

Sistema de Head Tracking para Linux con demostraciones

Descripción de los objetivos del proyecto

El head-trackingⁱ es una técnica que se basa en obtener en tiempo real la posición de la cabeza (o la cara) del usuario respecto al display de nuestro sistema, para poder adaptar el punto de vista de forma dinámica, dando una ilusión de profundidad, mejorando la inmersión del usuario en la escena 3d.

Pese a que el head-tracking no es una técnica de aparición reciente, a día de hoy aun no hay aplicaciones comerciales de sobremesa que la apliquen, y posiblemente la situación siga así hasta que sistemas como el Project Natalⁱⁱ de Microsoft proporcionen a los programadores una plataforma estándar en la que apoyarse.

El objetivo de este proyecto es proporcionar una plataforma sencilla y flexible, que aporte capacidades de tracking a sistemas de sobremesa Linux usando dispositivos de bajo coste que se encuentran al alcance de cualquier usuario corriente, y que además, presente una interfaz versátil para los desarrolladores, ya sea para crear nuevas aplicaciones o adaptar el head-tracking a aquellas ya existentes.

Este proyecto constará de dos partes bien diferenciadas: por una parte el sistema de tracking, encargado de procesar los datos y proporcionárselos a la aplicación, y por otra, el conjunto de demostraciones que ilustran los posibles usos y funcionalidades del sistema.

La plataforma de tracking constará a su vez de 2 partes: el servidor, que se encargará de procesar los datos para obtener la posición del objeto que interese (normalmente la cara o los ojos del usuario, pero no necesariamente), y transferir esta información a la librería cliente, que proporcionará la información y funciones necesarias para que la aplicación pueda usarla. En un principio el sistema podrá usar webcams (mediante detección de caras con OpenCVⁱⁱⁱ o detección de marcadores de ARToolkit^{iv}) o mandos de Wii (detectando marcadores infrarrojos), pero con capacidad para implementar otros medios de entrada en un futuro. Será posible combinar más de un dispositivo de entrada para poder calcular mejor la profundidad (mediante triangulación) o para aumentar el área útil de detección. El servidor y el cliente usarán VRPN^v para comunicarse, lo que aporta la ventaja de poder integrarse con otros sistemas que usen esas librerías y la de enviar los datos a través de la red.

También se incluirá en la memoria del proyecto una comparativa del rendimiento e información que aporta cada dispositivo o combinación de dispositivos usados por el servidor.

Las demostraciones por su parte, ilustrarán las capacidades y posibilidades del sistema: Adaptación de un visor de modelos 3d para ajustar la perspectiva según el punto de vista del usuario, o navegación por Google Street View ajustando la vista simplemente moviendo la cabeza, o un pequeño simulador de cómo podría verse un escritorio 3d usando esta técnica. Otras demos harán hincapié en capacidades mas específicas del sistema, como poder seguir a más de un usuario u objeto a la vez, o poder adaptar múltiples pantallas.

Otras demos, menos convencionales, consistirán en adaptar la técnica a un editor de modelos 3d (Blender^{vi}), reproducir los resultados del proyecto ChairIO^{vii} (usar una silla de oficina como input) sin hardware adicional, detectar las manos o los dedos del usuario para poder simular una superficie táctil o ampliar el uso tradicional del mando de wii para permitir al usuario apuntar usando ray tracing, teniendo en cuenta tanto la dirección a la que apunta como la posición del apuntador y los objetos virtuales en la escena.

Objetivos ya cubiertos

A día de hoy está disponible en el repositorio^{viii} del proyecto una versión funcional del cliente, que gestiona de manera transparente, gracias al uso de threads, tanto la conexión al servidor mediante VRPN como la actualización de los datos. La versión actual permite que el programador acceda a los últimos datos para tratarlos como sea necesario, pero aporta también una pequeña interfaz de llamadas para substituir a las originales de OpenGL, manteniendo casi intacta la estructura original del programa, y que gestionan de manera interna el uso de un Glfrustum asimétrico y el cambio de posición de la cámara.

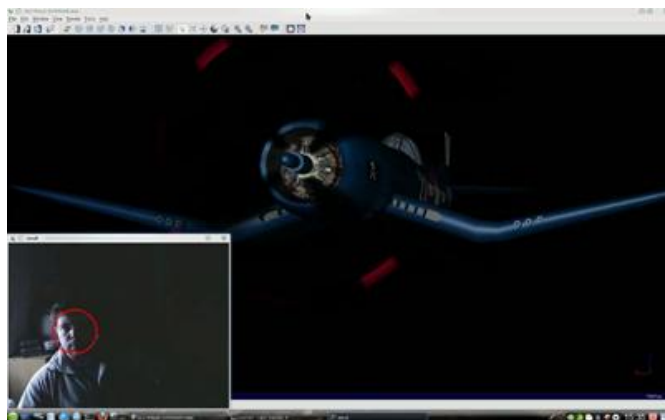


Imagen del video de la demo de GLC Player usando head-track que se puede encontrar en el blog del proyecto.

Así mismo, también está disponible la demostración ya acabada del visor de modelos 3d, en la que se ha adaptado el GLC Player^{ix} para que aplique el cambio dinámico de perspectiva usando las funciones de la librería cliente.

El resto del trabajo se ha invertido en investigar y familiarizarse en el uso del software externo que se usará en el proyecto (OpenCV, las librerías cwiid para la comunicación con el wiimote, el uso de los trackers de

VRPN, etc.), y aunque no formen parte del proyecto propiamente dicho, en la carpeta *tests* del repositorio, pueden encontrarse piezas de código de prueba que más tarde se integrarán en el proyecto. Por ejemplo, las versiones básicas de los servidores que se están usando a día de hoy para las pruebas (tanto los que envían datos prefijados como los que infieren la posición del usuario gracias a un wiimote o a la detección de caras de openCV), se encuentran ahí.

En el blog del proyecto^x se irán publicando los avances en el desarrollo.

Planificación del trabajo que falta por hacer

Quedan aproximadamente 2 meses (9 semanas) para la entrega del proyecto.

Durante las dos primeras semanas de diciembre deberá completarse la primera versión del servidor, que acepte tanto wiimote (con 1 y 2 leds) como openCV, con sus opciones de configuración.

La tercera semana de diciembre se dedicará a acabar las demos simples (Google Street View, la simulación del escritorio 3d, elementos de la escena que reaccionan ante el usuario).

La cuarta semana se dedicará a adaptar el uso para multiusuario y multipantalla.

La quinta semana se ampliará el servidor al uso de artoolkit,

La sexta y la séptima semana se dedicarán a las demos avanzadas (integración con blender, simulación de superficie táctil, ChairIO y la demo del ray-tracing).

Por último, las 2 últimas semanas servirán para completar y revisar la documentación, así como cualquier parte que pueda quedar pendiente.

Carlos Antonio Andújar Gran
director del proyecto

Pablo Campos Sáez
estudiante

ⁱ Para entender la idea de Head Tracking, lo mejor es ver el video que inspiró este proyecto:
<http://www.youtube.com/watch?v=Jd3-eiid-Uw>

ⁱⁱ Project Natal de Microsoft: <http://www.xbox.com/en-US/live/projectnatal/>

ⁱⁱⁱ OpenCV, librería de visión por computador: <http://opencv.willowgarage.com/wiki/>

^{iv} ARToolkit: <http://www.hitl.washington.edu/artoolkit/>

^v Virtual Reality Private Network: <http://www.cs.unc.edu/Research/vrpn/>

^{vi} Blender, editor de modelos 3d Open Source: <http://www.blender.org/>

^{vii} ChairIO: <http://imve.informatik.uni-hamburg.de/projects/chairio>

^{viii} Repositorio de este proyecto, donde está el código actualizado:
<http://code.google.com/p/linuxtrackingpfc/>

^{ix} GLC Player, visor de modelos 3d usado en la primera demo: <http://www.glc-player.net/>

^x El blog de este proyecto, donde se publican los avances y los videos de las demostraciones:
<http://plagaslair.blogspot.com/>