



PONTIFICIA UNIVERSIDAD CATOLICA DE CHILE  
ESCUELA DE INGENIERIA

# **HEAD TRACKING EN EL ESPECTRO INFRARROJO UTILIZANDO EL NINTENDO WII REMOTE**

**MAURICIO JAVIER UBILLA CARVAJAL**

Tesis para optar al grado de  
Magíster en Ciencias de la Ingeniería

Profesor Supervisor:  
**DOMINGO MERY Q.**

Santiago de Chile, Diciembre, 2009

© 2009, Mauricio Javier Ubilla Carvajal.



PONTIFICIA UNIVERSIDAD CATOLICA DE CHILE  
ESCUELA DE INGENIERIA

# **HEAD TRACKING EN EL ESPECTRO INFRARROJO UTILIZANDO EL NINTENDO WII REMOTE**

**MAURICIO JAVIER UBILLA CARVAJAL**

Tesis presentada a la Comisión integrada por los profesores:

**DOMINGO MERY Q.**

**MARCELO GUARINI H.**

**PATRICIO DE LA CUADRA B.**

**MIGUEL RÍOS O.**

Para completar las exigencias del grado de  
Magíster en Ciencias de la Ingeniería

Santiago de Chile, Diciembre, 2009

A mis Padres, quienes siempre me han apoyado y me han entregado lo mejor de ellos.

## **AGRADECIMIENTOS**

Son muchas las personas que me ayudaron a desarrollar este trabajo y a quienes quiero agradecer su apoyo.

En primer lugar y por sobre todo, a mi profesor supervisor, Domingo Mery, por haberme guiado durante todo el proceso, y sin cuyo apoyo, consejos y revisión este trabajo no habría sido posible.

A los profesores: Rodrigo Cádiz, por sus indicaciones y su excelente disposición para revisar el trabajo, y Patricio de la Cuadra, por sus valiosos consejos.

Finalmente a mis padres, por su apoyo incondicional, en especial a mi padre, Jaime Ubilla, por su ayuda en la construcción de varios prototipos del cintillo.

## INDICE GENERAL

|  | Pág. |
|--|------|
| DEDICATORIA.....   | ii   |
| AGRADECIMIENTOS .....                                      | iii  |
| INDICE DE TABLAS .....                                     | vi   |
| INDICE DE FIGURAS .....                                    | vii  |
| RESUMEN.....   | viii |
| ABSTRACT .....   | ix   |
| 1. Introducción .....                                      | 10   |
| 2. Sistema Propuesto .....                                 | 16   |
| 2.1 Reconstrucción 3D con una vista y 3 puntos .....       | 16   |
| 2.1.1 Geometría proyectiva básica .....                    | 16   |
| 2.1.2 Reconstrucción 3D a partir de 3 puntos .....         | 17   |
| 2.1.3 Newton-Raphson aplicado al problema .....            | 20   |
| 2.2 Head Tracking básico .....                             | 22   |
| 2.3 Head Tracking extendido.....                           | 25   |
| 2.3.1 Implementación del sistema .....                     | 27   |
| 2.3.2 Módulo de Tracking .....                             | 29   |
| 2.3.3 Módulo de Reconstrucción.....                        | 36   |
| 2.4 Eliminación de la inclinación de la cámara .....       | 39   |
| 2.4.1 Determinación del pitch y roll de la cámara .....    | 41   |
| 2.4.2 Corrección de la reconstrucción 3D del cintillo..... | 42   |
| 3. Experimentos y Resultados .....                         | 44   |
| 3.1 Precisión.....   | 44   |
| 3.2 Rapidez .....  | 46   |
| 3.3 Robustez.....  | 47   |
| 4. Conclusiones .....                                      | 48   |

|  |    |
|--|----|
| BIBLIOGRAFIA.....                                  | 51 |
| A N E X O S.....                                   | 53 |
| Anexo A: El cintillo y los diodos utilizados ..... | 54 |
| Anexo B: Calibracion de los acelerómetros .....    | 56 |

## INDICE DE TABLAS

|   | Pág. |
|---|------|
| Tabla 3.1-1: Error teórico de las reconstrucciones..... | 45   |

## INDICE DE FIGURAS

|  | Pág. |
|--|------|
| Figura 1-1: Dispositivos utilizados. ....                                      | 13   |
| Figura 2.1-1: Proyección de un punto en el modelo de cámara Pinhole.....       | 17   |
| Figura 2.1-2: Distancias entre $M_1$ , $M_2$ y $M_3$ .....                     | 18   |
| Figura 2.2-1: Esquema del sistema de HT básico (vista superior).....           | 22   |
| Figura 2.2-2: Distancias importantes para el sistema de HT. ....               | 23   |
| Figura 2.3-1: Esquema del sistema de HT extendido (vista superior). ....       | 25   |
| Figura 2.3-2: Fotografías del cintillo para HT extendido con y sin luz. ....   | 26   |
| Figura 2.3.1-1: Esquema del proceso de HT.....                                 | 28   |
| Figura 2.3.2-1: Casos posibles en una asignación con referencia.....           | 34   |
| Figura 2.3.3-1: Reconstrucción del cintillo.....                               | 37   |
| Figura 2.4-1: Corrección de la inclinación de la cámara. ....                  | 39   |
| Figura 2.4-2: Disposición de los acelerómetros en el Nintendo Wii Remote.....  | 40   |
| Figura 2.4.1-1: Observando la aceleración de gravedad. ....                    | 41   |
| Figura 3-1: Capturas de pantalla de la aplicación para 4 poses distintas. .... | 44   |
| Figura A-1: Angulo de emisión de los diodos utilizados.....                    | 54   |



## **RESUMEN**

Los sistemas de Head Tracking con estimación de pose tienen muchas aplicaciones en interacción humano-computador. Sin embargo, la mayor parte de los sistemas desarrollados a la fecha están restringidos a un rango de poses en que el usuario debe estar de frente a la cámara, son sensibles a las condiciones de iluminación y no son invariantes a la identidad del usuario. En este trabajo se propone un sistema de Head Tracking que estima la pose de la cabeza con precisión, utilizando el Nintendo Wii Remote para trackear un cintillo con diodos emisores infrarrojos. Este sistema no restringe al usuario a permanecer mirando a la cámara, es menos sensible a las condiciones de iluminación y completamente invariante a la identidad del usuario. Adicionalmente, el sistema elimina la inclinación de la cámara utilizando la información de los acelerómetros del dispositivo.

Palabras Claves: head tracking, head pose estimation, human-computer interaction

## **ABSTRACT**

Head Tracking systems with head pose estimation have several applications in human-computer interaction. However, most developed systems to date are restricted to a range of poses in which the user has to be facing the camera, are sensible to the lightning conditions and are not identity-invariant. In this work we propose a Head Tracking system with accurate head-pose estimation in the infrared spectrum that uses the Nintendo Wii Remote to track a circlet with infrared diodes. The system doesn't restrict the user to be facing the camera, is less sensitive to the lighting conditions and is completely identity-invariant. Additionally, the system eliminates the camera inclination using the accelerometer data of the device.

Keywords: head tracking, head pose estimation, human-computer interaction

## 1. INTRODUCCIÓN

Head Tracking (HT) se refiere al proceso de detectar la cabeza de una persona en una secuencia de video, determinando idealmente su posición y orientación en el espacio. De este modo, existen distintos niveles de HT, que van desde la simple detección de la cabeza, conseguida generalmente mediante *Face Detection* (Hjelmas y Low, 2001) hasta la estimación de su orientación o pose (tópico también conocido como *Head Pose Estimation*, Murphy-Chutorian y Trivedi, 2009).

HT es un tema bastante estudiado en los últimos 15 años, principalmente debido a:

- El gran interés que generan sus aplicaciones, fundamentalmente en Human-Computer Interaction (HCI). Durante los últimos 20 años las personas han interactuado con los computadores principalmente a través de un teclado y un mouse. HCI es un área de investigación que busca desarrollar nuevas formas de interacción. HT forma parte de un conjunto de estas nuevas formas de interacción, conocido como Vision-Based Interfaces (VBIs, Porta, 2009) .
- La masificación de las webcams, lo cual hizo que esta tecnología sea más accesible y que los sistemas desarrollados puedan ser aprovechados por cualquier persona.

Se han propuesto muchas aplicaciones de HT (Porta, 2009):

1. Scrolling: una operación típica en interfaces de usuario graficas (GUIs) comunes, es realizar scrolling sobre un contenido que no cabe en una ventana. Se puede utilizar la dirección de la cabeza para realizar scrolling en esa dirección.
2. Zooming - Magnification: otra operación típica, similar a la anterior, es realizar zooms para acercar o amplificar contenidos. Se puede utilizar la distancia de la cabeza con respecto a la pantalla para controlar estas operaciones.

3. Pointing: de manera similar a la primera aplicación, se puede utilizar la dirección de la cabeza para reemplazar la función de puntero del mouse. Esta aplicación es especialmente útil para usuarios con discapacidades.
4. 3D Navigation: utilizando una funcionalidad parecida a las aplicaciones de scrolling y zooming se puede utilizar movimientos de la cabeza para navegar por ambientes virtuales como los de videojuegos 3D.
5. Head Gestures: reconociendo gestos típicos de la cabeza como nod and shaking se puede aceptar o cancelar en los diálogos de GUIs comunes. Otros gestos similares pueden ser utilizados para navegar por páginas web, cambiar canciones en un reproductor de media, o avanzar páginas en un documento.

Todas estas aplicaciones no sólo representan formas más intuitivas de interactuar con los computadores para personas normales, sino que además constituyen formas posibles de interactuar para personas con discapacidades.

La mayoría de las técnicas de HT desarrolladas intentan estimar la orientación de la cabeza a partir de una única vista. Estas pueden clasificarse en dos categorías (Porta, 2009):

- Appearance-based methods: obtienen la información directamente de las imágenes adquiridas sin intentar construir una representación 3D de la cabeza (Bérard, 1999, y Toyama, 1998).
- Model-based methods: utilizan un modelo 3D para representar la cabeza, típicamente cilíndrico (La Cascia et al, 2000, Ryu y Kim, 2007) o elipsoidal (An y Chung, 2008). Las imágenes son utilizadas para estimar los 6 parámetros que definen la posición y orientación del modelo.

También se han desarrollado otras técnicas utilizando stereo-visión para obtener mejores resultados (Morency et al, 2002, Yang y Zhang, 2002). Sin embargo, todos estos sistemas comparten la limitación de requerir que el usuario deba mirar de frente a la cámara, para poder detectar su rostro.

Una aplicación interesante de HT es conseguir audio espacial virtual a través de audífonos, utilizando HRIRs (Head Related Impulse Responses, Mohan et al, 2003). Las HRIRs son señales que indican cómo el oído externo, en conjunto con la cabeza y el torso atenúan un sonido proveniente de una posición determinada antes de llegar a cada oído. Convolucionando un sonido monofónico con la HRIR de cada oído correspondiente a una posición de interés, es posible obtener una versión estéreo del sonido, que reproducido a través de audífonos se escucha como si proviniese de dicha posición. Esta tecnología es conocida Audio espacial o Audio 3D.

Integrando un sistema de Audio espacial con un sistema de HT que determine la orientación de la cabeza del usuario, es posible espacializar sonidos en tiempo real a través de audífonos, consiguiendo audio espacial “virtual”, la versión en sonido de la realidad virtual. En una aplicación como ésta, es de especial interés que el sistema de HT no restrinja la orientación de la cabeza del usuario.

Mohan et al utilizaron un sistema de HT en el espectro infrarrojo (IR) en que aplica un algoritmo de Object Pose (DeMenthon y Davis, 1995), para determinar la posición de 4 emisores IR montados sobre los audífonos. Este sistema otorga mayor libertad a la orientación de la cabeza, pero tiene dificultades en diferenciar algunas poses.

Se observa de la revisión bibliografía que los sistemas de HT desarrollados comparten varias limitaciones, entre otras:

- Están restringidos a poses en las que el usuario esté de frente a la cámara por trabajar con características faciales.
- Son sensibles a las condiciones de iluminación del ambiente y a las características físicas del usuario.
- Desconocen el efecto que pudieran tener pequeñas inclinaciones de la cámara en la determinación de la inclinación de la cabeza.

En este trabajo se propone un sistema de HT en el espectro IR que utiliza el Nintendo Wii Remote (Lee, 2008) para detectar un cintillo con diodos emisores IR. El sistema permite determinar la posición y la orientación de la cabeza en forma rápida y robusta para una amplia variedad de poses, gracias a que el cintillo posee diodos en todo su perímetro. Adicionalmente, el sistema detecta y elimina la inclinación de la cámara utilizando la información entregada por los acelerómetros incorporados en el Wii Remote.



Figura 1-1: Dispositivos utilizados.

La figura 1.1.a muestra el Wii Remote, el control principal de la consola Nintendo Wii. Este incorpora entre otras cosas, una cámara IR PixArt de alta resolución en el frente (ver círculo) y un acelerómetro ADLX330 que mide aceleraciones en los 3 ejes.

La figura 1.1.b muestra como funciona el sistema de HT. El Wii Remote se conecta al computador vía Bluetooth y se ubica frente al usuario para detectar el cintillo con LEDs IR ubicado en su cabeza.

Se escogió utilizar este enfoque en el espectro IR utilizando el Wii Remote por las siguientes razones:

1. Las características de la cámara incorporada en el Wii Remote. Esta cuenta con una resolución de 1024x768 píxeles y una tasa de refresco de 100 Hz, muy superiores a las webcams de precios similares, que ofrecen resoluciones de 640x480 a 30 Hz. La resolución de la cámara incide directamente en la precisión del sistema.
2. IR: Trabajar en el espectro IR simplifica mucho la implementación del sistema al prácticamente no requerir pre-procesamiento de la imagen, hace el sistema menos sensible a las condiciones de iluminación del ambiente, y completamente independiente a las características físicas del usuario.
3. Los acelerómetros incorporados en el Wii Remote. Estos permiten determinar la inclinación del dispositivo cuando se encuentra en reposo, y eliminarla para determinar la inclinación absoluta de la cabeza con respecto al plano terrestre.

El sistema desarrollado corre a 100 Hz y determina la posición de la cabeza con un error promedio de 3 mm cuando el usuario se encuentra a 1 m de la cámara. Este error aumenta a 1 cm a 2 m de distancia. Admite rotaciones de yaw y roll de cualquier tipo, pero está restringido a rotaciones de pitch en el rango de  $\pm 30^\circ$ . El error promedio de la estimación de los ángulos de rotación es inferior a  $1^\circ$  incluso a 2 m de la cámara.

El sistema de HT fue implementado en C# utilizando la Managed Library for Nintendo's Wiimote de Brian Peek. Adicionalmente, el sistema permite transmitir la posición y la orientación de la cabeza, a través del protocolo OSC (Open Sound Control, Wright y Freed, 1997) para ser utilizados como input por otras aplicaciones.

OSC es un formato creado para encapsular y enviar datos entre computadores, sintetizadores de sonido y otros dispositivos multimedia. Cualquier programa que entienda OSC puede recibir los datos del sistema de HT y utilizarlos a su gusto, por ejemplo, para controlar algún aspecto de la aplicación. Para la conexión OSC se utilizó la librería Bespoke OSC de Bespoke Software.

El resto del documento se estructura como sigue. En la sección 2 se describe en detalle el sistema de HT propuesto. En la sección 3 se comenta los resultados del sistema, en términos de precisión, rapidez y robustez. Finalmente, en la sección 4 se resume el sistema propuesto, mencionando sus fortalezas y debilidades, y proponiendo posibles mejoras.



## 2. SISTEMA PROPUESTO

En este capítulo se describe el sistema de HT propuesto. La sección 2.1 muestra cómo reconstruir la posición de tres puntos en el espacio a partir de sus proyecciones en una imagen. La sección 2.2 muestra cómo aplicar este método para realizar HT en el espectro IR utilizando un cintillo con sólo 3 diodos IR. La sección 2.3 presenta el sistema propuesto, que extiende el sistema de la sección 2.2 utilizando un cintillo con múltiples diodos. Finalmente, en la sección 2.4 se muestra cómo utilizar los acelerómetros del Wii Remote para detectar la inclinación del dispositivo y eliminar la inclinación de la cámara en el sistema de HT.

### 2.1 Reconstrucción 3D con una vista y 3 puntos

En esta sección se describe el método utilizado para reconstruir la posición en el espacio de los diodos a partir de sus proyecciones en una imagen. Como se verá mas adelante, basta con contar con 3 puntos cuyas distancias relativas sean conocidas, para poder reconstruir sus posiciones originales.

#### 2.1.1 Geometría proyectiva básica

Se considera un modelo clásico de cámara Pinhole como el de la figura 2.1.1. En este modelo, el sistema de referencia 3D o *sistema de la cámara*, está ubicado en su centro óptico  $O$ . El plano de la imagen está ubicado a una distancia  $f$  de  $O$ , conocida como *distancia focal*. El eje  $Z$  del sistema de la cámara coincide con el eje óptico, y sus ejes  $X$  e  $Y$  son paralelos con los ejes  $x$  e  $y$  del sistema de la imagen.

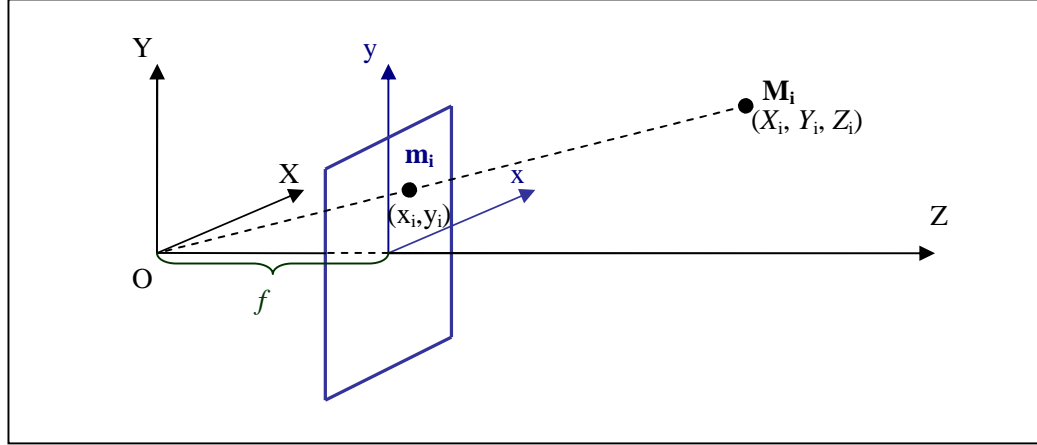


Figura 2.1-1: Proyección de un punto en el modelo de cámara Pinhole.

Un punto  $\mathbf{M}_i$  de coordenadas  $(X_i, Y_i, Z_i)$  en el espacio se proyecta como un punto  $\mathbf{m}_i$  de coordenadas  $(x_i, y_i)$  en la imagen. Las coordenadas de  $\mathbf{m}_i$  están determinadas por las coordenadas de  $\mathbf{M}_i$  y por la distancia focal  $f$ :

$$\begin{aligned} x_i &= X_i f / Z_i \\ y_i &= Y_i f / Z_i \end{aligned} \quad (2.1.1)$$

El problema inverso, de obtener las coordenadas de  $\mathbf{M}_i$  a partir de  $\mathbf{m}_i$ , está subdeterminado. Al despejar  $X_i$ ,  $Y_i$  y  $Z_i$  de (2.1.1), queda un parámetro libre:

$$\begin{aligned} X_i &= Z_i x_i / f \\ Y_i &= Z_i y_i / f \\ Z_i &= Z_i \end{aligned} \quad (2.1.2)$$

### 2.1.2 Reconstrucción 3D a partir de 3 puntos

Sean  $\mathbf{M}_1$ ,  $\mathbf{M}_2$  y  $\mathbf{M}_3$  tres puntos en el espacio de coordenadas desconocidas  $(X_i, Y_i, Z_i)$ ,  $i=1..3$ , pero cuyas distancias son conocidas:  $\|\mathbf{M}_1\mathbf{M}_2\| = a$ ,  $\|\mathbf{M}_2\mathbf{M}_3\| = b$ , y  $\|\mathbf{M}_1\mathbf{M}_3\| = c$ .

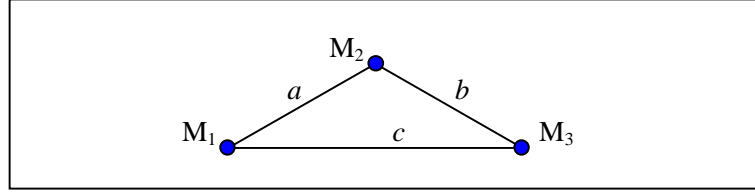


Figura 2.1-2: Distancias entre  $\mathbf{M}_1$ ,  $\mathbf{M}_2$  y  $\mathbf{M}_3$ .

Escribiendo las ecuaciones de distancia entre 2 puntos para  $\mathbf{M}_1$ ,  $\mathbf{M}_2$  y  $\mathbf{M}_3$ , se obtiene:

$$\begin{aligned}(X_2 - X_1)^2 + (Y_2 - Y_1)^2 + (Z_2 - Z_1)^2 &= a^2 \\ (X_3 - X_2)^2 + (Y_3 - Y_2)^2 + (Z_3 - Z_2)^2 &= b^2 \\ (X_3 - X_1)^2 + (Y_3 - Y_1)^2 + (Z_3 - Z_1)^2 &= c^2\end{aligned}\tag{2.1.3}$$

Sean  $\mathbf{m}_1$ ,  $\mathbf{m}_2$  y  $\mathbf{m}_3$  sus proyecciones en la imagen, de coordenadas conocidas  $(x_i, y_i)$ ,  $i = 1..3$ . Las coordenadas de  $\mathbf{M}_1$ ,  $\mathbf{M}_2$  y  $\mathbf{M}_3$  están relacionadas con las coordenadas de  $\mathbf{m}_1$ ,  $\mathbf{m}_2$  y  $\mathbf{m}_3$ , por las ecuaciones (2.1.2). Reemplazándolas en (2.1.3), se obtiene el siguiente sistema para  $Z_1$ ,  $Z_2$  y  $Z_3$  :

$$\begin{aligned}\left(\frac{Z_2 x_2}{f} - \frac{Z_1 x_1}{f}\right)^2 + \left(\frac{Z_2 y_2}{f} - \frac{Z_1 y_1}{f}\right)^2 + (Z_2 - Z_1)^2 &= a^2 \\ \left(\frac{Z_3 x_3}{f} - \frac{Z_2 x_2}{f}\right)^2 + \left(\frac{Z_3 y_3}{f} - \frac{Z_2 y_2}{f}\right)^2 + (Z_3 - Z_2)^2 &= b^2 \\ \left(\frac{Z_3 x_3}{f} - \frac{Z_1 x_1}{f}\right)^2 + \left(\frac{Z_3 y_3}{f} - \frac{Z_1 y_1}{f}\right)^2 + (Z_3 - Z_1)^2 &= c^2\end{aligned}\tag{2.1.4}$$

Expandiendo los binomios y factorizando por  $Z_1$ ,  $Z_2$  y  $Z_3$  el sistema toma la forma:

$$\begin{aligned}S_1 Z_1^2 + 2W_{12} Z_1 Z_2 + S_2 Z_2^2 &= a^2 \\ S_2 Z_2^2 + 2W_{23} Z_2 Z_3 + S_3 Z_3^2 &= b^2 \\ S_1 Z_1^2 + 2W_{13} Z_1 Z_3 + S_3 Z_3^2 &= c^2\end{aligned}\tag{2.1.5}$$

Donde  $S_i$  son las constantes que acompañan a los términos cuadráticos  $Z_i^2$  y  $W_{ij}$  son las constantes que acompañan a los términos mixtos  $Z_i Z_j$ .

$$\begin{aligned} S_i &= \left( \frac{x_i}{f} \right)^2 + \left( \frac{y_i}{f} \right)^2 + 1 \\ W_{ij} &= \frac{x_i x_j}{f^2} + \frac{y_i y_j}{f^2} + 1 \end{aligned} \quad (2.1.6)$$

Resolviendo el sistema (2.1.5) se puede obtener las coordenadas  $Z$  de  $\mathbf{M}_1$ ,  $\mathbf{M}_2$  y  $\mathbf{M}_3$ , con las cuales se puede determinar las otras 2 coordenadas utilizando las ecuaciones (2.1.2), completando así la reconstrucción 3D de los puntos. Sin embargo, este sistema es polinomial y no puede ser resuelto en forma analítica, por lo cual es necesario utilizar un método numérico para resolverlo.

Sean  $g_1$ ,  $g_2$  y  $g_3$  funciones  $\Re^3 \rightarrow \Re$  de  $Z_1$ ,  $Z_2$  y  $Z_3$  tales que:

$$\begin{aligned} g_1(Z_1, Z_2, Z_3) &= S_1 Z_1^2 + 2M_{12} Z_1 Z_2 + S_2 Z_2^2 - a^2 \\ g_2(Z_1, Z_2, Z_3) &= S_2 Z_2^2 + 2M_{23} Z_2 Z_3 + S_3 Z_3^2 - b^2 \\ g_3(Z_1, Z_2, Z_3) &= S_1 Z_1^2 + 2M_{13} Z_1 Z_3 + S_3 Z_3^2 - c^2 \end{aligned} \quad (2.1.7)$$

Sea  $\mathbf{G} : \Re^3 \rightarrow \Re^3$  una función vectorial tal que:

$$\mathbf{G}(Z_1, Z_2, Z_3) = (g_1(Z_1, Z_2, Z_3), g_2(Z_1, Z_2, Z_3), g_3(Z_1, Z_2, Z_3)) \quad (2.1.8)$$

Es fácil ver que resolver el sistema (2.1.5) para  $Z_1$ ,  $Z_2$  y  $Z_3$ , equivale a encontrar un cero (vectorial) de la función  $\mathbf{G}$ . Es decir, un trío  $(Z_1, Z_2, Z_3)$  es solución del sistema (2.1.5) si y solo si es raíz de la función  $\mathbf{G}$ . A continuación se describe cómo utilizar el método de Newton-Raphson para encontrar un cero de la función  $\mathbf{G}$  y resolver el sistema (2.1.5).

### 2.1.3 Newton-Raphson aplicado al problema

El método de Newton-Raphson (Hartley y Ziserman, 2000) es un algoritmo numérico para encontrar ceros o raíces de funciones reales. El algoritmo requiere un punto de partida razonablemente cercano a la raíz de interés, a partir del cual procede iterativamente generando nuevos puntos cada vez más cercanos a la raíz.

Sea  $g : \Re \rightarrow \Re$  una función derivable en el intervalo de interés. Desde un punto de partida  $x_0$ , el algoritmo genera en cada iteración  $n$ , un nuevo punto  $x_{n+1}$  a partir del punto anterior  $x_n$ , tomando la raíz de la aproximación lineal de  $g$  en  $x_n$  :

$$x_{n+1} = x_n - \frac{g(x_n)}{g'(x_n)} \quad (2.1.9)$$

Donde  $g'(x_n)$  denota la derivada de  $g$  en  $x_n$ . El algoritmo termina en la iteración  $n$  si se obtiene un punto lo suficientemente cercano a una raíz, es decir si se obtuvo un  $x_n$  tal que  $|g(x_n)| < \varepsilon$  para un  $\varepsilon$  dado, o bien si  $n$  excede un número máximo de iteraciones definido.

Newton-Raphson puede ser generalizado para encontrar ceros de funciones vectoriales como la de la ecuación (2.1.8). En este caso, tanto  $x$  como  $g(x)$  son vectores, y la ecuación (2.1.9) debe ser reemplazada por:

$$x_{n+1} = x_n - \mathbf{J}_{\mathbf{G}}^{-1}(x_n) \mathbf{G}(x_n) \quad (2.1.10)$$

Donde  $\mathbf{J}_{\mathbf{G}}^{-1}(x_n)$  denota la inversa de la matriz Jacobiana de la función  $\mathbf{G}$  en  $x_n$ . Las condiciones de término del algoritmo son las mismas que en el caso anterior, salvo que en este caso se pide  $\|\mathbf{G}(x_n)\| < \varepsilon$ , dado que se trata de una función vectorial.

En el caso de la función de interés (ecuación 2.1.8)  $x$  es un vector  $\mathbf{Z} = (Z_1, Z_2, Z_3)$  y la matriz Jacobiana  $\mathbf{J}_G(\mathbf{Z})$ , es de dimensión 3x3:

$$\mathbf{J}_G(\mathbf{Z}) = \begin{bmatrix} \frac{\partial g_1}{\partial Z_1} & \frac{\partial g_1}{\partial Z_2} & \frac{\partial g_1}{\partial Z_3} \\ \frac{\partial g_2}{\partial Z_1} & \frac{\partial g_2}{\partial Z_2} & \frac{\partial g_2}{\partial Z_3} \\ \frac{\partial g_3}{\partial Z_1} & \frac{\partial g_3}{\partial Z_2} & \frac{\partial g_3}{\partial Z_3} \end{bmatrix} \quad (2.1.11)$$

Los términos  $\partial g_i / \partial Z_j$  denotan la derivada parcial de la función  $g_i$  con respecto a la variable  $Z_j$ , y se obtienen derivando las funciones (2.1.7). Reemplazando estas derivadas en la matriz anterior, se obtiene la expresión para la matriz jacobiana de  $\mathbf{G}$ :

$$\mathbf{J}_G(\mathbf{Z}) = \begin{bmatrix} 2S_1Z_1 - 2W_{12}Z_2 & 2S_2Z_2 - 2W_{12}Z_1 & 0 \\ 0 & 2S_2Z_2 - 2W_{23}Z_3 & 2S_3Z_3 - 2W_{23}Z_2 \\ 2S_1Z_1 - 2W_{13}Z_3 & 0 & 2S_3Z_3 - 2W_{13}Z_1 \end{bmatrix} \quad (2.1.12)$$

Las coordenadas del punto de partida  $\mathbf{Z}_0 = (Z_{10}, Z_{20}, Z_{30})$  deben ser escogidas dentro del rango en el eje  $Z$  en que se espera encontrar  $\mathbf{M}_1$ ,  $\mathbf{M}_2$  y  $\mathbf{M}_3$ . Si la reconstrucción 3D se realiza en una secuencia de video, se puede utilizar como punto de partida para cada frame, la solución del frame anterior de modo de contar siempre con un buen punto de partida.

Utilizando este algoritmo se puede reconstruir la posición en el espacio de 3 puntos cuyas distancias son conocidas, a partir de sus proyecciones en una imagen. En la siguiente sección se muestra una forma de aplicar este método para realizar Head Tracking en el espectro infrarrojo utilizando el Nintedo Wii Remote.

## 2.2 Head Tracking básico

En esta sección se describe cómo utilizar el método de la sección anterior para implementar un sistema de HT en el espectro infrarrojo, utilizando el Nintendo Wii Remote para detectar un cintillo con tres diodos emisores infrarrojos ubicado en la cabeza del usuario. Posteriormente, en la sección 2.3 se mostrará como extender este sistema para no requerir que el usuario deba mirar de frente a la cámara.

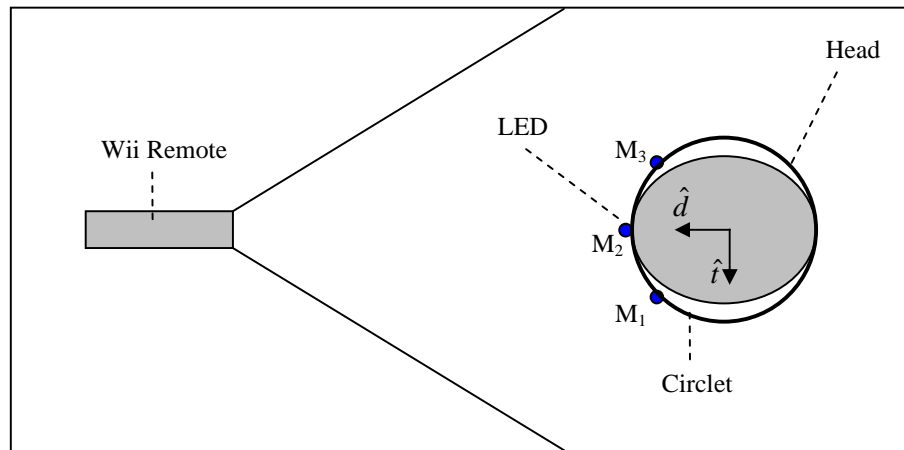


Figura 2.2-1: Esquema del sistema de HT básico (vista superior).

La figura 2.2.1 muestra un esquema del sistema. El vector  $\hat{d}$  indica la dirección en que apunta la cabeza y el vector  $\hat{t}$  indica la dirección en que apunta el oído izquierdo. Los tres diodos están ubicados sobre la frente del usuario en un cintillo circular de radio  $r$  conocido. El diodo central ( $M_2$ ) indica la posición del centro de la frente. Los otros dos ( $M_1$  y  $M_3$ ) están ubicados a ambos lados del diodo central, formando un ángulo de  $45^\circ$  con éste, desde el centro del cintillo.

El Nintendo Wii Remote incorpora una cámara infrarroja [13], de 1024 x 768 píxeles de resolución y una tasa de refresco de 100 Hz. El campo de visión (FoV) de la cámara en el plano horizontal es de  $45^\circ$ , lo que determina una distancia focal medida en píxeles de 1236.

Utilizando el método de reconstrucción descrito en la sección 2.1 se puede obtener la posición 3D de los 3 diodos del cintillo, y con esto obtener una buena estimación de la posición y la orientación de la cabeza del usuario. Los únicos datos necesarios para realizar la reconstrucción son las distancias entre los diodos, y sus posiciones en la imagen.

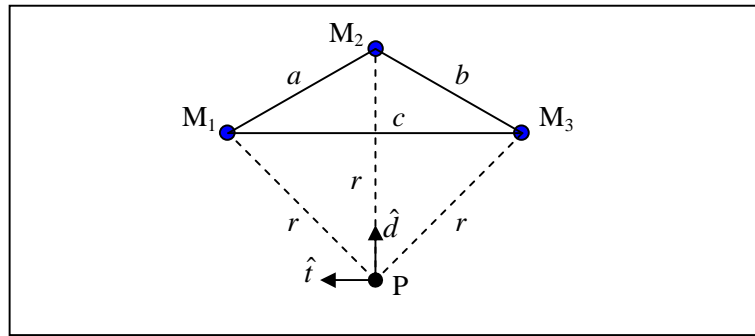


Figura 2.2-2: Distancias importantes para el sistema de HT.

Sean  $M_1$ ,  $M_2$  y  $M_3$  los 3 diodos del cintillo, siendo  $M_2$  el diodo central. Sean  $a$ ,  $b$  y  $c$  las distancias entre los diodos, como se muestra en la figura 2.2.2. Para un cintillo de radio  $r$ , los valores de las distancias  $a$ ,  $b$  y  $c$  son:

$$\begin{aligned} a &= b = 2r \sin(\pi/8) \\ c &= 2r \sin(\pi/4) \end{aligned} \tag{2.2.1}$$

Una vez reconstruidas las posiciones de los diodos  $M_1$ ,  $M_2$  y  $M_3$ , se puede calcular la dirección  $\hat{d}$  en que apunta la cabeza, como la diferencia vectorial entre el diodo central  $M_2$ , y el punto medio entre los diodos laterales:



$$\begin{aligned}\vec{d} &= \mathbf{M}_2 - \frac{1}{2}(\mathbf{M}_1 + \mathbf{M}_3) \\ \hat{d} &= \frac{\vec{d}}{\|\vec{d}\|}\end{aligned}\tag{2.2.2}$$

Con la posición de  $\mathbf{M}_2$  y la dirección  $\hat{d}$ , se puede obtener la posición  $\mathbf{P}$  del centro del cintillo, la cual es una buena representación de la posición de la cabeza:

$$\mathbf{P} = \mathbf{M}_2 - r\hat{d}\tag{2.2.3}$$

Una segunda dirección importante que define la inclinación de la cabeza, es la dirección del eje de los oídos  $\hat{t}$  (ver figura 2.2.1). Esta es ortogonal a la dirección  $\hat{d}$  en que apunta la cabeza, y puede ser calculada como la diferencia entre  $\mathbf{M}_1$  y  $\mathbf{M}_3$ .

$$\begin{aligned}\vec{t} &= \mathbf{M}_1 - \mathbf{M}_3 \\ \hat{t} &= \frac{\vec{t}}{\|\vec{t}\|}\end{aligned}\tag{2.2.4}$$

La rotación de la cabeza con respecto al sistema de la cámara puede describirse a partir de una matriz de rotación  $\mathbf{R}$ , cuya primera y tercera columna son los vectores  $\hat{t}$  y  $\hat{d}$  respectivamente y cuya segunda columna corresponde al producto cruz  $\hat{d} \times \hat{t}$ .

$$\mathbf{R} = [\hat{t} \quad \hat{d} \times \hat{t} \quad \hat{d}]\tag{2.2.5}$$

Puede verificarse que cuando no hay rotación de la cabeza,  $\mathbf{R}$  corresponde a la matriz identidad. En efecto, en dicho caso la dirección  $\hat{d}$  en que apunta la cabeza coincide con el vector canónico  $\hat{k}$ , que indica la dirección del eje Z. Por otra parte, la dirección  $\hat{t}$  del eje de los oídos coincide con el vector canónico  $\hat{i}$ , que indica la dirección del eje X. Finalmente,  $\hat{d} \times \hat{t}$  coincide con el vector canónico  $\hat{j}$ , que indica la dirección del eje Y.

### 2.3 Head Tracking extendido

El sistema de HT presentado en la sección anterior requiere que el usuario esté siempre mirando a la cámara, dado que el programa necesita ver siempre los tres diodos infrarrojos ubicados en su frente para poder realizar la reconstrucción 3D (ver figura 2.2.1). Esta restricción puede resultar indiferente para muchas aplicaciones de HT, pero puede volverse una limitación si va a ser utilizado por ejemplo, en un sistema de audio espacial.

En esta sección se presenta una forma de extender el sistema de HT visto en la sección anterior, para conseguir una solución que no requiera que el usuario permanezca mirando a la cámara, y que pueda ocuparse en otras aplicaciones como audio espacial.

Se utiliza ahora un cintillo similar al de la sección anterior, pero que incorpora 5 diodos adicionales, completando un octógono regular. Los nuevos diodos permiten al Wii Remote ver al cintillo desde cualquier punto, sin importar hacia donde esté mirando el usuario. La figura 2.3.1 muestra un esquema del sistema.

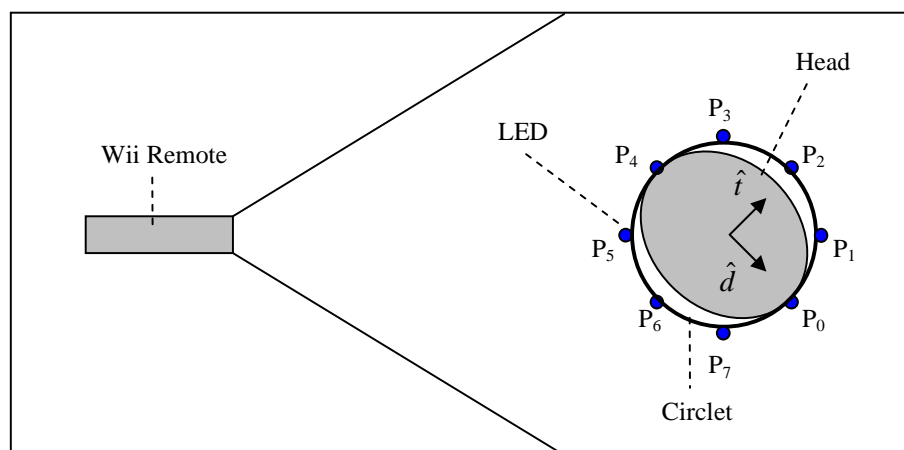


Figura 2.3-1: Esquema del sistema de HT extendido (vista superior).

El vector  $\hat{d}$  indica la dirección en que apunta la cabeza y el vector  $\hat{i}$  indica la dirección en que apunta el oído izquierdo. Los 8 diodos del octágono están enumerados desde el diodo frontal  $P_0$  hacia la derecha como muestra la figura 2.3.1.

El problema de este enfoque radica en identificar a qué parte del cintillo corresponden los diodos vistos en la imagen y por lo tanto, hacia dónde está mirando el usuario. Para ayudar con esta tarea, el cintillo incorpora un par de diodos adicionales ( $R_f$  y  $R_r$ ), ubicados sobre el diodo frontal ( $P_0$ ) y el opuesto a este ( $P_4$ ) para marcar el frente y el reverso del cintillo (figura 2.3.2.a).

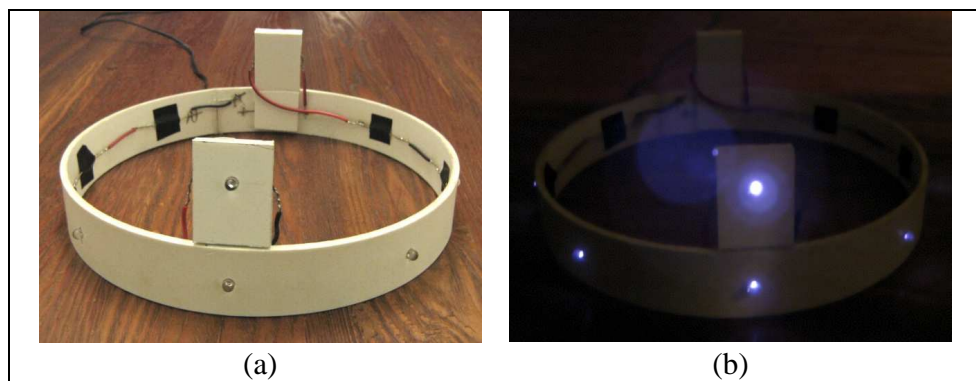


Figura 2.3-2: Fotografías del cintillo para HT extendido con y sin luz.

La figura 2.3.2.b da una mejor idea de lo que ve la cámara del Wii Remote. Los diodos ubicados en círculo ( $P_i$ ) serán llamados en adelante “diodos principales”, mientras los dos diodos adicionales ( $R_f$  y  $R_r$ ) serán llamados “diodos de referencia”.

A continuación se describe la implementación del sistema.

### 2.3.1 Implementación del sistema

#### Estructuras de datos

Estas son las principales estructuras de datos utilizadas:

- **Sensors[4]:** El procesador de la cámara del Wii Remote incorpora un multiobject tracking (MOT) engine [13], capaz que trackear hasta 4 fuentes de luz infrarroja. La información de las fuentes IR encontradas es entregada en este arreglo de 4 “sensores”, cuyos atributos más importantes son:

- **found:** indica si se encontró una fuente IR o no (booleano).
- **pos.x, pos.y:** indican la posición en la imagen de la fuente IR.

Esta estructura es la entrada del sistema.

- **Light:** representa un diodo IR del cintillo. Sus atributos mas importantes son:
  - **isVisible:** indica si el diodo es visible o no (booleano).
  - **pos.x, pos.y:** indican la posición del diodo en la imagen.
  - **Pos.x, Pos.y, Pos.z:** indican la posición del diodo en el espacio.
- **Circlet:** representa el cintillo IR. Sus atributos más importantes son:
  - **Prin[8]:** arreglo de 8 estructuras Light, correspondientes a los 8 diodos principales del cintillo ( $P_i$ ).
  - **RefF, RefR:** estructuras Light, correspondientes a los diodos de referencia frontal y trasero del cintillo ( $R_f$  y  $R_r$ ).
  - **Pos.x, Pos.y, Pos.z:** indican la posición en el espacio del centro del cintillo.
  - **Yaw, Pitch, Roll:** indican la orientación en el espacio del cintillo.

Esta estructura es la salida del sistema.

### Esquema del Sistema

El objetivo del sistema es obtener la mayor cantidad posible de información espacial (3D) del cintillo en la estructura Circlet a partir de la información recibida del Wii Remote en el arreglo Sensors[4]. En particular, se desea obtener su posición y orientación en el espacio.

Para lograr este objetivo, se separó el sistema en dos módulos o procesos principales: un Módulo de Tracking y un Módulo de Reconstrucción, como se muestra en la figura 2.3.3.

- El Módulo de Tracking toma la información recibida del Wii Remote en el arreglo Sensors[] y determina qué diodos del cintillo aparecen en la imagen, actualizando la información 2D de la estructura Circlet.
- El Módulo de Reconstrucción toma la información 2D de la estructura Circlet, y si es suficiente, reconstruye la posición y orientación del cintillo en el espacio, actualizando la información 3D de la estructura.

A continuación se describe en detalle ambos módulos.

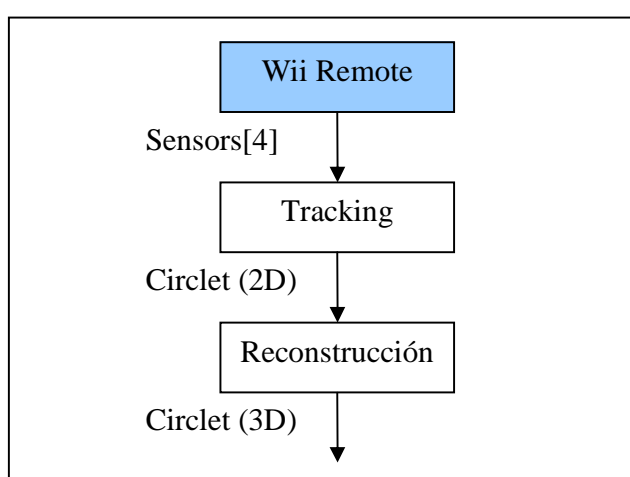


Figura 2.3.1-1: Esquema del proceso de HT.

### 2.3.2 Módulo de Tracking

El objetivo de este módulo es determinar a qué diodo del cintillo (Prin[i], RefF o RefR) corresponde cada luz vista en la imagen (Sensors[i]). Esto consiste en asignar a cada Sensors[i] un diodo del cintillo, utilizando dos funciones: Assign(i,j) y UnAssign(i,j) y un arreglo Assignment[]:

- **Assignment[4]:** registra a qué diodo del cintillo está asignado cada Sensors[i].
  - Si Assignment[i] = -1, Sensor[i] no tiene asignación.
  - Si Assignment[i] = j,  $j \in [0,7]$ , Sensors[i] fue asignado a Prin[j].
  - Si Assignment[i] = 8, Sensor[i] está asignado a RefF.
  - Si Assignment[i] = 9, Sensor[i] está asignado a RefR.
- **Assign(i,j):** asigna el Sensor[i] a un diodo X del cintillo, marcándolo visible (X.isVisible = true), actualizando su posición (X.pos = Sensors[i].pos) y registrando la asignación (Assignment[i] = j).
  - Si  $j \in [0,7]$ , X corresponde a Prin[j].
  - Si j = 8, X corresponde a RefF.
  - Si j = 9, X corresponde a RefR.
- **UnAssign(i,j):** desasigna el Sensor[i] de un diodo X del cintillo, marcándolo como no visible (X.isVisible = false) y eliminando la asignación (Assignment[i] = -1).
  - j determina a X igual que en Assign(i,j).

### Cuándo asignar

Para aprovechar al máximo el multiobject tracking engine (MOT) [13] del Wii Remote, se intenta conservar las asignaciones previas de cada Sensors[i] mientras no ocurran cambios importantes en la información recibida. Para detectar estos cambios y determinar cuando se deben asignar/desasignar, se definió una función Analyze() y dos estructuras: SensorsPrev[] y SensorsChange[]:

- `SensorsPrev[4]`: almacena el estado de `Sensors[]` en el frame anterior.
- `SensorsChange[4]`: indica qué se debe hacer con cada `Sensors[i]`.
- `Analyze()`: determina si hubieron cambios y qué se debe hacer con cada `Sensors[i]`.

`Analyze()` considera 3 cambios importantes posibles para cada `Sensors[i]`:

- Si `Sensors[i]` no tenía asignación (`Assignment[i] = -1`) y encontró una luz (`Sensors[i].Found = true`), entonces se debe asignar a algún diodo del cintillo, lo cual se indica con `SensorsChange[i] = 1`.
- Si `Sensors[i]` tenía asignación (`Assignment[i] != -1`) y no encontró una luz (`Sensors[i].Found = false`), entonces se debe eliminar la asignación, lo cual se indica con `SensorsChange[i] = 3`,
- Si `Sensors[i]` tenía asignación y encontró una luz, pero esta se desplazó mucho en la imagen (`Distance(SensorPrev.pos, Sensor.pos) > DistTreshold1`), entonces el sensor se cambio a otro diodo mas intenso, en cuyo caso se debe eliminar la asignación anterior y asignar un nuevo diodo. Esto se indica con `SensorsChange[i] = 2`.
- En cualquier otro caso, se considera que `Sensors[i]` no tuvo cambios y se indica con `SensorsChange[i] = 0`.

Finalmente, para saber si hay sensores que asignar, se definió una función `AnythingChanged()`. Esta retorna `true` si la suma de los elementos de `SensorsChange[]` es distinta de cero, y `false` en otro caso.

---

<sup>1</sup> `DistTreshold` representa el máximo desplazamiento que puede tener la luz detectada por un sensor antes de considerar que se cambió a otro diodo

## Cómo asignar

El módulo de Tracking tiene un código similar al siguiente:

```

Analyze();
if (AnythingChanged()) {
    if (TryReferenceAssignment());
    else {
        UnAssignLost();
        ReAssignAll();
        TrySpecialAssignment();
    }
}
else {
    ReAssignAll();
}

```

Primero, se llama a `Analyze()` para determinar si hubo cambios importantes para cada `Sensors[i]` y qué hacer en cada caso. Luego se llama a `AnythingChanged()` para preguntar si hubo cambios importantes:

- Si hubo cambios, se intenta realizar una “Asignación con Referencia” (`TryReferenceAssignment()`): la que utiliza los diodos de referencia (frontal o trasero) como guía para asignar los demás diodos. Si `TryReferenceAssignment()` fracasa, entonces
  - `UnAssignLost()` desasigna todos los sensores que deben ser desasignados (aquellos que fueron marcados con `SensorChange[i] = 2` o `3` por `Analyze()`)
  - `ReAssignAll()` reasigna todos los sensores que no cambiaron (aquellos que fueron marcados con `SensorChange[i] = 0` por `Analyze()`) con sus asignaciones anteriores.
  - Se intenta realizar una “Asignación Especial” (`TrySpecialAssignment()`): la cual utiliza diodos principales previamente asignados como guía para asignar los demás diodos.



- Si no hubo cambios, ReAssignAll() reasigna todos los sensores que no cambiaron (aquellos que fueron marcados con  $\text{SensorChange}[i] = 0$  por  $\text{Analyze}()$ ) con sus asignaciones anteriores (en este caso, todos).

Se intenta realizar Asignaciones con Referencia cada vez que sea posible. Las Asignaciones Especiales se realizan sólo cuando no es posible realizar una Asignación con Referencia. A continuación se describe estas dos estrategias de asignación.

### **Asignación con Referencia (TryReferenceAssignment())**

Para realizar una asignación con referencia primero se debe determinar si hay diodos de referencia en la imagen. Para esto se definió una función  $\text{SearchRefLight}()$  la cual identifica el diodo de referencia buscando el trío de diodos que forme el ángulo más recto.

$\text{SearchRefLight}()$  requiere que los 4 sensores sean visibles ( $\text{Sensor}[i].\text{found} = \text{true}$ ). Esta función sigue los siguientes pasos:

1. Para cada  $\text{Sensors}[i]$  busca el sensor  $A[i]$  mas cercano a este
2. Para cada  $A[i]$  busca el sensor  $B[i]$  mas cercano a este, excluyendo a  $\text{Sensors}[i]$ .
3. Para cada trío  $\{\text{Sensors}[i], A[i], B[i]\}$  determina el ángulo  $\text{Angle}[i]$  formado entre los vectores:  $(\text{Sensors}[i].\text{pos} - A[i].\text{pos})$  y  $(B[i].\text{pos} - A[i].\text{pos})$
4. Determinar el ángulo más recto de  $\text{Angle}[i]$ , es decir, que minimize  $(\pi/2 - \text{Angle}[i])$ .
5. Si el ángulo más recto es “suficientemente recto”, es decir si  $|\pi/2 - \text{Angle}[i]| \leq \pi/8^2$ , se toma el trío  $\{\text{Sensors}[i], A[i], B[i]\}$  y se continúa con el paso 6. Si no,  $\text{SearchRefLight}()$  termina retornando false.

A continuación se determina si el diodo de ref. es  $\text{Sensors}[i]$  o  $B[i]$ .

---

<sup>2</sup> Un ángulo es considerado “suficientemente recto” si esta en el rango  $[\pi/2 - \pi/8, \pi/2 + \pi/8]$

6. Sean  $U, V, W$  los 3 elementos de  $Sensors[]$  distintos a  $Sensors[i]$ , ordenados de izquierda a derecha en la imagen ( $U.pos.x \leq V.pos.x \leq W.pos.x$ ). Se determina  $angle1$ , el ángulo conformado entre los vectores:  $(U.pos - V.pos)$  y  $(W.pos - V.pos)$
  7. Sean  $P, Q, R$  los 3 elementos de  $Sensors[]$  distintos a  $B[i]$ , ordenados de izquierda a derecha en la imagen ( $P.pos.x \leq Q.pos.x \leq R.pos.x$ ). Se determina  $angle2$ , el ángulo conformado entre los vectores:  $(P.pos - Q.pos)$  y  $(R.pos - Q.pos)$
  8. Si  $angle1 \geq angle2$ , el diodo de referencia es  $Sensors[i]$ , si no, es  $B[i]$ . En los dos casos,  $A[i]$  es el diodo principal más cercano a éste. Se guarda los índices de ambos en variables  $ref$  y  $prin$  respectivamente, y se retorna true.
- Si  $SearchRefLight()$  retorna false, no se puede realizar una Asignación con Referencia y  $TryReferenceAssignment()$  retorna false. (Se deberá intentar una Asignación Especial).
  - Si  $SearchRefLight()$  retorna true, se puede realizar una Asignación de Referencia, pero falta determinar si el diodo de referencia encontrado corresponde al diodo delantero ( $RefF$ ) o al trasero ( $RefR$ ). Para determinar esto se definió otra función  $DetermineSide()$ .
  - Si en el frame anterior se veía alguno de los dos diodos contiguos al diodo principal frontal,  $DetermineSide()$  retorna 1, indicando que el diodo de referencia corresponde al frontal.
  - Si en el frame anterior se veía alguno de los dos diodos contiguos al diodo principal trasero,  $DetermineSide()$  retorna 2, indicando que el diodo de referencia corresponde al trasero.
  - En cualquier otro caso,  $DetermineSide()$  retorna 1, de modo que si falta información, se asume que se esta viendo el frente del cintillo.

Con el valor retornado por `DetermineSide()` se puede terminar la Asignación de Referencia. Sea `side` el valor retornado por `DetermineSide()` y sean `i, j` los índices de los 2 elementos de `Sensors[]` distintos a `ref` y `prin`, ordenados de izquierda a derecha en la imagen. ( $\text{Sensors}[i].\text{pos.x} \leq \text{Sensors}[j].\text{pos.x}$ )

Para `ref` y `prin` hay solo 2 opciones:

- 1) Si `side = 1`, se ejecuta `Assign(ref, 8)` y `Assign(prin, 0)`.
- 2) Si `side = 2`, se ejecuta `Assign(ref, 9)` y `Assign(prin, 4)`.

Para `i` y `j` hay 3 casos posibles (6 dependiendo de `side`):

- a) Si  $\text{Sensors}[i].\text{pos.x} \leq \text{Sensors}[j].\text{pos.x} \leq \text{Sensors}[\text{prin}].\text{pos.x}$ 
  - a.1) Si `side = 1`, se ejecuta `Assign(i,6)` y `Assign(j,7)`.
  - a.2) Si `side = 2`, se ejecuta `Assign(i,2)` y `Assign(j,3)`.
- b) Si  $\text{Sensors}[i].\text{pos.x} \leq \text{Sensors}[\text{prin}].\text{pos.x} \leq \text{Sensors}[j].\text{pos.x}$ 
  - b.1) Si `side = 1`, se ejecuta `Assign(i,7)` y `Assign(j,1)`.
  - b.2) Si `side = 2`, se ejecuta `Assign(i,3)` y `Assign(j,5)`.
- c) Si  $\text{Sensors}[\text{prin}].\text{pos.x} \leq \text{Sensors}[i].\text{pos.x} \leq \text{Sensors}[j].\text{pos.x}$ 
  - c.1) Si `side = 1`, se ejecuta `Assign(i,1)` y `Assign(j,2)`.
  - c.2) Si `side = 2`, se ejecuta `Assign(i,5)` y `Assign(j,6)`.

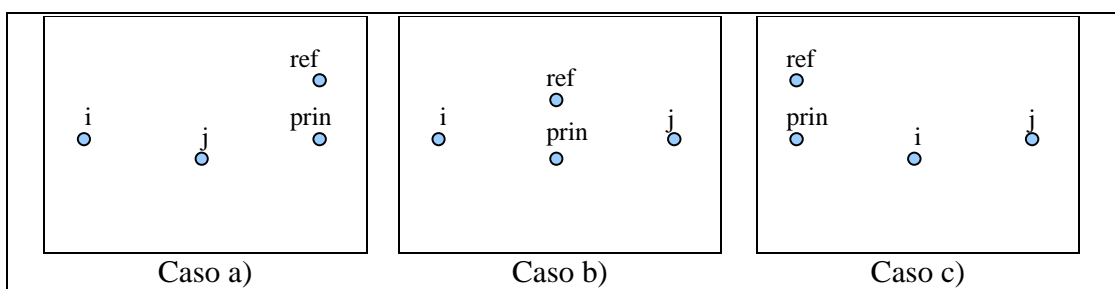


Figura 2.3.2-1: Casos posibles en una asignación con referencia.

### Asignación Especial (TrySpecialAssignment())

Para realizar una asignación especial primero se debe verificar si hay al menos un sensor asignado a un diodo principal. Para hacer esto se implemento una función SearchPrinLight(). Esta función itera sobre el arreglo Assignment[] buscando un sensor asignado a un diodo principal, es decir buscando un índice  $i$  tal que  $0 \leq \text{Assignment}[i] \leq 7$ . Si existe tal índice, SearchPrinLight() retorna true, en otro caso retorna false.

- Si SearchPrinLight() retorna false, no existe un diodo principal asignado, por lo que no se puede realizar una Asignación Especial, y TrySpecialAssignment() termina.
- Si SearchPrinLight() retorna true, entonces existe al menos un diodo principal asignado que puede ser utilizado como guía para asignar los demás sensores.

Para asignar los sensores pendientes por asignar (aquellos que fueron marcados con  $\text{SensorChange}[] = 1$  o  $2$  en el proceso Analyze()) se debe ordenarlos de izquierda a derecha (según su coordenada X en la imagen). Sea  $l$  el índice del sensor asignado al diodo principal ubicado más a la izquierda en la imagen, y  $r$  el índice del sensor asignado al diodo principal ubicado más a la derecha. (left y right pueden eventualmente coincidir)

Sea Left( $i$ ) una función que entrega el índice del sensor pendiente ubicado inmediatamente a la izquierda de  $\text{Sensors}[i]$  y sea Right( $i$ ) una función que entrega el índice del sensor pendiente ubicado inmediatamente a su derecha. En caso de no haber ningún sensor, ambas funciones retornan -1.

Para asignar los sensores pendientes ubicados a la izquierda de  $l$ , basta con ejecutar el siguiente ciclo:

```

while(Left(l)!=-1)
{
    Assign(Left(l),Assignment[l]-1);
    l = Left(l);
}

```

Finalmente, para asignar los sensores pendientes ubicados a la derecha de r, basta con ejecutar el siguiente ciclo análogo al anterior:

```

while(Right(r)!=-1)
{
    Assign(Right(l),Assignment[r]+1);
    r = Right(r);
}

```

### 2.3.3 Módulo de Reconstrucción

El objetivo de este módulo es reconstruir la posición 3D de los diodos del cintillo. Para esto, se requiere que haya 3 diodos principales contiguos visibles. Se definió una función CanReconstruct() para verificar si esto se cumple.

CanReconstruct() itera sobre cada uno de los diodos principales (Circlet.Prin[]) verificando si este y los dos diodos contiguos son visibles. Si se cumple esta condición para algún trío de diodos, guarda el índice del diodo central en una variable global midIndex, y retorna true. Si no, retorna false.

Seudo-Codigo para CanReconstruct ():

```

for(i = 1; i <= 8; i++) {
    if(Circlet.Prin[(i-1)%8].isVisible and
        Circlet.Prin[i%8].isVisible and
        Circlet.Prin[(i+1)%8].isVisible) {
        midIndex = i % 8;
        return true;
    }
}
return false;

```

- Si `CanReconstruct()` retorna false, entonces no hay un trío de diodos principales visible para reconstruir el cintillo en esta iteración y el Modulo de Reconstrucción termina.
- Si `CanReconstruct()` retorna true, entonces existe un trío de diodos principales visible, indicado por `midIndex`, y se puede reconstruir su posición 3D a partir de sus posiciones en la imagen, utilizando el algoritmo de la sección 2.1.

### Reconstrucción 3D del cintillo

Una vez reconstruidas las posiciones de 3 diodos principales del cintillo, se puede reconstruir fácilmente la posición del centro y de los demás diodos principales. Sean  $M_1$ ,  $M_2$  y  $M_3$  las posiciones reconstruidas de los 3 diodos principales visibles. Sea  $P$  la posición del centro, y sean  $M_i$ ,  $i = 4..8$ , las posiciones de los demás diodos principales.

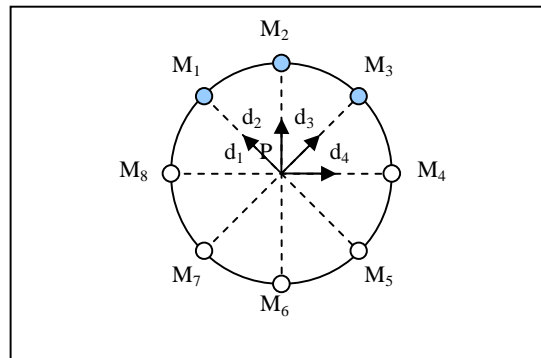


Figura 2.3.3-1: Reconstrucción del cintillo.

$P$  puede ser reconstruido a partir de  $M_1$ ,  $M_2$  y  $M_3$ , utilizando la ecuación 2.2.3. Para reconstruir las posiciones  $M_i$  de los demás diodos, se requiere determinar las 4 direcciones que van de  $P$  a  $M_1$ ,  $M_2$ ,  $M_3$  y  $M_4$ . Sean  $d_i$ ,  $i = 1..4$ , tales direcciones. Estas están dadas por:

$$\begin{aligned}
\mathbf{d}_1 &= (\mathbf{M}_1 - \mathbf{P}) / \|\mathbf{M}_1 - \mathbf{P}\| \\
\mathbf{d}_2 &= (\mathbf{M}_2 - \mathbf{P}) / \|\mathbf{M}_2 - \mathbf{P}\| \\
\mathbf{d}_3 &= (\mathbf{M}_3 - \mathbf{P}) / \|\mathbf{M}_3 - \mathbf{P}\| \\
\mathbf{d}_4 &= (\mathbf{M}_3 - \mathbf{M}_1) / \|\mathbf{M}_3 - \mathbf{M}_1\|
\end{aligned} \tag{2.3.1}$$

Las posiciones de los demás diodos principales son:

$$\begin{aligned}
\mathbf{M}_4 &= \mathbf{P} + r\mathbf{d}_4 \\
\mathbf{M}_5 &= \mathbf{P} - r\mathbf{d}_1 \\
\mathbf{M}_6 &= \mathbf{P} - r\mathbf{d}_2 \\
\mathbf{M}_7 &= \mathbf{P} - r\mathbf{d}_3 \\
\mathbf{M}_8 &= \mathbf{P} - r\mathbf{d}_4
\end{aligned} \tag{2.3.2}$$

Para determinar la Matriz de rotación  $R$  del cintillo se puede utilizar la ecuación 2.2.5, pero para esto, primero se debe definir los vectores  $\mathbf{d}$  y  $\mathbf{t}$ , para el cintillo utilizado en esta sección. Sea  $\mathbf{P}_0$  la posición del diodo del frente del cintillo y  $\mathbf{P}_2$  la posición del diodo de la izquierda (ver figura 2.3.1). La dirección  $\mathbf{d}$  en que apunta la cabeza puede definirse como:

$$\mathbf{d} = (\mathbf{P}_0 - \mathbf{P}) / \|\mathbf{P}_0 - \mathbf{P}\| \tag{2.3.3}$$

Finalmente, la dirección  $\mathbf{t}$  del eje de los oídos puede definirse como:

$$\mathbf{t} = (\mathbf{P}_2 - \mathbf{P}) / \|\mathbf{P}_2 - \mathbf{P}\| \tag{2.3.4}$$

## 2.4 Eliminación de la inclinación de la cámara

En esta sección se describe cómo utilizar los acelerómetros del Nintendo Wii Remote para detectar la inclinación del dispositivo y corregir la reconstrucción del cintillo, recalculando la posición de los diodos relativa a un nuevo sistema de referencia sin inclinación con respecto al plano de la Tierra.

Sea  $S$  el sistema de referencia de la cámara, de ejes  $X$ ,  $Y$  y  $Z$ , representado en gris en la figura (2.4.1), los cuales apuntan en las direcciones definidas en la sección 2.1. Este sistema se encuentra inclinado con respecto al plano de la Tierra, representado como  $\pi$  en la figura.

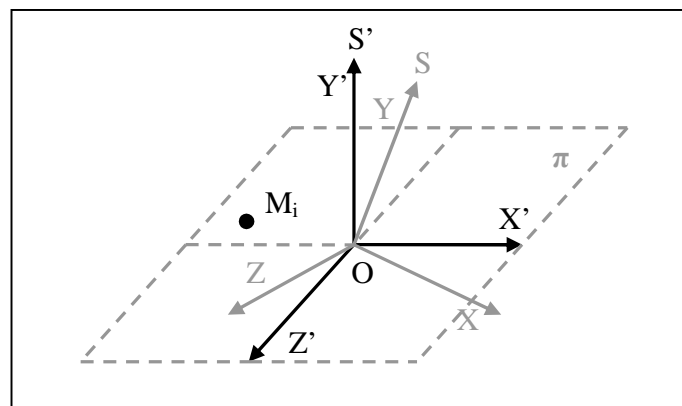


Figura 2.4-1: Corrección de la inclinación de la cámara.

Sea  $M_i$  la posición de un diodo del cintillo respecto al sistema  $S$  de la cámara. El objetivo es encontrar la posición de este diodo con respecto a un nuevo sistema de referencia  $S'$ , de ejes  $X'$ ,  $Y'$  y  $Z'$  (representado en negro), centrado en el mismo origen que  $S$ , pero:

- cuyo eje  $Y'$  es normal al plano de la Tierra y,
- cuyo eje  $Z'$  es la proyección ortogonal del eje  $Z$  sobre dicho plano.

Los ejes  $X'$  y  $Z'$  del sistema  $S'$ , quedan por lo tanto contenidos en el plano de la Tierra.



Para obtener posición  $M_i'$  del diodo relativa al sistema  $S'$ , se debe detectar la inclinación de la cámara con respecto al plano de la Tierra, la cual esta determinada por dos ángulos:

- Un ángulo de pitch, correspondiente al ángulo comprendido entre los ejes  $Z$  y  $Z'$
- Un ángulo de roll, correspondiente al ángulo comprendido entre los ejes  $X$  y  $X'$

El ángulo de yaw no es de interés en este caso.

Para determinar la inclinación de la cámara se puede utilizar los acelerómetros incorporados en el Wii Remote [13]. El dispositivo cuenta con 3 acelerómetros que miden aceleraciones en cada uno de los ejes. Cuando éste se encuentra en reposo, la única aceleración registrada es la aceleración de gravedad, de modo que se puede determinar su inclinación con respecto al plano de la Tierra observando como se descompone el vector aceleración en cada uno de los ejes.

La figura (2.4.2) muestra la disposición de los acelerómetros en el Nintendo Wii Remote. Notar que estos ejes no coinciden con los ejes del sistema de la cámara.

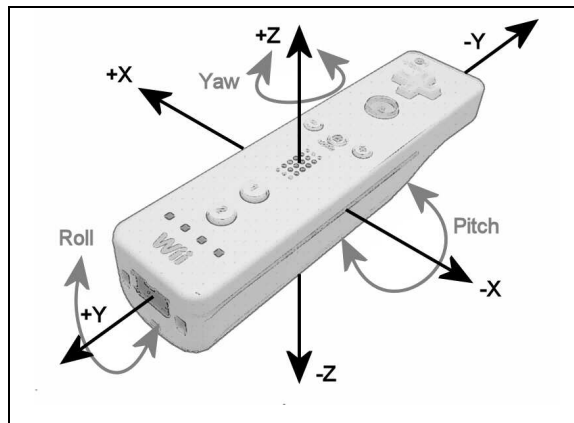


Figura 2.4-2: Disposición de los acelerómetros en el Nintendo Wii Remote.

El Nintendo Wii Remote entrega valores de aceleración normalizados por la aceleración de gravedad en el rango  $[-3g, +3g]$  [13]. Esto significa que si el dispositivo descansa en reposo sobre uno de sus costados en una superficie perfectamente horizontal, la aceleración registrada debería ser  $1g$  en el eje normal a ésta, y  $0$  en los otros dos. Si este no fuese el caso, se debe realizar un proceso de calibración para determinar el error de cada acelerómetro y substraerlo de las mediciones.

#### 2.4.1 Determinación del pitch y roll de la cámara

Sean  $a_x$ ,  $a_y$ ,  $a_z$ , las magnitudes de las aceleraciones registradas por el Wii Remote en cada uno de los ejes. El vector  $\vec{a} = (a_x, a_y, a_z)$ , indica la aceleración neta del dispositivo. Cuando éste se encuentra en reposo, los acelerómetros sólo registran la aceleración de gravedad, de modo que  $\|\vec{a}\| = 1g$ , y lo que es más importante,  $\vec{a}$  es normal al plano de la Tierra. La figura (2.4.3) muestra una instancia particular del vector  $\vec{a}$  en color negro:

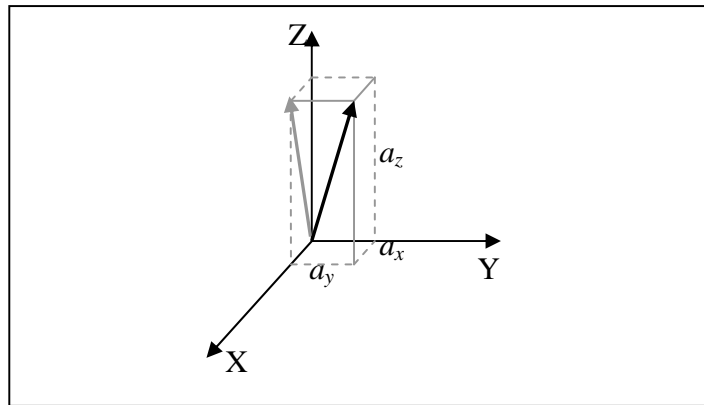


Figura 2.4.1-1: Observando la aceleración de gravedad.

El ángulo de pitch (en adelante  $\varphi$ ) de la cámara corresponde al conformado entre el vector  $\vec{a}$  y su proyección sobre el plano XZ (vector gris), el cual está dado por:

$$\varphi = \text{atan}\left(\frac{a_y}{\sqrt{a_x^2 + a_z^2}}\right) \quad (2.4.1)$$

Por otra parte, el ángulo de roll (en adelante  $\rho$ ) corresponde al conformado entre la proyección del vector  $\vec{a}$  sobre el plano XZ (vector gris) y el eje Z, el cual está dado por:

$$\rho = \text{atan}\left(\frac{a_x}{a_z}\right) \quad (2.4.2)$$

Ambos ángulos de rotación son independientes. Se puede verificar mirando la figura (2.4.2) que si  $a_y = 0$ , no hay rotación de pitch. Del mismo modo, si  $a_x = 0$ , no hay rotación de roll.

#### 2.4.2 Corrección de la reconstrucción 3D del cintillo

Una vez conocidos los ángulos de inclinación de la cámara, se puede determinar la posición 3D de los diodos del cintillo con respecto al nuevo sistema de referencia S', cuyo eje Y es normal a la superficie de la Tierra.

Sea  $M_i$  la posición de un diodo del cintillo en el sistema S de la cámara. Para obtener la posición  $M_i'$  del diodo con respecto al sistema S' se debe realizar una transformación lineal de rotación R sobre  $M_i$  que elimine las rotaciones de roll y pitch de la cámara.

$$\mathbf{M}_i' = \mathbf{R}\mathbf{M}_i \quad (2.4.3)$$

La matriz de rotación R es el producto de las matrices de rotación de pitch y roll:

$$\mathbf{R} = \mathbf{R}_{pitch} \mathbf{R}_{roll} \quad (2.4.5)$$

Estas matrices de rotación son las siguientes:

$$\mathbf{R}_{roll} = \begin{bmatrix} \cos(\rho) & -\sin(\rho) & 0 \\ \sin(\rho) & \cos(\rho) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.4.3)$$

$$\mathbf{R}_{pitch} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\varphi) & -\sin(\varphi) \\ 0 & \sin(\varphi) & \cos(\varphi) \end{bmatrix} \quad (2.4.4)$$

La matriz  $\mathbf{R}_{roll}$  rota el sistema S de la cámara en torno a su eje Z, haciendo coincidir su eje X con el eje X' del sistema S' (ver figura 2.4.1). Finalmente, la matriz  $\mathbf{R}_{pitch}$  rota el sistema S en torno a su eje X, haciendo coincidir los dos ejes restantes con sus respectivos del sistema S'. Ambas rotaciones no son conmutables por cómo se definieron los ángulos de rotación, por lo que la rotación de roll debe realizarse en primer lugar.

Multiplicando la matriz de rotación R por la posición  $M_i$  de cada diodo del cintillo, se corrige su reconstrucción, obteniéndose la inclinación de la cabeza con respecto al plano de la Tierra, sin importar la inclinación de la cámara.

### 3. EXPERIMENTOS Y RESULTADOS





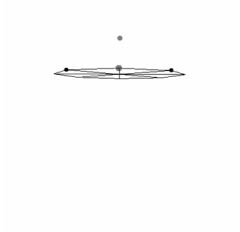
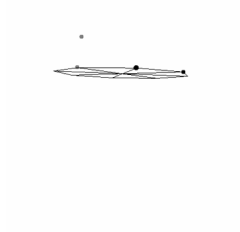
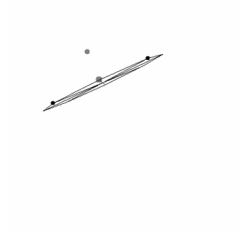
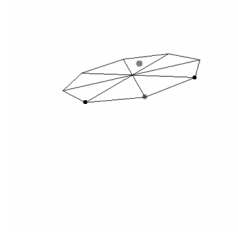
|   |   |  |   |
|---|---|--|---|
|  |  |  |  |
|  |  |  |  |
| Yaw: 1<br>Pitch: 3<br>Roll: 0   | Yaw: 35<br>Pitch: 6<br>Roll: 3  | Yaw: 5<br>Pitch: 4<br>Roll: 26   | Yaw: -7<br>Pitch: -11<br>Roll: 15   |

Figura 3-1: Capturas de pantalla de la aplicación para 4 poses distintas.

La primera fila muestra la pose real de la cabeza. La segunda muestra la reconstrucción 3D del cintillo realizada por la aplicación. La tercera muestra los ángulos (en grados) determinados por la aplicación. Los valores de los ángulos están redondeados al entero más cercano.

A continuación se analiza los resultados del sistema de HT en tres aspectos: Precisión, Rapidez y Robustez.

#### 3.1 Precisión

Para medir en forma empírica el error de las reconstrucciones del cintillo conseguidas por el sistema, sería necesario conocer las posiciones originales exactas de los diodos con respecto al sistema de la cámara, para poder compararlas con las posiciones

reconstruidas. Dado que esto resulta casi imposible en la práctica, se realizó una simulación en Matlab para medir este error en forma teórica.

La simulación considera distintas posiciones y rotaciones para el cintillo en el espacio, calculando la posición 3D de los diodos y proyectándolos en una imagen según el modelo de cámara Pinhole (ecuación 2.1.1). La posición (x,y) de los diodos en la imagen es discretizada, lo cual significa una pérdida de información. A partir de estas proyecciones reconstruye la posición 3D de los diodos para determinar la posición del centro del cintillo y sus ángulos de rotación. Finalmente, se compara los valores reconstruidos con los originales, tomando el valor absoluto de las diferencias.

Se considera un cintillo como el de la sección 2.4.2, de 10.5 cm de radio. El cintillo se ubicó en 11 rectángulos a lo largo del eje Z, desde  $Z = 100$  cm a  $Z = 200$  cm cada 10 cm. Cada rectángulo va de  $X = -30$  cm a  $X = 30$  cm y de  $Y = -20$  cm a  $Y = 20$  cm y contiene 35 posiciones posibles, cada 10 cm (7x5). Finalmente, cada posición considera 5 rotaciones posibles de cada ángulo de  $-20^\circ$  a  $+20^\circ$  cada  $10^\circ$ , dando un total de 125 rotaciones compuestas posibles. El número total de instancias simuladas es 48,125.

La tabla 3.1.1 muestra los resultados de la simulación. Se presenta el error promedio y la desviación estándar del error de la posición (en centímetros) y los ángulos (en grados) de todas las instancias de cada rectángulo ubicado a una distancia Z de la cámara.

Tabla 3.1-1: Error teórico de las reconstrucciones.

| <b>Z (cm)</b> | <b>Pos (cm)</b> | <b>Pitch (°)</b> | <b>Roll (°)</b> | <b>Yaw (°)</b> |
|---------------|-----------------|------------------|-----------------|----------------|
| <b>100</b>    | 0.28 ± 0.19     | 0.36 ± 0.28      | 0.11 ± 0.09     | 0.34 ± 0.22    |
| <b>110</b>    | 0.30 ± 0.22     | 0.47 ± 0.33      | 0.12 ± 0.09     | 0.37 ± 0.24    |
| <b>120</b>    | 0.37 ± 0.26     | 0.52 ± 0.36      | 0.15 ± 0.11     | 0.43 ± 0.30    |
| <b>130</b>    | 0.44 ± 0.30     | 0.56 ± 0.42      | 0.14 ± 0.11     | 0.42 ± 0.30    |
| <b>140</b>    | 0.51 ± 0.35     | 0.51 ± 0.39      | 0.16 ± 0.12     | 0.49 ± 0.36    |
| <b>150</b>    | 0.54 ± 0.37     | 0.61 ± 0.40      | 0.17 ± 0.13     | 0.54 ± 0.37    |
| <b>160</b>    | 0.68 ± 0.47     | 0.79 ± 0.57      | 0.18 ± 0.14     | 0.64 ± 0.41    |
| <b>170</b>    | 0.78 ± 0.54     | 0.74 ± 0.48      | 0.20 ± 0.14     | 0.61 ± 0.43    |
| <b>180</b>    | 0.75 ± 0.55     | 0.78 ± 0.49      | 0.20 ± 0.15     | 0.66 ± 0.48    |
| <b>190</b>    | 0.93 ± 0.65     | 0.83 ± 0.59      | 0.21 ± 0.15     | 0.70 ± 0.49    |
| <b>200</b>    | 0.96 ± 0.72     | 0.81 ± 0.62      | 0.20 ± 0.15     | 0.69 ± 0.49    |

El error de las reconstrucciones aumenta con la distancia como es de esperar. Sin embargo, el error promedio de la posición se mantiene bajo 1 cm, y el de los ángulos bajo  $1^\circ$  cuando el usuario se encuentra a menos de 2m de la cámara.

En la simulación, el error de las reconstrucciones se debe mayormente a la pérdida de información producida al proyectar un punto 3D de coordenadas continuas en una imagen discreta. Aquí radica la importancia de la resolución de la cámara. Al eliminar la discretización de las proyecciones en la simulación, el error de las reconstrucciones converge a cero.

En la práctica, hay que sumar al error anterior el producido por imperfecciones en la construcción del cintillo. El cintillo considerado en la simulación es un cintillo “ideal” sin imperfecciones, de modo que esta nos entrega solo una cota inferior del error real.

### **3.2 Rapidez**

La implementación en C# del sistema de HT extendido con eliminación de la inclinación de la cámara activada, corre fluidamente a 100 hz, aprovechando al máximo la tasa de refresco de la cámara IR del Nintendo Wii Remote.

Las reconstrucciones 3D del sistema convergen en 5 a 6 iteraciones del método Newton-Raphson cuando el cintillo está ubicado de 100 a 200 cm de la cámara, utilizando un punto de partida fijo  $\mathbf{Z}_0 = (100, 102, 100)$  y un  $\varepsilon = 0.1$  (en centímetros). Reutilizando las soluciones anteriores como puntos de partida dinámicos, Newton-Raphson converge en 4 a 5 iteraciones para el mismo  $\varepsilon$ .

### 3.3 Robustez

El sistema de HT funciona en forma robusta en ausencia de ruido IR. Se considera ruido IR a cualquier luz infrarroja detectada por el Wii Remote que no corresponda a un diodo del cintillo. Las principales fuentes de ruido IR identificadas son:

- La luz del sol: contiene radiación IR que puede alcanzar al Wii Remote directamente a través de ventanas, o reflejándose sobre espejos u otras superficies.
- Reflexiones de los diodos del cintillo: espejos u otras superficies lisas ubicadas muy cerca del cintillo, pueden reflejar la luz IR de los diodos hacia Wii Remote.
- Suciedad del lente: puede distorsionar la luz IR recibida por el Wii Remote, dividiendo la luz de un diodo en varias luces, o uniendo varias luces en una sola.

Este ruido IR hace que el sistema de HT se confunda, identificando ruido como diodos del cintillo, lo cual produce reconstrucciones erróneas e inestables en el tiempo.

Finalmente, la robustez del sistema de HT extendido está limitada por la certeza del método utilizado por el Módulo de Tracking para encontrar los diodos de referencia. Éste método falla cuando el cintillo presenta rotaciones de pitch muy pronunciadas con respecto al sistema de referencia de la cámara, llevando a detecciones incorrectas del diodo de referencia, y en consecuencia, a reconstrucciones incorrectas del cintillo. Debido a esto, el sistema funciona en forma robusta sólo si el cintillo presenta rotaciones moderadas de pitch ( $\pm 30^\circ$ ).



#### 4. CONCLUSIONES

Se propuso un sistema de Head Tracking en el espectro infrarrojo que utiliza el Nintendo Wii Remote para detectar un cintillo con diodos emisores infrarrojos. El sistema detecta con precisión la posición y la orientación de la cabeza del usuario, presentando además dos características nuevas con respecto a trabajos anteriores:

1. Funciona sin requerir que el usuario mire de frente a la cámara, gracias a que el cintillo utilizado incorpora diodos en todo su perímetro.
2. Detecta y elimina la inclinación de la cámara, gracias a los acelerómetros del Wii Remote, obteniendo la inclinación absoluta de la cabeza con respecto a la Tierra.

Otras características destacables del sistema son:

- Bajo costo: El Wii Remote se puede obtener por un precio comparable al de una Webcam normal<sup>3</sup>, y el cintillo utilizado se puede fabricar por menos de 10 USD.
- Desempeño: Con sus 1024x768 pixeles de resolución y 100 Hz de refresco, el Wii Remote supera ampliamente a las webcams normales, que típicamente ofrecen 640x480 pixeles a 30 Hz. La resolución de la cámara incide directamente en la precisión de las reconstrucciones. Por otra parte, el sistema de HT funciona a 100 Hz aprovechando al máximo las capacidades del Wii Remote.
- Simpleza: El enfoque en el espectro IR utilizando el MOT engine del Wii Remote, simplifica altamente la implementación al no requerir mayor pre-procesamiento. Además, lo simple de la geometría del cintillo utilizado hace más fácil el análisis geométrico.
- Modularidad: El sistema propuesto ataca el problema descomponiéndolo en dos módulos: un módulo de tracking encargado de identificar los diodos del cintillo, y un módulo de reconstrucción, encargado de reconstruir su geometría en el

---

<sup>3</sup> Amazon.com lo vende en 34.96 USD en Agosto de 2009.

espacio. Este esquema es válido para cualquier sistema con un enfoque similar, y permite proponer mejoras fácilmente.

- Usabilidad: La salida del sistema de HT puede ser transmitida a otros programas mediante OSC, permitiendo usarla en otras aplicaciones.

En cuanto a los aspectos negativos del sistema, se considera que este cuenta con dos debilidades principales:

1. La simetría del cintillo: La parte frontal y trasera del cintillo lucen exactamente iguales para el sistema, dado que éste no tiene como diferenciar los diodos de referencia frontal y trasero. Para resolver este problema, se determinó que el usuario deba mirar a la cámara al comenzar el tracking, para contar con una situación inicial conocida. Este requerimiento es desventajoso por dos razones:
  - Hace que el sistema no sea suficientemente autónomo, ya que depende de un set-up inicial del usuario.
  - Introduce el riesgo de que se produzca un error permanente en el largo plazo. Por ejemplo, si el cintillo se sale momentáneamente del FoV de la cámara, se puede confundir en adelante el frente con el revés del cintillo.

La forma más simple de resolver este problema, es ubicar el diodo de referencia trasero, en la posición totalmente opuesta al delantero, es decir, bajo el plano de los diodos principales. El método para detectar los diodos de referencia sería el mismo, pero se diferenciarían por estar sobre o bajo los diodos principales.

2. La detección de los diodos de referencia: El Módulo de Tracking encuentra los diodos de referencia buscando un trío de diodos que forme un ángulo lo suficientemente recto. Este criterio funciona bien siempre y cuando el cintillo no presente rotaciones de pitch muy pronunciadas con respecto al sistema de la cámara. En caso contrario, el ángulo más recto puede formarse en otro trío de diodos, llevando a una detección incorrecta del diodo de referencia.

El criterio de los ángulos rectos para detectar los diodos de referencia restringe las rotaciones posibles de pitch que el cintillo puede adoptar para ser detectado correctamente, por lo cual se debe buscar una mejor alternativa.

## BIBLIOGRAFIA

- Hjelmas, E. y Low, B.K. (2001). Face detection: A survey. *Computer Vision and Image Understanding*, 83(3), 236-274.
- Murphy-Chutorian, E. y Trivedi, M.M. (2009). Head Pose Estimation in Computer Vision: A Survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(4), 607-626.
- Porta, M. (2002). Vision-based user interfaces: methods and applications. *International Journal of Human-Computer Studies*, 57(1), 27-73.
- Bérard, F. (1999). The perceptual window: head motion as a new input stream. *Proceedings of the 7th IFIP conference on Human-Computer Interaction (INTERACT)*, Edinburgh, Scotland, 238-244.
- Toyama, K. (1998). "look, ma—no hands!" Hands-Free Cursor Control with Real-Time 3D Face Tracking. *Proc. Workshop Perceptual User Interfaces*, 49-54.
- La Cascia, M., Sclaroff, S. y Athitso, V. (2000). Fast, Reliable Head Tracking under Varying Illumination: An Approach Based on Registration of Texture-Mapped 3D Models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(4), 322-336.
- Ryu, W. y Kim, D. (2007). Real-time 3D Head Tracking and Head Gesture Recognition. *16th IEEE International Conference on Robot & Human Interactive Communication*. August 26-29, Jeju, Korea. 169-172.
- An, K.H. y Chung, M.J. (2008). 3D Head Tracking and Pose-Robust 2D Texture Map-Based Face Recognition using a Simple Ellipsoid Model. *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*. Acropolis Convention Center. Nice, France, Sept. 22-26, 307-312.
- Morency, L.P., Rahimi, A., Checka, N. y Darrell, T. (2002). Fast Stereo-Based Head Tracking for Interactive Environments. *Proc. IEEE Int'l Conf. Automatic Face and Gesture Recognition*, 375-380.
- Yang R. y Zhang, Z. (2002). Model-Based Head Pose Tracking with Stereovision. *Proc. IEEE Int'l Conf. Automatic Face and Gesture Recognition*, 242-247.

Mohan, A., Duraiswami, R., Zotkin, D.N., DeMenthon, D. y Davis, L.S. (2003). Using Computer Vision to generate Customized Spatial Audio. *Proceedings of the 2003 International Conference on Multimedia and Expo (ICME)*, 3, 57-60.

DeMenthon, D. y Davis, L.S. (1995). Model-Based Object Pose in 25 Lines of Code. *International Journal of Computer Vision*. 15(1-2), 123-141.

Lee, J.C. (2008). Hacking the Nintendo Wii Remote. *IEEE Pervasive Computing*, 7(3), 39-45.

Wright, M. y Freed, A. (1997). Open Sound Control: A New Protocol for Communicating with Sound Synthesizers. *International Computer Music Conference*, Thessaloniki, Greece, 101-104.

Hartley, R. y Zisserman, A. (2000). Multiple View Geometry in Computer Vision. *Cambridge University Press*.

WiimoteLib - .NET Managed Library for the Nintendo Wii Remote. Disponible en: <http://www.brianpeek.com/blog/pages/wiimotelib.aspx>

Bespoke - .NET OSC Library. Disponible en: <http://www.bespokesoftware.org>

## **ANEXOS**

## ANEXO A: EL CINTILLO Y LOS DIODOS UTILIZADOS

Para construir el cintillo se utilizó diodos infrarrojos de 5mm Lumex modelo OED-EL-1L2, los cuales operan a una intensidad de 100 mA y emiten en un cono de 60°. La intensidad y el ángulo de emisión de los diodos son de especial importancia, ya que los diodos escogidos deben asegurar que la cámara pueda ver permanentemente 3 de ellos para poder realizar la reconstrucción 3D.

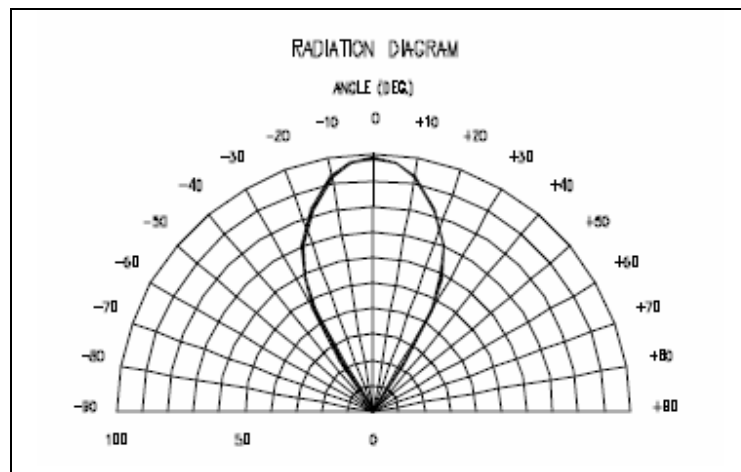


Figura A-1: Ángulo de emisión de los diodos utilizados.

Como los diodos del cintillo están dispuestos en círculo, sus haces de emisión apuntan en direcciones divergentes, por lo que para que la cámara pueda ver 3 al mismo tiempo, se requiere que sus conos de emisión coincidan en la posición que se encuentra. Un prototipo inicial del cintillo que utilizaba diodos de menor intensidad y menor ángulo de emisión (30°), falló por que la cámara no conseguía ver permanente los 3 diodos necesarios para la reconstrucción, a ninguna distancia de la cámara.

Para el marco del cintillo se utilizó una tira circular de PVC de 10.8 cm de radio, con 8 perforaciones equidistantes para los 8 diodos principales. Los diodos de referencia se ubicaron en dos extensiones de rectangulares de PVC adheridas al cintillo, como muestra la figura 2.3.2. Para alimentar el cintillo se creó un circuito en serie con los 10 diodos, utilizando un transformador de 12V (cada diodo requiere 1.2 V).



## ANEXO B: CALIBRACION DE LOS ACELERÓMETROS

El objetivo de este proceso es estimar el error de fábrica de los acelerómetros del Nintendo Wii Remote. Para esto se realizan 3 ensayos, en que se ubica el dispositivo descansando sobre cada uno de sus costados, sobre una superficie horizontal. El valor esperado de las aceleraciones registradas en cada uno los ensayos es, 1 en el eje normal a la superficie, y 0 en los otros dos. Midiendo los errores de cada ensayo, se puede estimar el error de fábrica de cada acelerómetro como el promedio de los errores de los 3 ensayos:

- Ensayo 1: Dispositivo descansando horizontalmente (eje Z paralelo a  $\vec{g}$ )
- Ensayo 2: Dispositivo descansando verticalmente (eje Y paralelo a  $\vec{g}$ )
- Ensayo 3: Dispositivo descansando lateralmente (eje X paralelo a  $\vec{g}$ )

Denotando  $a_{ij}$  la aceleración registrada por el acelerómetro  $i$  en el ensayo  $j$  y  $E(a_{ij})$  el valor esperado de dicha aceleración, se tiene que los valores de los valores esperados son:

$$\begin{aligned} E(a_{x1}) &= 0 & E(a_{y1}) &= 0 & E(a_{z1}) &= 1 \\ E(a_{x2}) &= 0 & E(a_{y2}) &= 1 & E(a_{z2}) &= 0 \\ E(a_{x3}) &= 1 & E(a_{y3}) &= 0 & E(a_{z3}) &= 0 \end{aligned} \quad (\text{B.1})$$

De esta manera los errores de fábrica de los acelerómetros pueden ser estimados como el promedio de las desviaciones del valor esperado en cada ensayo:

$$\begin{aligned} \varepsilon_x &= \frac{1}{3} \sum_{i=1}^3 (E(a_{xi}) - a_{xi}) \\ \varepsilon_y &= \frac{1}{3} \sum_{i=1}^3 (E(a_{yi}) - a_{yi}) \\ \varepsilon_z &= \frac{1}{3} \sum_{i=1}^3 (E(a_{zi}) - a_{zi}) \end{aligned} \quad (\text{B.2})$$