A una frecuencia de obturación elevada, el parpadeo es imperceptible.

5.3. Tipos de *Head Mounted Displays* (HMD's).

Los HMD's, (Fig. 4) como su nombre en inglés lo indica, son sistemas de visión colocados sobre la cabeza del usuario. Estos dispositivos son considerados los más importantes para lograr la inmersión en los entornos de realidad virtual, ya que, la vista es el sentido que más información nos brinda sobre el mundo que nos rodea.

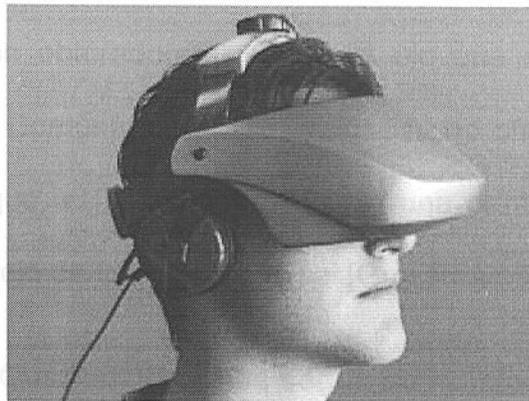


Fig. 4. Head Mounted Display.

Fuente: <http://www.stereo3d.com/hmd.htm>

5.3.1. HMD's Monoculares.

El más simple de los HMD's consiste en una pantalla colocada delante de los ojos que presentan una misma imagen para ambos, o para un solo ojo. Este dispositivo tiene incorporado un sistema de rastreo que capta la orientación de la cabeza, lo que permite que el usuario pueda ver el mundo virtual como si este lo rodeara [Hollands, 1996].

5.3.2. HMD's Binoculares o Estéreo.

Estos dispositivos son similares a los HMD's Monoculares, con la diferencia de que utilizan pantallas independientes para cada ojo, de forma tal que se pueden presentar imágenes distintas para cada uno, aprovechando así la forma de visión natural o el efecto de visión libre cruzada mencionado anteriormente [Hollands, 1996].

5.3.3. Shutter Glasses.

Estos son lentes con obturadores para cada ojo que explotan el efecto de imagen entrelazada. Comúnmente este dispositivo no ofrece la misma libertad de movimiento, ya que, el usuario debe ver a una pantalla que presenta las imágenes para cada ojo de forma secuencial [Grossman, 2004].

5.4. Tipos de Guantes Virtuales según su funcionamiento.

Luego de los HMDs, uno de los dispositivos de realidad virtual más famosos son los guantes de realidad virtual. El propósito de estos guantes es medir la

posición de los dedos de la mano. Generalmente estos guantes deben permitir al usuario trabajar con objetos virtuales de la misma manera que trabajaría con objetos reales, alcanzándolos, tocándolos o de cualquier forma, manipulándolos por medio de la mano sin tener que recurrir al teclado o al *Mouse*.

Los guantes más conocidos se pueden clasificar, según su construcción y la forma en la que miden las acciones por parte del usuario, de la siguiente manera:

5.4.1. Guantes Ópticos (*DataGlove*):

Este guante, originalmente desarrollado por VPL, utiliza un conjunto de cables de fibra óptica (uno por cada articulación de cada dedo de la mano) para medir la flexión y extensión de los dedos. La luz emitida por un LED infrarrojo en uno de los extremos del cable, pasa a través de la fibra y es medida por un fotorreceptor en el otro extremo, variando la intensidad de luz que llega al receptor de acuerdo con la flexión de los dedos [Zimmerman, 1985]. Utilizando diodos fotorreceptores que reaccionan ante la amplitud de onda de la luz y forrando las paredes del tubo conductor con líneas longitudinales de color, es posible medir también el grado de desviación lateral de los dedos.

5.4.2. Guantes de Tinta Conductiva (*PowerGlove*):

El *PowerGlove* de *Nintendo Home Entertainment System*, utiliza cintas de plástico recubiertas de tinta conductiva dispuestas sobre las articulaciones de la mano. Al flexionar los dedos se dobla la cinta conductiva, esta aumenta su

longitud dispersando las partículas de carbón que lo componen, aumentando su resistencia eléctrica, este cambio corresponde proporcionalmente con el grado de flexión de la cinta [Hollands, 1996].

5.4.3. Guantes de Tensión (*CyberGlove*):

El *Virtex Cyberglove*, utiliza delgadas películas cuya resistencia varía ligeramente de acuerdo con la presión aplicada a cada articulación. Colocando dos de esas "cintas" de forma tal que una de ellas se estira al flexionar el dedo mientras que la otra se encoge, al conectar ambas a un circuito conocido como puente de "Wheatstone" [Dorf, 1995], se utiliza la relación de extensión y retracción entre las cintas para obtener un voltaje linealmente proporcional a la flexión del dedo [Krammer, 1990].

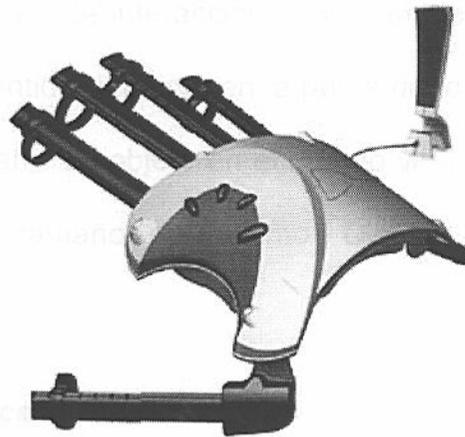


Fig. 5. Guante P5 de Essential Reality.

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

5.4.4. Guante Electro-Mecánico (EXOS *Dextrous Hand Máster*):

Este guante consiste en un exoesqueleto montado sobre la mano y conectado a cada dedo. Cada articulación de cada dedo coincide con su correspondiente articulación en el exoesqueleto, en la cual se encuentra un sensor capaz de medir el ángulo de flexión de dicha articulación. Se dice que este guante es más preciso que los modelos anteriormente mencionados y que inclusive es capaz de medir la desviación radial-cubital de los dedos (movimientos laterales). Pero lamentablemente es menos práctico de utilizar [Hollands, 1996].

5.5. Métodos de respuesta al usuario para los guantes virtuales.

Para que el uso de los guantes virtuales sea una experiencia más cercana a la realidad, es posible enviar una respuesta al usuario, a través del mismo, que complemente la sensación de interacción con el ambiente simulado, como por ejemplo, estimular el sentido del tacto en la punta de los dedos si se oprime un botón virtual o si se agarra un objeto en el mundo virtual, limitar la flexión de la mano proporcional al tamaño del mismo. Dichas respuestas se pueden clasificar en:

5.5.1. Force Feedback:

Permite al usuario "sentir" las fuerzas que podrían ser ejercidas por objetos virtuales. Para lograr este efecto y aplicar realmente una fuerza sobre el cuerpo humano, es necesario un exoesqueleto con articulaciones mecánicas sobre la parte del cuerpo que se desea afectar, limitando de alguna manera el

movimiento natural de dicha parte. Específicamente para la mano, existen dos enfoques, un exoesqueleto complementado con tendones metálicos que se tensan para restringir el movimiento o un conjunto de pistones neumáticos colocados entre los dedos y la palma de la mano que restringen el movimiento según sea necesario [MELLOTS, 2004].

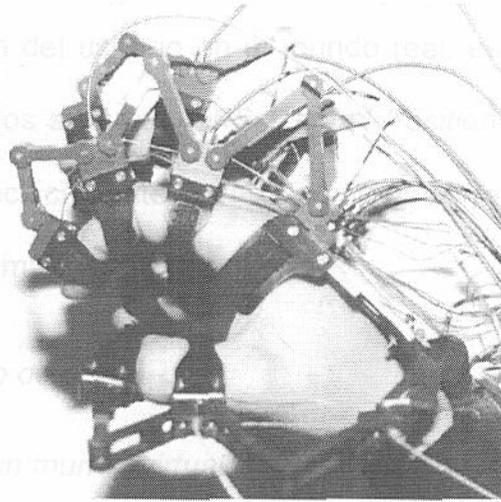


Fig. 6. Sistema force feedback para la mano por Immersion Ltd.
Fuente: <http://www.geocities.com/mellott124/>

5.5.2. Tactile Feedback:

El propósito de este sistema es estimular el sentido del tacto del usuario. Dado el amplio espectro de sensaciones que se pueden percibir por el tacto, ha resultado imposible implementar un sistema que logre imitar todas estas sensaciones. Los sistemas actuales están restringidos a sensaciones específicas como por ejemplo, dureza, temperatura, textura, vibración, etc. [Hollands, 1996].

5.6. Tecnologías de rastreo de movimiento y posicionamiento.

Para lograr una sensación integral de inmersión en un mundo virtual es importante tener una relación directa entre el movimiento del cuerpo y la acción en el mundo virtual, lo que se denomina "paralelismo cinético" (*motion-parallax*).

Los dispositivos de rastreo tienen la capacidad de proveer al sistema sobre la orientación y/o posición del usuario en el mundo real, esta información puede ser absoluta como en los sistemas GPS (*Global Position Satellite*) o relativa a un estado inicial. Específicamente para la computación gráfica, los usos más comunes de estos sistemas son los siguientes:

- *Control del punto de vista.*
- *Navegación en un mundo virtual.*
- *Selección y/o manipulación de objetos.*
- *Rastreo de objetos o instrumentos:* que tienen su contraparte en el mundo virtual, como por ejemplo la cirugía asistida por computador.
- *Animación de personajes:* a través de una técnica denominada rastreo de cuerpo completo (*full-body motion tracking*), donde personajes virtuales son animados por actores reales para la industria del cine.

5.6.1. Rastreadores Mecánicos.

El rastreador de posición mecánico implica un acoplamiento físico directo con el punto que se desea rastrear. La solución típica implica una serie articulada de

partes rígidas, dos o más, interconectadas por transductores electromecánicos, como potenciómetros, similar a un brazo robótico con sensores en las articulaciones. En la medida que se mueve el objetivo, se mueve el arreglo articulado y en consecuencia los potenciómetros varían su resistencia eléctrica. Conociendo la medida de las partes rígidas y utilizando los potenciómetros para calcular el ángulo formado entre dos piezas articuladas del arreglo, se puede estimar la posición de un extremo del brazo (acoplado al objetivo de rastreo) con respecto a la base (el otro extremo del brazo) mediante cálculos geométricos. Este método proporciona estimaciones muy exactas de la posición y orientación del objetivo; lamentablemente la estructura mecánica limita los movimientos que se pueden efectuar según su forma de construcción [VR-NAVY, 2000].

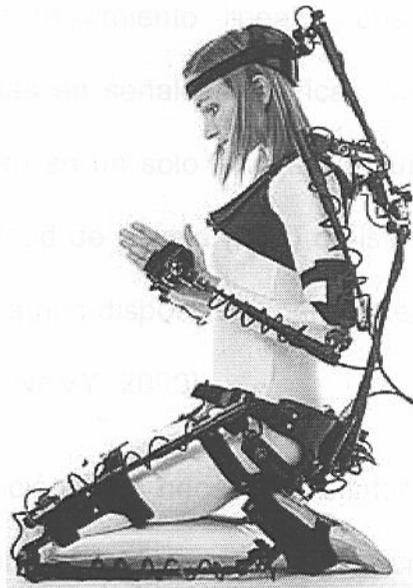


Fig. 7. Sistema de rastreo mecánico de cuerpo completo, Gypsy 4 de Metamotion
Fuente: <http://www.metamotion.com/gypsy/gypsy-motion-capture-system.htm>

5.6.2. Rastreadores Inerciales.

Los sistemas de navegación inercial o INS por sus siglas en inglés (*Inertial Navigation Systems*), llegaron a utilizarse extensivamente en barcos, submarinos y aviones durante los años 50. La navegación inercial se basa en dispositivos que detectan la aceleración y rotación en los tres ejes de coordenadas. Estos dispositivos son los llamados **giroscopios** y **acelerómetros**. El propósito de un giroscopio es medir la rotación angular y se han ideado múltiples métodos para lograrlo. Un giroscopio mecánico clásico detecta la estabilidad de una masa que rota sobre cardanes o gimbales. Los actuales avances en los sistemas Micro-Electro-Mecánicos (MEMS) ofrecen la potencial ventaja de desarrollar giroscopios que son muy pequeños y económicos. Mientras que los giroscopios miden el movimiento angular, los acelerómetros miden el movimiento lineal. Las aceleraciones de estos dispositivos son traducidas en señales eléctricas. Cuando un giroscopio y un acelerómetro se combinan en un solo dispositivo, junto con un mecanismo de control, se llama una unidad de inercia (IMU) o sistema de navegación inercial (INS). El INS utiliza estos dos dispositivos para detectar el movimiento relativo a un punto de origen [VR-NAVY, 2000].

Para conseguir la posición, es necesario contar con tres acelerómetros dispuestos ortogonalmente, de tal forma que puedan medir la aceleración en los tres ejes de coordenadas. Con las tres medidas resultantes se puede representar un vector de aceleración, que posteriormente se rota de acuerdo

con las medidas tomadas por los giroscopios, los cuales deben estar dispuestos ortogonalmente, de forma tal que sean capaces de medir las variaciones o la velocidad de rotación sobre los ejes. El tamaño de los acelerómetros y giroscopios electromecánicos actuales no es mayor al de una moneda, Los sensores de inercia son totalmente autónomos y no son vulnerables a interferencia por ruido o campos electromagnéticos. Sin embargo, a pesar de que miden la aceleración y la tasa de velocidad de forma muy precisa, la integración matemática de estos valores para calcular la posición, aumenta y propaga exponencialmente los pequeños errores en las mediciones; por ejemplo, si uno de los acelerómetros tiene un error diagonal de apenas 1 mili-g, la posición calculada divergiría de la posición verdadera con una aceleración de $0,0098 \text{ m/s}^2$. Luego de tan solo 30 segundos, la posición estimada estaría errada por 4,5 metros. Por otra parte, en relación a los giroscopios, un error en la orientación de 1 mili radián produciría un error de 1 mili-g en uno de los acelerómetros. Incluso los giroscopios más costosos presentan pequeños "errores", esta situación es conocida como "el caminante perdido", representada en la Fig. 8.

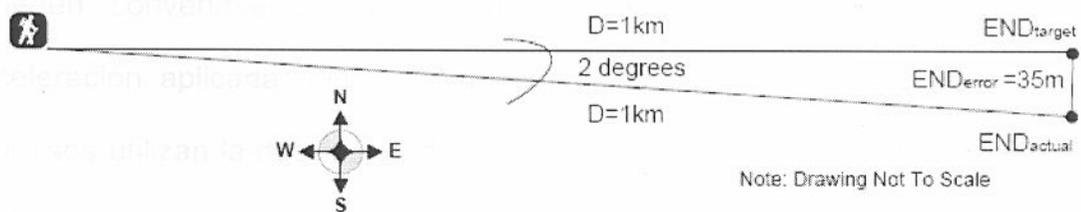


Fig. 8. Problema del caminante perdido.
Fuente: Freescale Semiconductors, RateGyro App. Notes.

5.6.2.1. Funcionamiento de los Giroscopios.

Los nuevos giroscopios MEM's, hacen uso del efecto de Coriolis para medir la tasa de velocidad angular con respecto a su eje normal, utilizando un masa microscópica ajustada a un marco de silicón de forma que solo se pueda mover en una sola dirección, al estar influenciada por una velocidad angular, la masa vibra con una frecuencia equivalente. El Anexo 1, presenta una explicación bastante detallada del funcionamiento interno de los giroscopios de *Analog Devices* [ANALOG, 2004].

5.6.2.2. Funcionamiento de los Acelerómetros.

Los acelerómetros MEM's, por su parte, pueden implementarse de varias formas, haciendo uso de un efecto piezoeléctrico, donde estructuras de cristal microscópicas son excitadas por fuerzas acelerativas, generando un voltaje. Otra forma más común es haciendo uso del principio de diferencia capacitiva, donde la aceleración causa el desplazamiento de pequeñas estructuras de silicón móviles, que a su vez causan un cambio de capacitancia al acercarse o alejarse de otras estructuras de silicón fijas. Estos cambios de capacitancia pueden convertirse en señales analógicas (voltajes) proporcionales a la aceleración aplicada (Fig. 9). Actualmente acelerómetros más sofisticados y precisos utilizan la desviación de la luz para medir la aceleración, siendo estos más costosos y muy grandes para poder ser utilizados en equipos de rastreo portátiles [KIONIX-TECH, 2004].

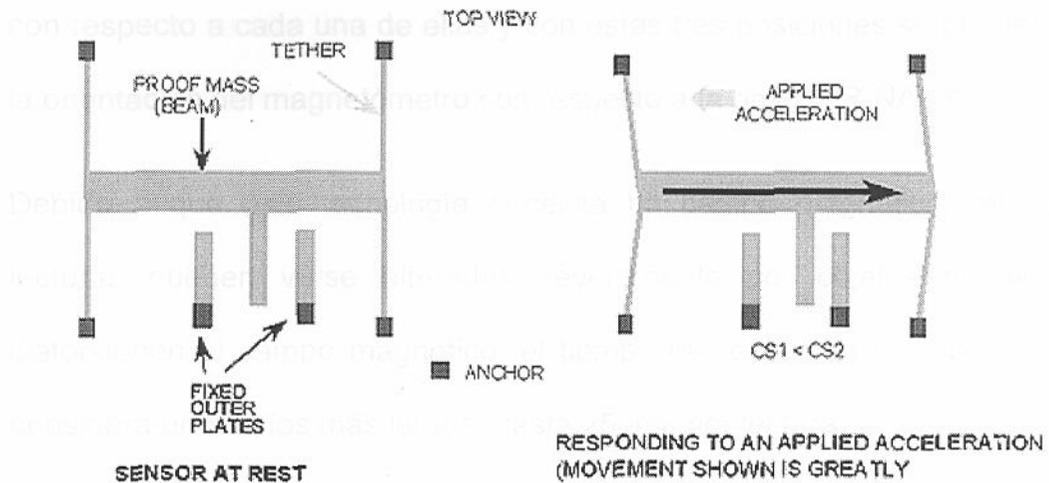


Fig. 9. Funcionamiento interno de los acelerómetros por diferencia de capacitancia
 Fuente: Freescale Semiconductors, Z-Axis Tilt Sensor Design.

Los acelerómetros son capaces de medir aceleraciones estáticas, como en el caso de la gravedad de la tierra, y aceleraciones dinámicas o inducidas. Comúnmente se utilizan para medir la inclinación haciendo uso de la gravedad de la tierra.

5.6.3. Rastreadores Magnéticos.

Los sistemas de rastreo electromagnéticos están conformados por una estación base que incluye tres bobinas dispuestas ortogonalmente una de la otra, cada bobina genera un campo magnético que luego es captado a su vez por otro dispositivo (magnetómetro) conformado igualmente por tres bobinas dispuestas de igual forma, cada una de ellas es capaz de medir la intensidad del campo magnético que las atraviesa. A medida que varía la distancia entre la base y el magnetómetro, varía la intensidad del campo magnético detectado. Las bobinas de la base se activan en secuencia, obteniendo la posición del magnetómetro

con respecto a cada una de ellas y con estas tres posiciones se puede conocer la orientación del magnetómetro con respecto a la base [VR-NAVY, 2004].

Debido a que esta tecnología necesita un campo magnético uniforme, las lecturas pueden verse alteradas severamente por objetos metálicos que distorsionen el campo magnético, el tiempo de respuesta de los sensores se considera uno de los más largos, hasta 250ms por lectura.

5.6.4. Rastreadores Ultrasónicos.

Estos sistemas calculan el tiempo que tarda un viajar una onda o pulso ultrasónico de un punto a otro. Para calcular únicamente la posición, es necesario un transmisor ultrasónico y un “dispositivo receptor”, conformado por tres receptores ultrasónicos como mínimo. Un pulso ultrasónico es emitido por el transmisor, en el mismo instante que comienzan a contar tres relojes de precisión, cuando el pulso es recibido por uno de los receptores, su reloj asociado se detiene. Con la velocidad del sonido (que es prácticamente constante) y los tiempos obtenidos se puede calcular la distancia entre el emisor y cada uno de los receptores (el tiempo que tarda en llegar una onda de un punto a otro se conoce como “tiempo de vuelo” de la onda) y por triangulación se puede conocer la posición casi exacta del dispositivo receptor. Agregando dos transmisores más al sistema se puede estimar la orientación de los receptores. El sonido viaja relativamente, rápido pero inclusive así, el tiempo de respuesta del sistema puede llegar a los 200ms por medición; además, el sistema es vulnerable a interferencia por otras fuentes de ondas ultrasónicas.

Otra consideración importante es que los receptores deben mantenerse orientados hacia los transmisores o viceversa [VR-NAVY, 2004].

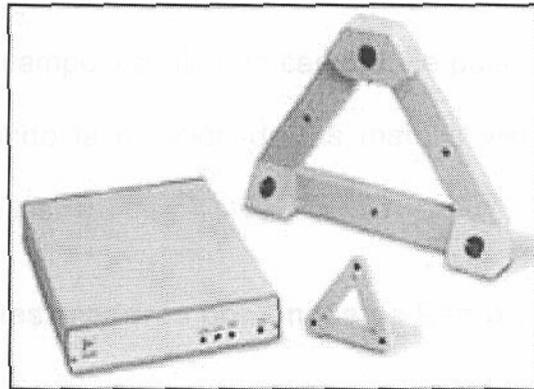


Fig. 10. Sistema de rastreo ultrasónico desarrollado por Logitech.
Fuente: <http://www.vrealities.com/logitech.html>

5.6.5. Rastreadores Ópticos.

Existe una variedad de técnicas de rastreo óptico. La técnica más utilizada en dispositivos de realidad virtual y HMD's funciona de forma similar al rastreo por ultrasonido, denominada PLADAR (*Pulsed Light Angle Direction And Range*). Esta utiliza luz infrarroja en vez de ondas de ultrasonido, siendo la velocidad de la luz tan rápida que la distancia entre el emisor y el receptor no se calcula midiendo el "tiempo de vuelo" sino la fuerza o intensidad de luz recibida [Hollands, 1996].

Otros sistemas de rastreo óptico están basados también en cámaras. Una de las técnicas con cámara consiste en colocar la cámara sobre la cabeza del usuario dentro de una habitación con *leds* que parpadean siguiendo un patrón que distingue su ubicación en la habitación, al detectar los parpadeos con la

cámara y conociendo el patrón, se puede calcular el resto. Otro uso para la cámara, es dirigirla hacia el usuario, el cual tiene un conjunto de marcas reflectivas sobre la parte del cuerpo que desea rastrear, cuando estas marcas se encuentran en el campo visual de la cámara, se puede calcular la posición y orientación comparando la posición de las marcas vista por la cámara y la distancia real entre ellas.

5.6.6. Rastreadores por Ondas de Radio y Microondas.

Las ondas de radio y las microondas no se han explotado plenamente para el rastreo de los movimientos del cuerpo humano, se utilizan extensamente en sistemas de navegación y ayudas de aterrizaje en aeropuertos y en sistemas de radar. Las técnicas basadas en ondas electromagnéticas (radio y microondas) tienen un mayor alcance que los campos magnéticos, ya que, la energía irradiada en un radio r se disipa en $\frac{1}{r^2}$ mientras que el gradiente de los campos magnéticos decae con una razón de $\frac{1}{r^4}$. Además las ondas de radio no se ven afectadas prácticamente por el viento o la temperatura, por lo que no se ven comprometidas en espacios abiertos, como los sistemas acústicos podrían estarlo. Desafortunadamente las ondas de radio se atenúan rápidamente en el agua, por lo que el cuerpo humano resulta opaco para todas las radio frecuencias utilizables para mediciones [VR-NAVY, 2000].

La mayoría de los sistemas de navegación por radio funcionan siguiendo el

mismo principio que los sistemas acústicos, calculando el "tiempo de vuelo" de las ondas, solo que estas viajan un millón de veces más rápido que el sonido, esto también aumenta la dificultad en el cálculo, ya que se necesitan relojes con una frecuencia superior a los 300GHz para medir la distancia con una exactitud de 1mm, lo que implica electrónica costosa. Todavía se realizan estudios para reducir el volumen y costo de los componentes necesarios.

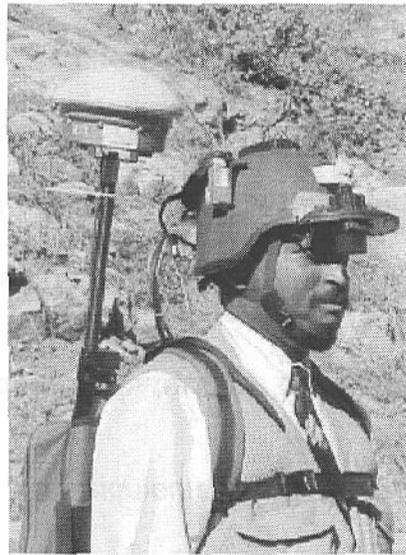


Fig. 11. Prototipo de rastreo por GPS, departamento de defensa de USA
Fuente: <http://www.cs.nps.navy.mil/people/faculty/capps/4473/projects/chang2/Full.htm>

Los sistemas de posicionamiento global (GPS) necesitan, para el cálculo de la orientación, una variación de posición o velocidad con la cual extrapolar la orientación, ya que cuentan con un solo receptor. Sin embargo, por el potencial que ofrece el GPS y su utilización para corregir los errores de posición y orientación de otros sistemas de rastreo, se convirtió en la base de los sistemas Híbridos.

• **Ve 5.6.7. Rastreadores Híbridos.**

Los sistemas de rastreo híbrido están compuestos por dos o más de los sistemas mencionados anteriormente. Estas combinaciones se llevan a cabo generalmente para corregir las mediciones realizadas por un método más rápido pero menos preciso, utilizando otro método más preciso, pero contrariamente, más lento.

Las combinaciones más comunes utilizadas hoy en día son los sistemas de rastreo Inercial con sistemas de GPS o ultrasonido, de manera que se pueden compensar o corregir los errores derivados de la integración de la aceleración cada cierto intervalo de tiempo [VR-NAVY, 2000].

• **5.6.8. Comparación entre los diferentes tipos de rastreadores.**

Se deben observar las siguientes características que diferencian los sistemas de rastreo y posicionamiento:

- *Grados de libertad:* Si son capaces de registrar desplazamiento en los ejes cartesianos (posición en X, Y, Z) y orientación o giros sobre los ejes (*Yaw, Pitch, Roll*), se dice que el dispositivo permite “seis grados de libertad” (6DOF: *six-degrees-of-freedom*).
- *Precisión:* La resolución de los dispositivos según la medida más pequeña que pueden rastrear, como por ejemplo 1 mm en posición o 0.1 grados de orientación, sobre cada uno de los ejes.

el voltaje

- *Velocidad:* Con que frecuencia se realizan los muestreos y con que retraso o "latencia" con respecto al movimiento causante.

Un voltaje $V_r = V_{max} / 2$, donde V_{max} es el voltaje máximo que puede ser convertido por el ADC.

- *Rango de Movimiento:* Conocido también como el rango de acción del usuario. Viene expresado según las limitaciones inherentes a la tecnología y construcción del dispositivo que limitan de alguna forma el movimiento del usuario.

- *Robustez:* Resistencia a interferencias.

- *Propensión al error.*

- *Precio.*

5.7. Funcionamiento de un Convertidor Analógico Digital (ADC).

En electrónica, un convertidor Analógico Digital (abreviado ADC o A/D) es un dispositivo que transforma una señal continua en números discretos, específicamente un voltaje en un número "binario" equivalente a su magnitud. El principio de operación básico es utilizar un comparador digital que determina cuando o no activar un bit particular del número binario resultante. El método implementado por el A/D utilizado en este proyecto se denomina "**método de aproximaciones sucesivas**", tiene como principal ventaja su simplicidad de implementación pero como inconveniente presenta un retardo en el tiempo de conversión, necesario para contar los "n" bits de resolución [ADC-2001].

El método, como su nombre lo indica, consiste en realizar comparaciones entre

el voltaje de entrada y un voltaje de referencia que varía en cada uno de los ciclos que tarda la conversión. La primera comparación se realiza con respecto a un voltaje $V_r = V_{max} / 2$, donde V_{max} es el voltaje máximo de referencia que especifica el rango de la conversión (usualmente el voltaje con que se alimenta el ADC), y su valor binario corresponde a una palabra de tamaño igual al rango máximo del convertidor con solo el bit más significativo (1000 para un A/D de 4-bits de rango), si el voltaje de entrada es mayor o igual a V_r determina que el bit más significativo debe ser uno (1), por el contrario si es menor que V_h debe ser cero (0). En el siguiente impulso de reloj, se efectúa una segunda comparación del valor de entrada V_e correspondiente a la palabra 1100 si la comparación anterior da un valor positivo, en caso contrario se compara contra 0100, la salida del comparador determina el valor que debe memorizarse con un bit de peso $V_{max}/4$, configurándose de esta forma la palabra de salida digital una vez efectuadas las "n" comparaciones sucesivas.

Algunas características importantes de un ADC son las siguientes:

Rango: Indica el número de bits de salida o número de valores discretos que el ADC puede producir, usualmente esta expresado en bits. Por ejemplo, un ADC que convierte una entrada analógica a uno de 256 valores discretos, se dice que tiene una resolución de 8 bits, que son el número de bits necesarios para representar 256 valores distintos. $2^8 = 256$, $2^{10} = 1024$, etc.

Resolución: También se entiende como el voltaje necesario (señal analógica)

para lograr que en la salida (señal digital) haya un cambio del bit menos significativo. (LSB). Para hallar la resolución se utiliza la siguiente fórmula:

$$\text{Resolución} = \frac{V_{\text{max}}}{2^n - 1}$$

Donde:

n = número de bits del ADC

V_{max} = es el voltaje de referencia a la entrada del convertidor para obtener una conversión máxima (todas las salidas son "1")

Precisión: depende del error en la conversión o error de cuantización, dicho error está expresado en unidades LSB (*Least Significant Bit* – bit menos significativo), típicamente 1 sobre el rango, para el caso de 8 bits de rango un LSB equivale a 1/256. El error de cuantización se debe a la resolución finita de los ADC, la magnitud de este error en el momento en que se toma la muestra varía entre cero y LSB/2 o la mitad del valor más pequeño que el ADC puede convertir.

Tasa de muestreo y conversión: Principalmente es el tiempo que se tarda el ADC en realizar la conversión, este tiempo define una tasa en la que nuevas lecturas pueden ser procesadas. Para que una señal analógica, acotada pero variante, sea convertida con mayor exactitud, la tasa de muestreo debe ser mayor que el doble del ancho de banda de la señal (Teorema de muestreo de Shannon-Nyquist). Ya que la tasa de muestreo de un ADC dado no puede ser

cambiada fácilmente y los mismos no pueden hacer conversiones instantáneas, el valor a convertir debe mantenerse "constante" durante el tiempo que se realiza la conversión, para ello se utiliza un circuito conocido en inglés como "sample and hold", conformado por un capacitor que almacena el voltaje a convertir y una compuerta o *switch* digital que desconecta el capacitor de la entrada hasta terminar la conversión. Como resultado la tasa de muestreo es el número de muestras que se pueden "tomar" y convertir en un tiempo determinado.

5.8. Puerto USB (Universal Serial Bus).

El puerto USB es considerado actualmente la panacea para conectar dispositivos al computador. Son muchas las ventajas que ofrece sobre otros puertos de entrada/salida, hoy en día casi todos los periféricos que comúnmente se conectaban a otros puertos ofrecen la opción de conectarse por USB.

Las especificaciones del USB 2.0 contienen más de 650 páginas de información sobre hardware, software, dimensiones de los conectores utilizados, características de las señales eléctricas entre los dispositivos y el computador.

En primer lugar el puerto USB implementa un diseño asimétrico, que consiste en un solo "servidor" y múltiples dispositivos conectados en una estructura de árbol a través de "concentradores" especiales (*host controlled*). El número de dispositivos que se pueden conectar a un solo servidor se encuentra restringido

a 127 por limitaciones en los tiempos de propagación de las señales. La comunicación a través del bus se realiza bajo la supervisión del *host* (servidor), ningún dispositivo conectado puede enviar información sin ser solicitada por el *host*, la información viaja en paquetes similares a las redes *ethernet*, donde esta contenida la identificación del dispositivo que debe recibir la información cuando esta se origina en el *host*. Para enviar información, los dispositivos deben primero recibir una señal (*token*) por parte del *host*, que viene incrustada en los paquetes de control por medio de los cuales el *host* hace solicitud de información o activación de alguna función por parte del dispositivo.

Cada cierto tiempo el controlador USB del *host* transmite un paquete de control con la finalidad de detectar nuevos dispositivos conectados o si se desconecta algún dispositivo, esta fase se conoce como enumeración; a los nuevos dispositivos se les asigna una dirección por la cual responderán las solicitudes del *host* a partir de ese momento. Inmediatamente después de asignar una dirección a un dispositivo, el *host* le pide información adicional sobre él para completar su proceso de enumeración. La información requerida consiste en su descripción e identificación, sus funciones y requerimientos de energía, tamaño y frecuencia en que se intercambiara información entre otros datos dependientes de los tipos de dispositivo, clasificados y definidos en las especificaciones USB [Axelson, 2005].

Durante el proceso de enumeración, la información enviada por el dispositivo al *host*, esta definida y organizada en una estructura jerárquica de "descriptores"

(Figura 12). Los descriptores definen las características específicas de los dispositivos. Todos los dispositivos USB deben implementar una jerarquía de descriptores que puede variar entre dispositivos. Los descriptores definidos en las especificaciones USB son [USB-SPEC, 2000]:

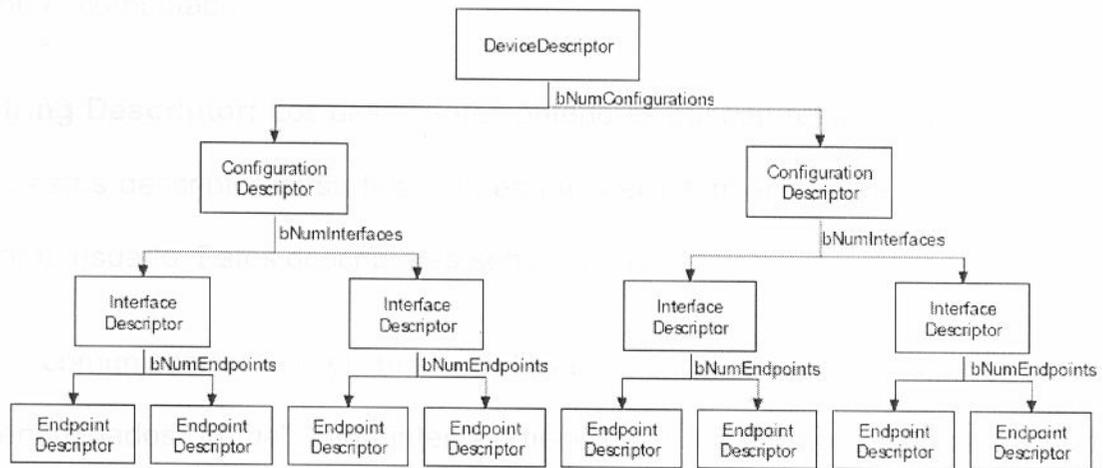


Fig. 12. Jerarquía de los descriptores de un dispositivo USB.

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

Device Descriptor: Contiene la información general del dispositivo, como el número que identifica la empresa fabricante, el código del producto, el número de serie, la clase de dispositivo y el número de configuraciones. Este descriptor es único.

Configuration Descriptor: Provee información sobre los requerimientos de corriente del dispositivo y el número de interfaces activas bajo dicha configuración.

Interface Descriptor: Especifica el número de puntos Entrada/Salida que tiene el dispositivo, además de el tipo de interfaz.

Endpoint Descriptor: Identifica el tipo de transferencia y las características de esta, como la frecuencia de transmisión. Describe un punto de Entrada/Salida con el computador.

String Descriptor: Los descriptores anteriores pueden referenciar a uno o más de estos descriptores, su finalidad es proveer información que pueda ser leída por el usuario. Estos descriptores son opcionales.

La comunicación en el bus se lleva a cabo en intervalos de tiempo, denominados "*frame*", cada intervalo tiene una duración típica aproximada de 1 ms, dentro de ellos pueden ocurrir varias transacciones entre el host y distintos dispositivos. USB especifica cuatro tipos distintos de paquetes: paquetes *Token*, que indican el tipo de transacción; paquetes *Data*, que contienen la carga de información; paquetes *Handshake*, utilizados para reportar errores o confirmar la recepción de otros paquetes y los paquetes conocidos como *Start of frame*, que indican el comienzo de un nuevo intervalo [Peacock, 2002].

Paquetes Token: Existen 3 tipos de ellos.

- In – Informan al dispositivo USB que el host desea recibir / leer información.
- Out - Informan al dispositivo USB que el host desea enviar / escribir información.
- Setup – Utilizados para dar comienzo a transferencias de control.

Estos paquetes tienen el siguiente formato

Sync	PID	ADDR	ENDP	CRC5	EOP
------	-----	------	------	------	-----

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

Paquetes Data: Son capaces de transmitir entre 0 y 1023 bytes de datos por paquete, se dividen en 2 tipos. Data0 y Data1.

Sync	PID	Data	CRC16	EOP
------	-----	------	-------	-----

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

Paquetes Handshake: Se pueden clasificar en 3 tipos y contienen simplemente el PID del dispositivo.

- ACK – Confirma que un paquete a sido recibido con éxito.
- NAK – Reporta que el dispositivo no puede enviar o recibir paquetes temporalmente, también se utiliza para especificar que no hay datos disponibles para enviar.
- STALL – Indica que el dispositivo se encuentra en un estado que requiere la atención del Host.

Paquetes Start of Frame: Consiste en un número de 11 bits enviados por el host cada milisegundo \pm 500 nanosegundos.

Las especificaciones USB también cuentas con cuatro tipos de transferencia:

- **Isochronous:** Este método permite transferencias de grandes

6. Muchas cantidades de datos (hasta 1023 bytes), sin embargo la integridad de los datos no se comprueba. Este tipo de transferencias es ideal para aplicaciones donde la transferencia debe ser rápida y la pérdida de datos no es un factor crítico, como en el manejo de audio.

- **Bulk:** Este método permite igualmente enviar gran cantidad de datos asegurando la integridad de los mismos, pero la frecuencia de envío no puede ser garantizada.
- **Interrupt:** Este tipo de transferencia permite enviar pequeñas cantidades de datos asegurando la integridad y frecuencia de envío.
- **Control:** Utilizada principalmente para el control y configuración de los dispositivos, todos los dispositivos deben implementar este tipo de transferencia.

[Peacock, 2004] ofrece una versión resumida y muy completa de las especificaciones del protocolo USB 2.0 en su página Web <http://www.beyondlogic.com>

6. Metodología.

El proyecto se dividió en 5 etapas correspondientes a cada uno de los componentes fundamentales del set, permitiendo enfocar la atención en el desarrollo de cada uno de los componentes por separado (divide y vencerás): Convertidor A/D e Interfaz con el computador, Guante Virtual, Rastreador de posición para el guante, Rastreador de posición para la cabeza y software de control para los dispositivos.

Cada una de estas etapas fueron realizadas siguiendo la metodología de cascada iterativa mejorada [Charvat, 2003]. Esta metodología parte del método de cascada tradicional [Royce, 1970]. Las fases propuestas en la metodología de cascada cubren específicamente el establecimiento claro y concreto de los requisitos del proyecto, la creación de un diseño claro a partir de los requisitos, la implementación del diseño y por último probar la implementación. La metodología de cascada simple no puede ser aplicada en la práctica por su inflexibilidad y por no poseer mecanismos para corregir errores en fases anteriores o salvar imprevistos que vayan apareciendo, es por esto que se combina con ciclos iterativos sobre las etapas, permitiendo regresar y repasar sobre las fases anteriores, con el fin de modificar los resultados. La incorporación de prototipos como resultado de cada iteración en un escalón, permite medir la factibilidad del proyecto antes de completar el desarrollo y decidir si es posible pasar al próximo escalón que forma la cascada.

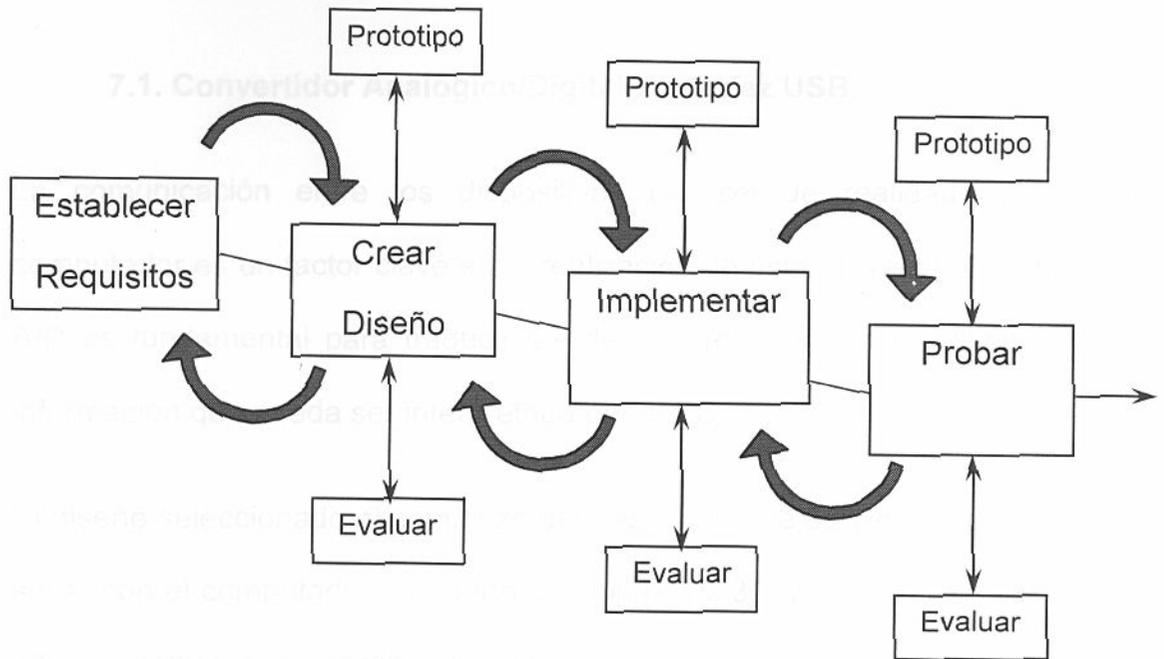


Fig. 13. Metodología de cascada iterativa mejorada.

Fuente: El Autor.

El primer escalón de cada cascada se dedicó a recopilar la información necesaria para seleccionar el tipo de dispositivo a desarrollar y la solución a diseñar. El resto de los escalones se combinaron con prototipos para ir evaluando los resultados y de ser necesario volver a fases anteriores de desarrollo para incorporar una nueva funcionalidad o corregir algún error.

Las cinco partes principales presentaron varias interdependencias que no permitieron su desarrollo en paralelo y limitaron su desarrollo de forma independiente. La fase más importante fue el desarrollo del Convertidor A/D y la Interfaz con el Computador, ya que el resto de los dispositivos dependen de este.

7. Desarrollo.

7.1. Convertidor Analógico/Digital e Interfaz USB.

La comunicación entre los dispositivos del *set* de realidad virtual y el computador es un factor clave en la realización de este proyecto. El convertidor A/D es fundamental para traducir los datos arrojados por los dispositivos en información que pueda ser interpretada por el PC.

El diseño seleccionado al comienzo del desarrollo, fue basado en una conexión serial con el computador utilizando un chip MAX232, y un chip ADC0805 con 8 bits de rango para realizar la conversión y un microcontrolador PIC16F85 encargado de transmitir secuencialmente los 8 bits arrojados en paralelo por el ADC. Esta implementación fue seleccionada por ser económica, los componentes se consiguen fácilmente en el país y su funcionamiento demostró ser confiable durante las pruebas con una sola entrada analógica. Lamentablemente, durante la construcción del Guante virtual, fue necesario incorporar un circuito para multiplexar varias entradas analógicas al ADC, la nueva versión del circuito dio resultados erráticos durante las pruebas con cinco entradas analógicas (proveniente de los dedos) por falta de una lógica de "sample and hold" (ver sección 10.7) y por falta de sincronización entre el microcontrolador y el ADC. Debido a esto, fue necesario reevaluar el diseño, ya que se estaba tornando muy complejo y no satisfacía los objetivos del proyecto. Optando por utilizar un microcontrolador de última generación de Microchip (por poseer una arquitectura familiar) el PIC18F4550, con ADC integrado, puerto

UART (conexión serial) y módulo para comunicación USB integrado, entre otras ventajas.

Este nuevo circuito (Fig. 13), además de ser más rápido que la versión anterior y ofrecer otro número de ventajas, sigue siendo igual de económico y más confiable.

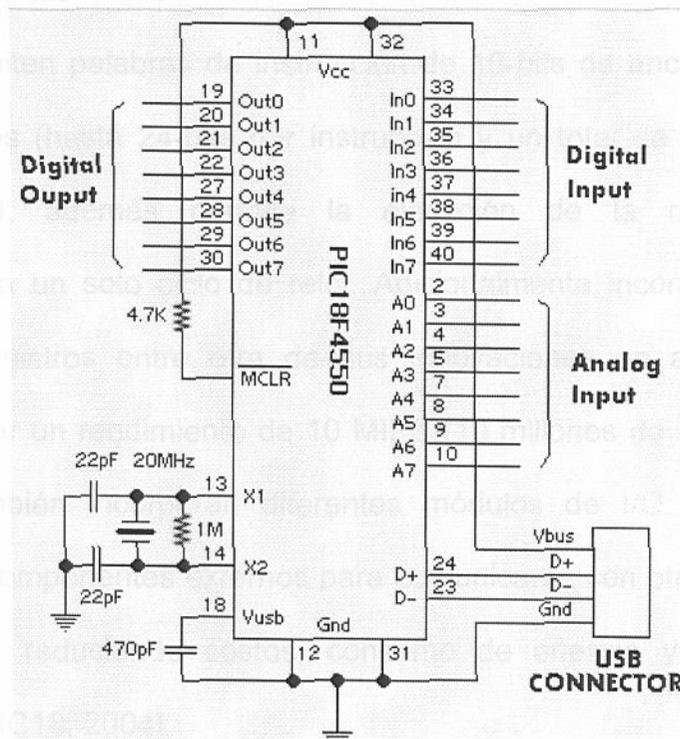


Fig. 14. Circuito base del pic18f4550 con conexión USB.

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

7.1.1. Microcontrolador Pic18f4550.

La familia PIC18CXXX son microcontroladores de alto rendimiento, con procesadores de 16-bits que incorporan una arquitectura RISC (*Reduced Instruction Set*) avanzada. Esta serie de MCU's (*MicroControler Units*) posee características mejoradas con respecto a sus predecesores, canales (buses) separados para las instrucciones y los datos, basados en la arquitectura *Hardvard*, permiten palabras de instrucción de 16-bits de ancho con 8-bits de datos separados (hasta 24-bits por instrucción y un total de 77 instrucciones implementadas), además permite la ejecución de la mayoría de las instrucciones en un solo ciclo de reloj. Adicionalmente incorporan un amplio conjunto de registros entre otra de sus innovaciones en arquitectura, que permite alcanzar un rendimiento de 10 MIPS (10 millones de instrucciones por segundo). También incorporan diferentes módulos de I/O que reducen la necesidad de componentes externos para comunicarse con otros dispositivos o el computador, reduciendo costos, consumo de energía y aumentando la confiabilidad [PIC18, 2004].

Específicamente el Microcontrolador PIC18F4550 de la familia PIC18CXXX, fue seleccionado por las siguientes características:

- Fabricado por *Microchip* y sucesor de la familia PIC16CXXX con la que se ha trabajado anteriormente en los laboratorios de Arquitectura del Computador.

- Las herramientas de programación facilitadas por *Microchip* incluyen un compilador en lenguaje C, que entre otras ventajas, permite una estructura de código más coherente que el lenguaje ensamblador (utilizado comúnmente para programar los MCU's).
- Programación *FLASH* mejorada, permite borrar y escribir el *firmware* en cuestión de segundos, además se puede borrar secciones específicas dejando intactas las demás, permitiendo la “auto programación” del Microcontrolador.
- Módulo USB integrado que cumple las especificaciones USB 2.0, permite *Low Speed* (1.5 Mb/s) y *Full Speed* (12 Mb/s), soporta los modos de transferencia, *Control*, *Interrupt*, *Isochronous* y *Bulk*. Permite hasta 32 “*endpoints*” (16 bidireccionales). 1 Kbyte de memoria RAM compartida entre el chip y el computador para USB. Puerto SPP (*Streaming Parallel Port*) para transferencias de USB de alta velocidad.
- Convertidor Analógico Digital de 10-bits de rango, hasta 13 canales de entrada, con tiempo de adquisición programable.
- Módulo USART para comunicación serial.

Entre muchas otras características especificadas en el Anexo 2.

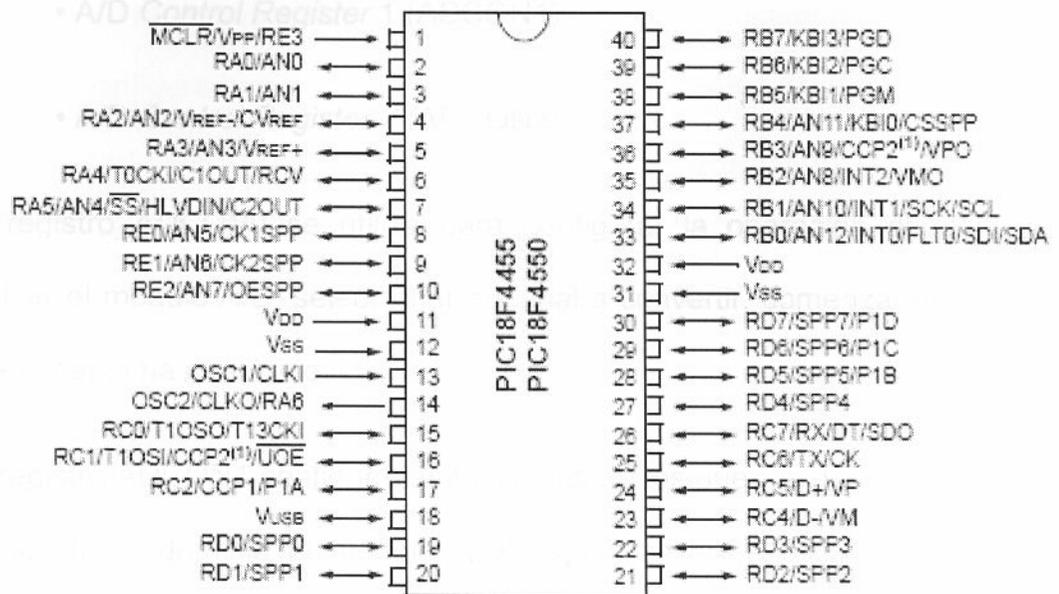


Fig. 15. Diagrama del Chip PIC18F4550.
Fuente: Data Sheet PIC18F4550

12.1.2. Configuración del ADC.

El módulo convertidor Analógico a Digital (A/D) integrado en el pic18f4550 puede utilizar hasta 13 canales de entrada, permitiendo la conversión de un valor analógico en su correspondiente valor digital con 10-bits de precisión. Para operar y configurar el módulo, el microcontrolador cuenta con cinco registros específicos, 3 de control y 2 para almacenar el resultado de la conversión.

- A/D Result High Register (ADRESH)
- A/D Result Low Register (ADRESL)
- A/D Control Register 0 (ADCON0)

- *A/D Control Register 1 (ADCON1)*

- *A/D Control Register 2 (ADCON2)*

El registro ADCON0 se utiliza para configurar la operación de conversión, activar el módulo A/D, seleccionar el canal a convertir, comenzar la conversión y verificar si ha concluido.

El registro ADCON1 configura las funciones de los puertos de entrada sobre los cuales tiene dominio el módulo A/D, especifica si se utilizara un voltaje de referencia distinto al voltaje de entrada del chip, especifica cuales de estas entradas funcionaran como entradas digitales o analógicas del Pic.

El registro ADCON2 sirve para configurar el reloj del módulo A/D, seleccionar la fuente de reloj, los tiempos de adquisición del valor de entrada y la justificación del resultado en los registros ADRESH y ADRESL.

Las especificaciones detalladas de cada registro se presentan en el Anexo 5.

La conversión se realiza utilizando el método de aproximaciones sucesivas. Para este proyecto los valores de operación calculados del ADC son los siguientes:

Voltaje de entrada: 5 volts.

Rango: 10 bits.

Resolución: 5 volts / 1023 valores discretos posibles = 0,00488 volts (4,88 milivolts).

Error de Quantización = 2,44 mVolts.

La Tasa de muestreo el A/D depende directamente del tiempo de adquisición, el cual es configurado por en el registro ADCON2 y depende de la frecuencia de reloj utilizada para operar el microcontrolador. Según las especificaciones del MCU (Anexo 6) el tiempo de adquisición calculado dentro de los valores de operación máximos del ADC es de hasta 6.4 microsegundos, utilizando un pulso de conversión máximo equivalente a 64 pulsos de reloj del microcontrolador operando a 20Mhz y utilizando 10 pulsos de conversión para realizar una conversión completa de 10bits, el convertidor A/D del Pic18f4550 presenta una tasa de muestreo máxima de:

$\text{Frecuencia} / 64 * 10 = 31250 \text{ ciclos por segundo}$

$1\text{seg} / 31250 = 0,000032 \text{ segundos} = 32 \text{ microsegundos} + 6.4 = 40 \text{ microsegundos aproximadamente.}$

7.1.3. Implementación y configuración del módulo USB.

Explicar el funcionamiento interno del módulo USB integrado en el microprocesador PIC se escapa del alcance de este proyecto, algunos detalles sobre el funcionamiento e implementación del módulo están presentados en el Anexo 3. La información contenida es muy específica por su naturaleza, y cierto conocimiento sobre USB, es necesario para su comprensión. Más información sobre USB puede ser consultada en las especificaciones del puerto, disponibles en www.usb.org, en el foro de desarrolladores de www.usb-if.org y en los extractos de dichas especificaciones del Anexo 4.

Para programar el módulo USB se utilizaron los registros específicos del módulo explicados también en el anexo.

- *USB Control register (UCON)*
- *USB Configuration register (UCFG)*
- *USB Transfer Status register (USTAT)*
- *USB Device Address register (UADDR)*
- *Frame Number registers (UFRMH: UFRML)*
- *Endpoint Enable registers 0 through 15 (UEPn)*

Siguiendo las especificaciones en el documento de especificaciones de USB 2.0, el programa de control del módulo fue diseñado para utilizar un método de transferencia de interrupción y responder a las especificaciones para dispositivos de interfaz humana (HID). El *firmware* fue desarrollado basándose en algunos ejemplos proporcionados por Microchip, siguiendo la guía de desarrollo para usuarios en el Anexo 4.

Para implementar un dispositivo USB, este, según las especificaciones del protocolo, debe estar en la capacidad de responder a un conjunto mínimo de peticiones por parte del *host* a través de un *endpoint* con método de transferencia control, como realizar el proceso de enumeración y solicitar la transmisión o notificar que se enviaran datos.

El *framework* de desarrollo proporcionado por *Microchip* consiste en un grupo de archivos que manejan las tareas generales de USB como llevar a cabo el proceso de enumeración y responder a paquetes de control; en general, contiene la estructura mínima necesaria para implementar un dispositivo USB simple con tipo de transferencia control y un solo *endpoint*. Sobre este *framework* se realizaron los cambios para implementar los dispositivos del proyecto. La estructura del *framework* es la siguiente:

Usbmmmap.c: reserva memoria para las variables, *endpoints* y otros *buffers* usados durante el proceso de comunicación USB.

Usbdrv.c: Contiene las funciones para detectar cuando se conecta y desconecta el dispositivo, monitorea y responde interrupciones generadas en el controlador USB del chip.

Usbctrltrg.c: Contiene las funciones para manejar las transacciones en las transferencias de tipo control. Estas funciones decodifican los paquetes de *setup* recibidos, administran el envío y recepción de datos en esta forma e transferencia, incluyendo las fases de *setup* con el *host*.

Usb9.c: Contiene las funciones necesarias para responder a las peticiones definidas en el capítulo 9 de las especificaciones del protocolo USB. Dichas funciones proveen de apuntadores a los descriptores y otros datos requeridos por el *host* y tomar acciones ante peticiones como cambiar de configuración.

Hid.c: Contiene funciones para administrar las tareas específicas de los

dispositivos HID, decodifica los paquetes *setup* dirigidos a las interfaces HID, define los apuntadores a las estructuras de datos a enviar y los espacios de memoria donde se recibirán los datos durante las transferencias de control e *interrupt*, responde a otras peticiones específicas de la clase HID.

Usbdesc.c: Contiene las estructuras que guardan los descriptores del dispositivo.

User_generic_hid.c: Contiene la mayor parte del código agregado para el funcionamiento del dispositivo, Las funciones que obtienen los datos o convierten los datos que el dispositivo enviara al *host*.

Main.c: Inicializa el sistema y ejecuta el ciclo que chequea el estado del *bus*, llama las funciones especificadas en *user_generic_hid.c* y se repite hasta que se cumple cierta condición.

7.2. Construcción del guante virtual.

El diseño del guante virtual desarrollado para este proyecto, esta basado en los sensores de flexión ópticos patentados por Tomás G. Zimmerman en 1982 (Patente N° 4,542,291, Anexo 7). El principio de su invento es utilizar, como se explica en el marco teórico, un diodo emisor de luz y una fotorresistencia colocados en los extremos de un tubo flexible. Cuando el tubo se encuentra completamente extendido, la luz del diodo llega directamente a la fotorresistencia, cuando se flexiona una cantidad de luz directa y luz reflejada por las paredes del tubo llega a la fotorresistencia, pero en menos intensidad,

hasta el punto que solo luz reflejada o ninguna intensidad de luz en lo absoluto llega a ser captada.

El circuito ensamblado para cada dedo se resume en la Fig. 15. La fotorresistencia utilizada no presenta ninguna resistencia en ausencia de luz, de forma que la diferencia de potencial en R2 se hace igual a $V_{ss} - V_{cc} = 5$ volts. Cuando la fotorresistencia se satura de luz, dicho de otra manera, recibe una intensidad grande de luz directa, llega a presentar una resistencia equivalente a 20kOhms, de forma que la diferencia de potencial disminuye hasta 0 volts. Cada uno de los “tubos” ensamblados se fijó a cada uno de los dedos de un guante de tela, para que el usuario pudiera ponerse y quitarse el dispositivo fácilmente.

El voltaje que utiliza el guante proviene directamente del voltaje del puerto USB, las salidas de cada circuito construido para los dedos (Vout) se conectan a un canal del convertidor A/D del PIC (Específicamente las entradas AN0 a AN4 – pines 2, 3, 4, 5 y 7 del microcontrolador), comenzando por el dedo pulgar hasta el dedo meñique.

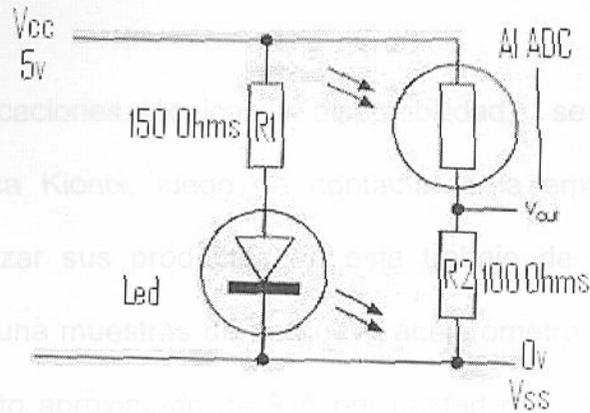


Fig. 16. Conexión de la fotorresistencia y el LED para medir la flexión de los dedos.
Fuente: El Autor.

7.3. Construcción del dispositivo de rastreo de posición y orientación para el guante.

Para la construcción de este dispositivo se intentó replicar el sistema de rastreo por medio de ultrasonido. El diseño consistía en implementar tres sonares ultrasónicos, pero a pesar de ser un modelo económico, los conocimientos técnicos necesarios y las herramientas de medición necesarias para implementar el prototipo eran insuficientes. Para la fecha en que se presentó la propuesta de este proyecto, apenas se habían introducido al mercado dispositivos MEM's (giroscopios y acelerómetros) de bajo costo que se hicieron más populares al momento de reevaluar el diseño. Se decidió entonces desarrollar un dispositivo de rastreo Inercial haciendo uso de tres giroscopios y tres acelerómetros MEM's.

Por sus especificaciones técnicas y disponibilidad, se decidió adquirir los giroscopios marca Kionix, luego de contactar a la empresa y expresar la intención de utilizar sus productos en este trabajo de investigación. Kionix accedió a donar una muestra de su nuevo acelerómetro tri-axial KMX52-1050 que tiene un costo aproximado de \$15 por unidad para compras superiores a las 100 piezas, pero el costo de sus giroscopios de precisión se encuentra fuera del presupuesto de este dispositivo y la empresa no los podía donar. Las especificaciones del acelerómetro KMX52-1050 pueden ser consultadas en el Anexo 8.

Para el giroscopio se decidió adquirir el modelo ADXR300 de *Analog Devices*, con un costo de \$17 por unidad. Las especificaciones pueden ser consultadas en el Anexo 9.

Para ensamblar el dispositivo de rastreo se colocaron los giroscopios con sus ejes normales perpendiculares entre sí formando 3 caras de un cubo, de forma que puedan medir las rotaciones en los ejes. El acelerómetro tri-axial se colocó sobre el giroscopio de la base del cubo, con su normal alineada con este. Los ejes del dispositivo y el signo de los valores medidos está representado en la Fig. 16. El eje Z de arriba hacia abajo, el eje X hacia los lados y el eje Y de atrás hacia adelante.

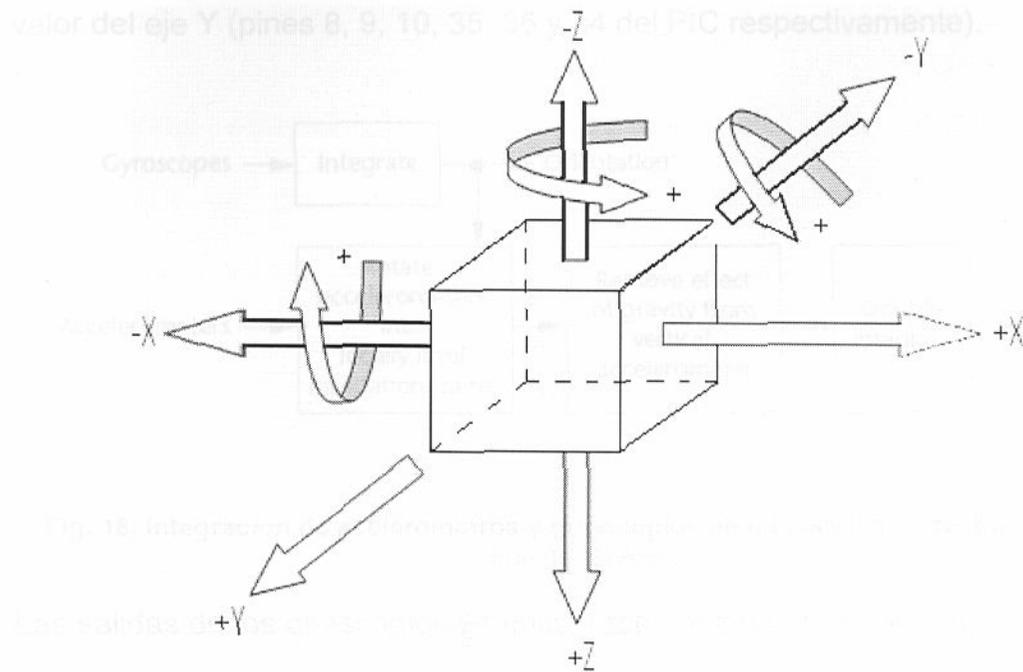


Fig. 17. Ejes de coordenadas locales del dispositivo de rastreo inercial.
Fuente: El Autor

La salida de los dispositivos es relativa al eje de coordenadas locales del cubo, con la salvedad de que el valor en la salida del acelerómetro esta influenciado por la gravedad. Por ello, es necesario seguir las operaciones de la Fig. 17 y rotar los ejes de referencia del giroscopio o del vector de gravedad según los valores obtenidos por los giroscopios para compensar las lecturas del acelerómetro restando la aceleración de gravedad.

Luego se conectaron las salidas de los dispositivos a las entradas digitales del Pic18f4550, respectivamente AN5 para el giroscopio de la Base (eje Z), AN6 para el giroscopio del eje X, AN7 para el giroscopio el eje Y, AN8 para la salida del eje Z del acelerómetro, AN9 para el eje X del acelerómetro y AN10 para el

valor del eje Y (pines 8, 9, 10, 35, 36 y 34 del PIC respectivamente).

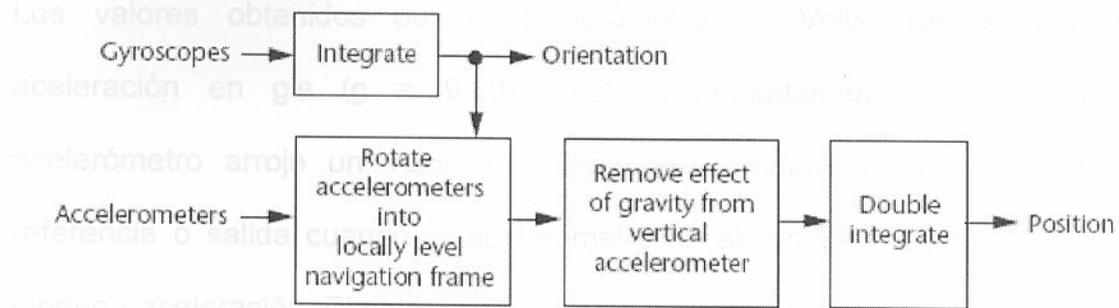


Fig. 18. Integración de acelerómetros y giroscopios en un sistema de rastreo inercial.
Fuente: El Autor.

Las salidas de los giroscopios en volts, representan la tasa de velocidad en que gira el dispositivo sobre sus ejes, esta medida, conociendo la sensibilidad del aparato e integrando el valor se consigue la rotación en grados.

La sensibilidad típica de los giroscopios ADXR300 es de $5 \text{ mv}/^\circ/\text{s}$ en condiciones de operación normales. Para conocer el sentido de la rotación, el voltaje de salida se compara con el voltaje de referencia o voltaje cuando el giroscopio se encuentra estático, normalmente debería ser la mitad del voltaje con que opera el dispositivo 2.5 volts, lamentablemente estos dispositivos sufren de cierto porcentaje de desviación con respecto a su respuesta lineal, por ello es necesario calibrar los giroscopios (hallar el valor de referencia, para ello se realizan un número de muestras sucesivas mientras el giroscopio se encuentra estático y se toma el promedio de ellas como punto de referencia. Si la lectura es menor al punto de referencia, la rotación es en sentido anti-horario, si es mayor que el punto de referencia la rotación es en sentido horario.

$$\text{Rotación } \Omega = ((V_{\text{salida}} - V_{\text{referencia}}) / \text{Sensibilidad}) * \Delta t$$

Los valores obtenidos por el acelerómetro en Volts, representan una aceleración en g's ($g = 9,807 \text{ m/s}^2$ aproximadamente). Igualmente el acelerómetro arroja un valor que debe ser comparado con un punto de referencia o salida cuando el acelerómetro no se encuentra influenciado por ninguna aceleración. El valor de referencia también puede ser calculado en las condiciones mencionadas (estático), de igual forma que se realizó con los giroscopios. Según [Ramstand, 2005] en su trabajo de caracterización de acelerómetros, los acelerómetros Kionix KMX52-1050, presentan valores característicos distintos para el eje Z en sensibilidad y valor de referencia con respecto a los ejes X y Y, Figura 19. Para calibrar el acelerómetro utilizado y obtener los valores de sensibilidad en cada uno de los ejes, se tomaron muestras de la salida al posicionar el giroscopio de forma que la aceleración de gravedad no influya en la muestra, los valores de sensibilidad obtenidos son muy similares a los obtenidos por Ramstand (1070 mVolts/g para los ejes X y Y, 990 mVolts/g para el eje Z). Con estos valores de sensibilidad constantes y el valor de referencia, es posible calcular el valor de la aceleración positiva o negativa sobre el eje, siguiendo un procedimiento similar que el utilizado para el giroscopio.

From characterization test	3,3V	5V	Unit	Proportion rate
Sensitivity X-axis	659,51	1070,95	mV/g	Ratiometric
Sensitivity Y-axis	660,11	1071,27	mV/g	Ratiometric
Sensitivity Z-axis	658,89	978,57	mV/g	Ratiometric
Offset X-axis	1,67401	2,57304	V	Ratiometric
Offset Y-axis	1,64716	2,55354	V	Ratiometric
Offset Z-axis	1,62349	2,57839	V	Ratiometric
Bandwidth, 32kΩ and 0,1μF	49,73592	49,73592	Hz	Unchanged
Noise 2nd order BW, x and y axis	0,2608616	Unspecified	mg	Unlinear
Noise 2nd order BW, z axis	0,4844572	Unspecified	mg	Unlinear
A/D converter:				
Step size	152,59022	152,59022	μV	Ratiometric
Resolution X-axis	0,2313691	0,1424812	mg	Ratiometric
Resolution Y-axis	0,2311588	0,1424386	mg	Ratiometric
Resolution Z-axis	0,2315868	0,1559318	mg	Ratiometric

Fig.19. Caracterización del Acelerómetro KXM50 por Ramstad y Holst-Pedersen.
Fuente: [Ramstad, 2005]

$$A = (V_{\text{valida}} - V_{\text{referencia}}) / \text{Sensibilidad.}$$

Una vez calculado el vector de aceleración, es posible determinar si el dispositivo se encuentra influenciado por una aceleración distinta a la gravedad, siguiendo la relación $1 = \sqrt{Ax^2 + Ay^2 + Az^2}$ y aplicando el Teorema de Pitágoras al triángulo formado por la proyección del vector de aceleración sobre los ejes. La raíz de la suma de los cuadrados es la magnitud del vector de aceleración en g's, si es igual a la unidad el acelerómetro está midiendo simplemente la aceleración de gravedad de la tierra y no se está desplazando. Esta técnica también evita confundir inclinaciones del acelerómetro como una aceleración dinámica sobre los ejes.

Para calcular la distancia desplazada, primero se debe trasladar el vector de

aceleración al sistema de coordenadas del dispositivo. Para ello se multiplica el vector de aceleración por una matriz de rotación, construida con los valores obtenidos por los giroscopios, siguiendo el teorema de Euler sobre rotaciones, donde "una rotación arbitraria puede ser representada por tres rotaciones sucesivas sobre cada uno de los ejes de coordenadas".

Dado α un ángulo de rotación sobre el eje X:

$$R_x(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha) & \text{sen}(\alpha) \\ 0 & -\text{sen}(\alpha) & \cos(\alpha) \end{bmatrix}$$

Dado β un ángulo de rotación sobre el eje Y:

$$R_y(\beta) = \begin{bmatrix} \cos(\beta) & 0 & -\text{sen}(\beta) \\ 0 & 1 & 0 \\ \text{sen}(\beta) & 0 & \cos(\beta) \end{bmatrix}$$

Dado σ un ángulo de rotación sobre el eje Z:

$$R_z(\sigma) = \begin{bmatrix} \cos(\sigma) & \text{sen}(\sigma) & 0 \\ -\text{sen}(\sigma) & \cos(\sigma) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Luego de trasladar el vector mediante las rotaciones sucesivas, se puede sustraer la aceleración de gravedad y calcular el desplazamiento sobre los ejes

utilizando las componentes del vector resultante, donde: $d = \frac{1}{2} * a * t^2$.

longitud de **7.3.1. Firmware del Guante virtual.** positivo, se puede calcular

posición y orientación de la punta del brazo mecánico por trigonometría.
El código programado en el microcontrolador para el guante virtual funciona como dos dispositivos USB distintos, lo que se denomina un "Combo Device", el primer dispositivo sigue las especificaciones HID para Mouse, haciendo uso del acelerómetro para calcular la inclinación de la mano, se puede mover el cursor sobre el escritorio y utilizar los dedos índice y medio como botón derecho e izquierdo del Mouse virtual. Es posible activar y desactivar el Mouse enviando un reporte al dispositivo.

La funcionalidad principal, además del Mouse virtual, consiste en enviar un reporte de 22 bytes al computador de los valores convertidos por el ADC para los cinco dedos y los seis valores que conforman el dispositivo de rastreo, utilizando 2 bytes para cada valor. El código fuente de estas funciones y los descriptores asociados para este dispositivo se encuentran en el Anexo 10.

7.4. Construcción del dispositivo de rastreo de posición y orientación para la cabeza.

El diseño de este dispositivo sigue el modelo de los dispositivos de rastreo mecánicos. Se construyó en madera rígida y liviana, colocando potenciómetros comunes de $20k\Omega$ en los puntos articulares (puntos de rotación) sobre los que giran las piezas, los potenciómetros tienen una relación de $20m\Omega$ por grado, pueden rotarse un máximo de 300° . Con los valores límites (máximo y mínimo, ángulo / voltaje) que alcanza cada potenciómetro, sus características y la

longitud de las partes que conforman el dispositivo, se puede calcular la posición y orientación de la punta del brazo mecánico por trigonometría. La Fig. 18 muestra un diagrama simple del dispositivo.

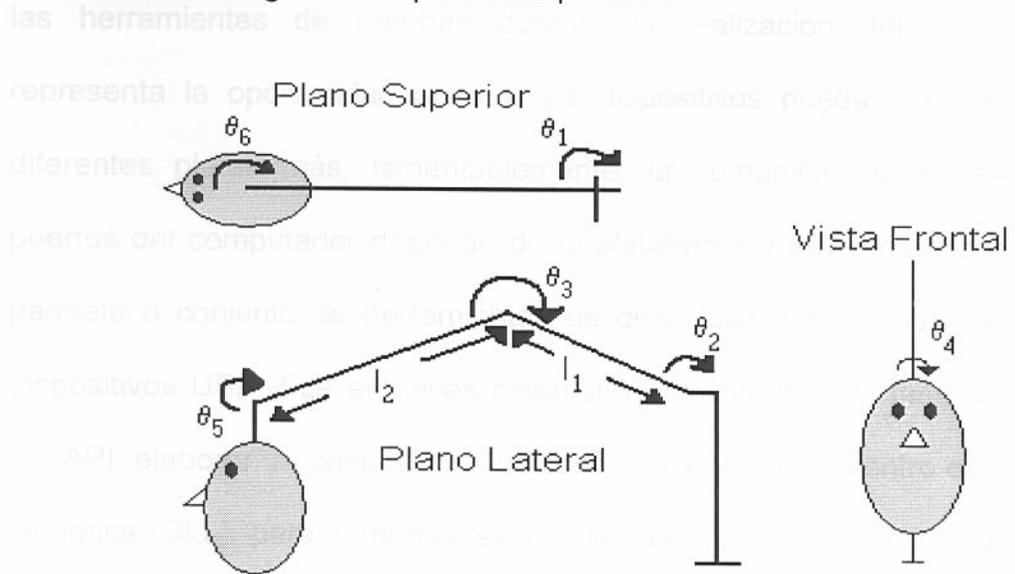


Fig. 20. Esquema simplificado del sistema de rastreo mecánico construido.
Fuente: El Autor

Seis potenciómetros son utilizados para calcular los 6 grados de libertad de la cabeza, el conjunto de los 3 primeros potenciómetros (contando desde la base del dispositivo) son utilizados para calcular la posición de la cabeza en el espacio y los otros tres rastrean la orientación de la misma. Los potenciómetros están alimentados por una diferencia de potencial de 5Volts suministrada por el puerto USB. Las salidas de los potenciómetros están conectadas a entradas analógicas del convertidor A/D de un microcontrolador Pic18f4550.

7.5 Diseño y desarrollo del software de control y API.

Se escogió JAVA como lenguaje de programación para el desarrollo del API y las herramientas de pruebas durante la realización del proyecto. Java representa la oportunidad de que los dispositivos puedan ser utilizados en diferentes plataformas, lamentablemente, la comunicación a través de los puertos del computador depende de la plataforma y Java todavía no tiene un paquete o conjunto de herramientas de desarrollo para comunicarse con los dispositivos USB. Fue entonces necesario, para la primera fase de desarrollo del API, elaborar un conjunto de funciones en lenguaje C dentro de una librería dinámica (DLL), para comunicarse con los dispositivos y envolver (*wrap*) estas funciones en una clase de Java utilizando JNI (*Java Native Interface*). Las funciones desarrolladas en la librería conforman un conjunto mínimo necesario para establecer comunicación con dispositivos USB-HID (*Human Input Device*) específicamente.

Como se mencionó anteriormente, la comunicación a través del puerto USB se realizó bajo la supervisión del módulo *host* (servidor) en el cual los dispositivos están conectados. Por esto, la comunicación no se realiza directamente con los dispositivos, sino por medio de dicho *host*. Lo primero que se debe hacer es preguntar al *host* si el dispositivo está conectado y obtener de él la dirección del este, con esta dirección se puede leer y escribir información en el *buffer* compartido entre el dispositivo y el *host*. Las funciones básicas implementadas en la librería de comunicación pueden observarse en el *Javadoc* generado para

la clase *DLLWrapper* que funciona como nexo entre la aplicación creada en Java y la librería, Anexo 12, entre ellas se encuentra “preguntar” por la dirección de un dispositivo, dado el código del fabricante y su código de producto, si este se encuentra conectado, enviar y recibir datos del dispositivo dada una dirección válida.

El software elaborado en Java, se encarga de monitorear la conexión y desconexión de los dispositivos, decodificar la información recibida, realizar los cálculos de posición y orientación, haciéndolos disponibles para el usuario de forma clara, para que puedan ser utilizados luego en otras aplicaciones como representaciones tridimensionales, realidad virtual, etc.

El software esta compuesto por un paquete de clases que representan los dispositivos y sus componentes, estas se encargan de realizar los cálculos necesarios para obtener la posición y orientación. El API esta conformado por un hilo de control llamado *USBController* que monitorea el puerto USB y se encarga de intercambiar información con los dispositivos, haciendo uso de las funciones desarrolladas en lenguaje C a través de una clase estática llamada *DLLWrapper*. Un segundo hilo llamado *MainLoop*, se encarga de decodificar la información recibida por el *USBController* y alimentar el conjunto de clases que conforman los dispositivos. La información obtenida por el *USBController*, es puesta a disposición del *MainLoop* a través de un conjunto de clases que emulan un canal de comunicación USB, haciendo uso de una variante del algoritmo productor/consumidor (*USBPipe* y *USBPacket*), la variante tiene como

condición que se pueden perder paquetes para evitar retrasos en los resultados mostrados al usuario. Nuevos datos están disponibles para *USBController* por parte del dispositivo cada 10ms y estos se hacen disponibles al hilo principal con aproximadamente la misma frecuencia. El diagrama de clases del software se encuentra en el Anexo 13.

Los usuarios pueden utilizar el API implementando la interfaz *MainInterface* que contiene el enunciado de las funciones necesarias para comunicarse con el hilo principal y el *USBController*, también es necesario instanciar y ejecutar ambos hilos de control en la misma clase que implementa la Interfaz, un ejemplo de uso del API se encuentra en el Anexo 14.

8. Resultados.

Los objetivos de este trabajo especial de grado fueron alcanzados con gran satisfacción, creando una base para futuros desarrollos e investigaciones sobre los dispositivos de realidad virtual construidos, así como las tecnologías utilizadas y sus aplicaciones.

Los dispositivos se desarrollaron utilizando el puerto USB como interfaz y protocolo de comunicación con el computador, aun cuando esto no estaba definido en el alcance del proyecto, lo cual representa una innovación y una oportunidad para futuros desarrollos de periféricos.

Se elaboró un guante de realidad virtual que mide eficazmente el grado de flexión de cada uno de los dedos de la mano derecha, aunque no es capaz de medir la separación entre los dedos. Esta medida se podría obtener implementado un sistema similar al utilizado en cada uno de los dedos, colocando sensores de flexión en los pliegues interdigitales de la mano.

Se construyó un sistema de rastreo de posición y orientación inercial para el guante virtual que brinda mayor libertad al usuario y un sistema de rastreo electromecánico para la cabeza.

Se desarrollo un software para utilizar los dispositivos en forma conjunta o independiente, brindando al usuario la opción de utilizar los dispositivos y el paquete de control en otras aplicaciones. La portabilidad del API esta limitada únicamente por la librería de comunicación con el puerto USB que fue

desarrollada en lenguaje C para maquinas con Sistema operativo Windows, aunque solo es necesario reimplementar una clase en JAVA para poder utilizar cualquier otra librería de comunicación USB en los sistemas operativos y plataformas donde se pueda ejecutar programas en JAVA.

El único dispositivo que no fue desarrollado completamente es el casco de realidad virtual (HMD), el cual no dispone de la pantalla LCD, pero durante los últimos 2 años se ha extendido el uso de este tipo de pantallas en diversas aplicaciones y para varios sistemas de entretenimiento portátil como televisores en el automóvil, reproductores de DVD portátiles, entre otros. Estas nuevas pantallas permiten, hoy en día, una conexión directa con las salidas de video comunes encontradas en los computadores, son cada vez más pequeñas, livianas y económicas. Lamentablemente estos equipos son importados y durante el desarrollo del trabajo no se pudo adquirir por el control cambiario vigente a la fecha.

9. Conclusiones y Recomendaciones.

El diseño y construcción de dispositivos de realidad virtual y periféricos para computadores, es una tarea multidisciplinaria, ya que no solo se requieren conocimientos en programación y arquitectura del computador. Es por ello que no es recomendable embarcarse solo en esta tarea sin asesoramiento calificado.

Los giroscopios y acelerómetros utilizados en este proyecto, a pesar de ser fabricados utilizando lo último en tecnología, presentan, en la mayoría de los casos pequeños defectos de fabricación que desvían su funcionamiento de los parámetros normales mencionados en las especificaciones del producto. Por ello, es necesario realizar una caracterización rigurosa de los dispositivos o implementar algoritmos de calibración antes y/o durante el uso de los dispositivos.

El desarrollo de dispositivos USB específicos es una tarea compleja, desde el diseño de su arquitectura hasta el desarrollo de los controladores específicos para un sistema operativo determinado. Las clases específicas de dispositivos USB como la clase HID, brindan un patrón de diseño que puede ser adaptado por los sistemas operativos como en el caso de Windows y algunas versiones de Linux, de manera que los desarrolladores pueden usar los controladores del sistema operativo, ahorrando tiempo y recursos de desarrollo.

El uso de acelerómetros para obtener el desplazamiento no es la mejor opción cuando se requiere gran precisión, ya que los pequeños errores en esta se reflejan en mayor magnitud al integrar dos veces la aceleración.

Para aquellas personas que deseen continuar el desarrollo de estos dispositivos o simplemente mejoras sobre ellos, se recomienda, en el caso del Guante, agregar o reemplazar la conexión USB con el computador por una conexión *Bluetooth* que permita al usuario moverse con mayor libertad.

En el caso del dispositivo de rastreo electromecánico, se recomienda cambiar el potenciómetro situado en la base, por un sistema de poleas para evitar cargar con todo el peso del aparato sobre el potenciómetro y evitar así que este se pueda romper, además de cambiar el materia que conforma el brazo por algo más ligero e igual o más rígido para evitar la flexión de las piezas.

Fig. 1. Convergencia de los ejes ópticos de la vista. 15

Fig. 2. Fusión de imágenes en el cerebro (estereoscopia). 16

Fig. 3. Estereoscopio de Sir Charles Wheatstone. 17

Fig. 4. Head Mounted Display. 19

Fig. 5. Guante P5 de Essential Reality. 22

Fig. 6. Sistema force feedback para la mano por Immersion Ltd. 24

Fig. 7. Sistema de rastreo mecánico de cuerpo completo, Gipsy 4 de Metamotion. 26

Fig. 8. Problema del caminante perdido. 28

Fig. 9. Funcionamiento interno de los acelerómetros por diferencia de capacitancia. 30

Fig. 10. Sistema de rastreo ultrasónico desarrollado por Logitech. 32

Fig. 11. Prototipo de rastreo por GPS, departamento de defensa de USA 34

Fig. 12. Jerarquía de los descriptores de un dispositivo USB. 41

Fig. 13. Metodología de cascada iterativa mejorada. 46

Fig. 14. Circuito base del pic18f4550 con conexión USB.....	48
Fig. 15. Diagrama del Chip PIC18F4550.....	51
Fig. 16. Conexión de la fotorresistencia y el LED para medir la flexión de los dedos.....	58
Fig. 17. Ejes de coordenadas locales del dispositivo de rastreo inercial.....	60
Fig. 18. Integración de acelerómetros y giroscopios en un sistema de rastreo inercial.....	61
Fig.19. Caracterización del Acelerómetro KXM50 por Ramstad y Holst- Pedersen.....	63
Fig. 20. Esquema simplificado del sistema de rastreo mecánico construido.....	66

Bibliografía.

- [Axelson, 2005] Axelson, J. (2005). USB Complete (3ra. Edición). USA. Lakeview Research.
- [Calligo, 2003] Calligo, Aníbal, Marcelo Schenone. (2003). Técnicas de Producción de Software I – Modelos de Proceso de Desarrollo, Facultad de Ingeniería, UBA.
- [Case, 1997] Case, Hovden T. (1997). Electronic Pointing Apparatus and Method. Unite States Patentes Office. Pat# 5, 825,350
- [Donald, 1994] Donald, Baker P. (1994). Graficas por Computadora (2da. Edición). Prentice-Hall.
- [Dorf, 1995] Dorf. (1995). Circuitos Eléctricos: Introducción al Análisis y Diseño (2da. Edición). México. AlfaOmega.
- [Fishbane, 1994] Fishbane, Gasiorowicz, Thornton, (1994). Fisica Para Ciencias e Ingenieria (Volumen I, Capítulos II, III, IV, V, IX y X). Prentice-Hall.
- [Grossman, 2004] Grossman, Cooper M. (2004). Magic Eye: The 3-D Guide. USA. N.E. Thing Enterprise Inc.
- [Hollands, 1996] Hollands, R. (1996). The Virtual Reality Homebrewer's Handbook. England. John-Wiley & Sons.
- [Krammer, 1990] Krammer, George, Lindener P. (1990). Strain-Sensing

Goniometers, System and Recognition Algorithms. Unite States Patent Office.

Pat# 5, 280,265

[Larman, 2003] Larman, Craig. (2003). Agile and Iterative Development: A Manager's Guide. Addison-Wesley Agile Software Development Series.

[Luinge, 2002] Luinge H. (2002). Inertial Sensing of Human Movement. Holanda. Twente University Press.

[Ramstad, 2005] Ramstad, Holst-Pedersen H. (2005). Characterization of Accelerometers in LabView, Proyecto Principal en Micro tecnología, Vestfold University College, Noruega.

[Rosenberg, 1996] Rosenberg, Schena, Brave y Jackson B. (1996). Method for producing a precision 3D apparatus. Unite States Patents Office. Pat# 6,015,473

[Royce, 1970] Royce, W. (1970). Managing the Development of Large Software Systems. Proceedings of IEEE.

[Welch, 2002] Welch, Foxlin E. (2002). Motion Tracking: No Silver Bullet but a Respectable Arsenal. IEEE.

[Xavier, 2002] Xavier, F. (2002). El Bus USB. España. Paraninfo.

[Zimmerman, 1982] Zimmerman, T. (1982). Optical Flex Sensor. Unite States Patents Office. Pat# 4, 542,291.

Bibliografía en línea.

[ADC, 2001] Microchip Technologies Inc. (2001). Getting Started – Analog to Digital Converters [PDF], Disponible en: <http://www.microchip.com>.

[ANALOG, 2004] Analog Devices Inc. (2004), New iMEMS Angular-Rate-Sensing Gyroscope [PDF], Disponible en: <http://www.analog.com>

[ANALOG-APP, 2004] Analog Devices Inc. (2004), Accelerometer Design and Applications [PDF], Disponible en: <http://www.analog.com>

[ANALOG-SPEC, 2004] Analog Devices Inc. (2004), ADXRS300 Rate Gyro Chip Data Sheet [PDF], Disponible en: <http://www.analog.com>

[C18, 2005] Microchip Technologies Inc. (2005). MPLAB C18: C Compiler User's Guide [PDF], Disponible en: <http://www.microchip.com>, code: DS51288G

[C18LIB, 2005] Microchip Technologies Inc. (2005). MPLAB C18: C Compiler Libraries [PDF], Disponible en: <http://www.microchip.com>, code: DS51297D

[EVL, 2005] Electronic Visualization Laboratory, University of Illinois at Chicago (2005). Virtual and Augmented Reality, [Internet], Disponible en: http://www.cs.uic.edu/~i422/VRandAR_2.html, [Febrero, 2005]

[GONZALES, 2004] Gonzáles A. (2004), 3D Stereo Web [Internet], Disponible en: <http://www.users.red3i.es/~stereoweb/index.htm> [Octubre, 2005]

[GYRATION, 2004] Gyration Inc. (2004). Application Notes for GyroPoint

Technology [PDF]. Disponible en: <http://www.gyration.com/>, code: DE00053-001

[KIONIX-MAN, 2004] Kionix Inc. (2004), Kionix Demo User's Manual [PDF], Disponible en: <http://www.kionix.com>

[KIONIX-SPEC, 2005] Kionix Inc. (2005), KXM52 Series Data Sheet [PDF], Disponible en: <http://www.kionix.com>

[KIONIX-TECH, 2004] Kionix Inc. (2004), KXM52 Technical Notes [PDF], Disponible en: <http://www.kionix.com>

[MELLOTS, 2004] Mellot's K. (2004). Mellot's VR Page [Internet], Disponible en: <http://www.geocities.com/mellott124/>, [Octubre, 2005]

[PEACOCK, 2002] Peacock, C. (2002). USB in a Nutshell: Making Sense of the USB Standard (3ra. Edicion). [PDF], Disponible en: <http://www.beyondlogic.org>

[PIC18, 2004] Microchip Technologies Inc. (2004). PIC18F4550 Data sheet [PDF], Disponible en: <http://www.microchip.com>, code: DS39632B

[PIC18C, 2000] Microchip Technologies Inc. (2000). PICmicro 18C MCU Family Referente [PDF], Disponible en: <http://www.microchip.com>, code: DS39500A

[SILICON, 2005] Silicon Sensing Systems – Japan, Ltd. (2005) Frequently asked questions on rate gyros. [Internet] Disponible en: <http://www.spp.co.jp/sssj/qanda-e.html> [Octubre, 2005]

[TEXAS, 2005] Texas Instruments Inc. (2005), Accelerometers and How they

Work [PDF], Disponible en: <http://www.texasinstruments.com>

[USB-FIRMWARE, 2002] Microchip Technologies Inc. (2002). USB Firmware User's Guide [PDF], Disponible en: <http://www.microchip.com>

[USB-HID, 2001] USB Implementer's Forum. (2001). Device Class Definition for Human Interface Devices (HID) [PDF], Disponible en: <http://www.usb.org/developers/docs>.

[USB-PID, 1999] USB Implementer's Forum. (1999). Device Class Definition for Physical Interface Devices (PID) [PDF], Disponible en: <http://www.usb.org/developers/docs>.

[USB-SPEC, 2000] Compaq, Hewlett-Packard, Intel, Lucent, Microsoft, Nec, Philips. (2000). Universal Serial Bus Specification [PDF], Disponible en: <http://www.usb.org/developers/docs>.

[VR-NAVY, 2000] US NAVAL Postgraduate school (2000). Ultrasonic, Inertia and Hybrid Tracking Technologies, [Internet], Disponible en: <http://www.cs.nps.navy.mil/people/faculty/capps/4473/projects/chang2/Full.htm>, [Octubre, 2005]

[WEIMER, 2004] Weimer M. (2004). Moving Average Algorithm, [Internet] <http://www.dataq.com/applicat/articles/an14.htm>, DataQ Instruments, [Noviembre, 2004]