CEIG - Spanish Computer Graphics Conference (2018)
I. García-Fernández and C. Ureña (Editors)

Experiences from the development of a prototype of an Spatial Augmented Reality system

José-Luis Cárdenas-Donoso†, Ángel-Luis García-Fernández, Francisco-de-Asís Conde-Rodríguez, Carlos-Javier Ogáyar-Anguita

Departamento de Informática. Universidad de Jaén

Abstract

This paper describes the hardware and software of a prototype of a portable Spatial Augmented Reality system. This prototype is completely autonomous and projects the information on the real world, so that it is visible for a group of users. The system was built using mostly low-cost hardware (Raspberry Pi, plus a camera module and an inertial measurement unit), and the software was developed using C++ and OpenGL ES 2.0.

CCS Concepts

•Human-centered computing → ***Mixed / augmented reality;*** ***•Computing methodologies*** → *Camera calibration;* ***•Computer systems organization*** → *Embedded hardware;*

1. Introducción

1.1. Contexto

La Realidad Aumentada (RA) es una variante de la Realidad Virtual consistente en la modificación de un entorno real a través de la adición de elementos multimedia generados por computador. En su forma más habitual, la RA se aplica mediante la superposición sobre imágenes reales (estáticas o vídeo) de información generada por computador, ya sean personajes virtuales, información textual o gráfica, etcétera.

En los últimos años la RA ha tenido una gran expansión, dado el auge de dispositivos móviles como *smartphones* y *tablets*, que combinan el hardware y la potencia de procesamiento suficiente para ejecutar aplicaciones sencillas de RA. Esta tecnología se aplica en muchos campos: desde videojuegos a aplicaciones con fines educativos, médicos, científicos, etcétera.

La RA basada en dispositivos móviles tiene un inconveniente, y es que el tamaño de las pantallas impide trabajar cómodamente con grupos numerosos de usuarios. Algunos productos como las HoloLens de Microsoft [Mic18] están pensados para ser utilizados por un único usuario, cubriendo en mayor o menor medida su cara y su cabeza (lo cual puede llegar a ser incómodo para algunas personas), y en caso de ser necesario su uso simultáneo por un grupo, se requeriría de tantos dispositivos como usuarios hubiera; dependiendo del tipo de dispositivo, el coste económico de esta solución puede ser verdaderamente elevado.

Para superar este inconveniente, se plantea la separación de la tecnología de visualización del usuario, dando lugar a la Realidad Aumentada Espacial (Spatial Augmented Reality, SAR) [BR05]. Los sistemas SAR proyectan las imágenes sobre pantallas de distintos tipos, e incluso sobre la misma realidad. En este trabajo se plantea el prototipo de un sistema SAR portable, elaborado a partir de componentes de bajo coste, y que proyecta sobre el mundo real la información calculada. Este prototipo se diseñó en el ámbito de un trabajo de fin de grado de Ingeniería Informática.

Podría decirse que el funcionamiento general de los sistemas de RA, y por extensión de los sistemas SRA, se basa en dos subsistemas: un subsistema se encarga de generar los contenidos que se añaden a la realidad, y para que éste funcione correctamente, se hace necesario otro subsistema de captación, a través del cual se obtiene la posición y orientación desde la que se percibe la realidad que va a ser aumentada. Para determinar esta posición y orientación se puede recurrir a distintos tipos de tecnologías, tales como GPS, brújulas, marcadores (códigos QR, imágenes objetivo, marcadores fiduciarios...), posicionamiento basado en redes inalámbricas, etcétera; como es de esperar, la precisión y robustez varía con cada una de ellas, por lo que se suele recurrir a una combinación de varias.

Algunas soluciones relacionadas con el trabajo que aquí presentamos han sido planteadas en trabajos previos, como el de Raskar y otros [RBvB*04], que utilizan etiquetas RFID con sensores de luz para marcar paquetes en el contexto de un almacén, o el de Schöning y otros [SRK*09], que utilizan SAR para proyectar sobre un mapa de una ciudad información relevante sobre puntos de interés.

† Autor para correspondencia. E-mail: jcdonoso@ujaen.es

© 2018 The Author(s)
Eurographics Proceedings © 2018 The Eurographics Association.

delivered by EUROGRAPHICS DIGITAL LIBRARY www.eg.org diglib.eg.org

DOI: 10.2312/ceig.20181158

1.2. Objetivo

El objetivo principal para este proyecto ha sido la creación de un sistema SRA autónomo y portable, basado en componentes de bajo coste. El desarrollo ha implicado no sólo la creación de un dispositivo hardware portable y fácil de usar, sino también del software que gestione la generación y adecuada proyección del contenido virtual sobre el mundo real que lo rodea.

2. Prototipo Hardware

El prototipo diseñado está formado por una placa Raspberry Pi a la que hemos añadido como subsistema de captación un medidor inercial (IMU) y una cámara. Hemos diseñado y fabricado con una impresora 3D una carcasa para contener la Raspberry Pi, la IMU y la cámara. Posteriormente, hemos unido la carcasa a un videoproyector portátil mediante un tornillo de calibre estándar, similar al usado para los anclajes de los trípodes (Figura 1).

Figure 1: *Prototipo del dispositivo*

El funcionamiento general del prototipo se ilustra en la Figura 2:

1. La cámara toma imágenes del mundo real a intervalos regulares. Estas imágenes se procesan en la Raspberry Pi para buscar marcadores que permitan posicionar el dispositivo en el entorno. En los instantes en los que no se toma una foto o no se encuentran marcadores en la imagen, se utilizan los valores proporcionados por la IMU para calcular sólo la orientación. En la Sección 3.1 se detalla cómo se ha realizado este proceso.
2. La Raspberry Pi genera la imagen que se va a proyectar a partir de un modelo 3D de los datos.
3. Se proyecta la imagen a través del videoproyector, superponiendo la información en el lugar correcto.

2.1. Raspberry Pi

Raspberry Pi es una familia de ordenadores en una placa, ampliamente utilizados en tareas relacionadas con IoT y multimedia. Para este proyecto decidimos utilizar la versión 3, equipada con un procesador ARM Cortex-A53 de 4 núcleos a 1,2 GHz, y con una GPU Broadcom VideoCore 4, con soporte para OpenGL ES 2.0, OpenVG y OpenMax. Como sistema operativo emplea la distribución de GNU/Linux Raspbian.

Para alimentar la Raspberry Pi se puede utilizar cualquier fuente de 5 V con más de 2 amperios. En este caso, para conseguir que el dispositivo sea portátil, lo conectamos a una batería con salida USB de 10000 mAh, con lo que conseguimos una autonomía de varias horas.

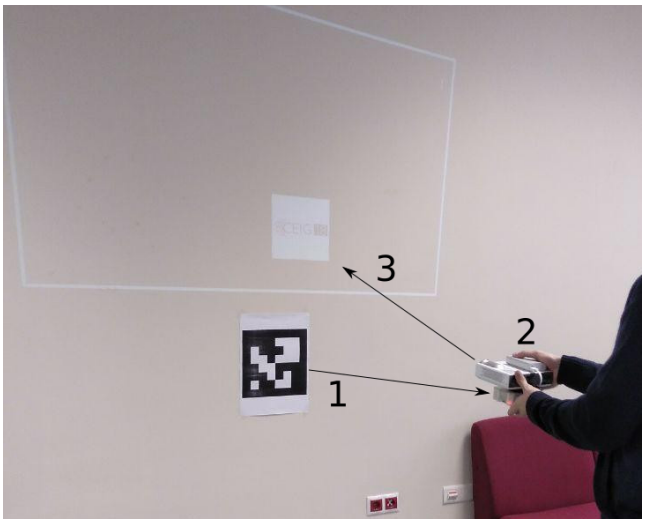

Figure 2: *Funcionamiento básico del dispositivo*

2.2. IMU

La IMU (Inertial Measurement Unit, Unidad de Medición Inercial) permite obtener la aceleración y velocidad angular del dispositivo. La utilizamos para conocer la orientación del dispositivo en el espacio.

El modelo que hemos utilizado para el dispositivo es el MPU-6050, un chip que integra un acelerómetro y un giroscopio. Las lecturas que proporciona cada sensor vienen codificadas en dos vectores de 3 enteros (aceleración y velocidad angular en los 3 ejes cartesianos). Estos valores deberán ser escalados según el rango de captura de datos, que puede seleccionarse entre 2G, 4G, 8G y 16G para el acelerómetro y 250, 500, 1000, 2000 grados/s para el giroscopio [Inv13]. En nuestro caso, probamos distintos valores y decidimos utilizar los rangos de 2G para el acelerómetro y 250 grados/s para el giroscopio, porque el uso previsto del dispositivo se basa en movimientos suaves.

Es posible calcular los cambios de posición utilizando el valor de la aceleración lineal, obtenida una vez sustraído el valor de la aceleración por la acción de la gravedad. De esta aceleración se puede obtener la velocidad integrando respecto al tiempo, y a partir de la velocidad obtener la posición de manera análoga.

2.3. Cámara

Hemos utilizado el módulo de cámara oficial de Raspberry Pi, en su versión 2 [Ras]. Su resolución máxima es de 8 megapíxeles, su distancia focal es de 3.04 mm, y su campo de visión es de 62.2^{o} de ancho por 48.8^{o} de alto. Permite grabar vídeo 1080p a 30 fps, 720p a 60 fps y VGA a 90 fps. El enfoque de la cámara es manual, por lo que al montarla en la carcasa la enfocamos de forma que permitiera

© 2018 The Author(s)
Eurographics Proceedings © 2018 The Eurographics Association.

reconocer los marcadores a una distancia razonable. La cámara se conecta a la Raspberry Pi mediante su puerto CSI (*Camera Serial Interface*).

2.4. Proyector

Para que el dispositivo fuera realmente funcional, era requisito indispensable que el videoproyector funcionara de manera autónoma, y que su luminosidad fuera suficientemente alta. Tras valorar distintas opciones, seleccionamos el ASUS P3B. Este proyector dispone de una batería de 12000 mAh y su luminosidad es de 500 lúmenes cuando funciona con batería. Su autonomía es de hasta 3 horas, pesa 750 gramos y tiene tiro corto, con lo que se obtiene un mayor ángulo de proyección a poca distancia de la pared. Su precio varía entre los 520 y 600 euros.

3. Prototipo Software

Hemos utilizado C++ para desarrollar el software que controla el dispositivo. Para la interfaz gráfica hemos utilizado Qt (versión 5.6); nos hemos apoyado en las bibliotecas OpenGL ES 2.0 para la generación de la imagen a proyectar, y OpenMax (a través de otras bibliotecas que proporcionan una capa de abstracción) para el control y obtención de los datos de la cámara. A continuación se detallan los componentes del sistema.

3.1. Subsistema de captación

Como ya comentamos en la introducción, el subsistema de captación se encarga de determinar la posición y orientación del dispositivo, de forma que se puedan sincronizar los elementos del mundo real con los elementos virtuales proyectados. Es requisito esencial que la precisión del cálculo sea alta, y que la latencia sea mínima.

El subsistema de captación implementado está formado por:

- Los drivers de la IMU: Desarrollados sobre la interfaz para plug-ins de sensores de Qt, utilizan el bus I2C para la comunicación con el chip MPU-6050.
- El módulo de acelerómetro y giroscopio: Permite la calibración de la IMU y la obtención de los valores proporcionados por ésta para obtener la orientación del dispositivo.
- La cámara y los marcadores: Utilizados para la calibración de la cámara y la estimación de la pose del dispositivo utilizando los marcadores reconocidos en la imagen.

Hemos implementado dos aplicaciones auxiliares a la aplicación principal: una con herramientas para la calibración de la IMU y otra para la calibración de la cámara utilizando un tablero de marcadores. Estas calibraciones se guardan en un fichero de configuración, de manera que no sea necesario calibrar cada vez que se inicie la aplicación de SRA.

3.1.1. Obtención de la pose con la IMU

Previo a la obtención de la pose (esto es: la posición y la orientación del dispositivo), es preciso calibrar el acelerómetro y el giroscopio de la IMU.

Hemos implementado las rutinas de calibración del acelerómetro utilizando la biblioteca Eigen (versión 3) para resolver los cálculos matriciales. Partimos del supuesto que los valores reales de aceleración son una transformación lineal en un espacio de 3 dimensiones (al que se añade la coordenada homogénea) de los valores brutos que proporciona el chip. Para conocer la matriz de transformación, primero se obtienen los valores brutos de algunos vectores de aceleración conocidos: los medidos por la gravedad cuando se orienta el dispositivo en distintas direcciones, que para simplificar, estarán alineadas con los ejes del espacio ortonormal [STM14].

Hemos simplificado la calibración del giroscopio, pues no disponemos más que de una velocidad angular de referencia: el dispositivo estático tiene velocidad angular 0 en todos los ejes. Por este motivo, consideramos el valor de velocidad angular bruto obtenido con el dispositivo en reposo como sesgo (es decir, se resta a cualquier medición).

Una vez calibrados el acelerómetro y el giroscopio, podemos combinar el valor obtenido por ambos sensores para calcular la orientación del dispositivo:

- Del giroscopio podemos obtener la orientación integrando el valor de la velocidad angular. Puesto que las lecturas tomadas contienen un error, este error se va acumulando a lo largo del tiempo, provocando una deriva, esto es, el valor estimado de la orientación se aleja cada vez más del real. Por este motivo, este método para calcular la orientación no es viable a largo plazo.
- El acelerómetro nos indica movimiento, pero también el valor de la gravedad. Este valor de la gravedad puede proporcionarnos también información sobre la inclinación del dispositivo, pero no es muy fiable a corto plazo, ya que cualquier movimiento modifica el vector de aceleración, que en vez de ser la gravedad, pasa a ser la suma de ésta con la aceleración debida al movimiento.

Hemos combinado ambos valores mediante un filtro complementario, asignando pesos a los valores proporcionados por los dos sensores. También es posible utilizar filtros más complejos, como el filtro de Kalman.

3.1.2. Obtención de la pose con la cámara

Este sistema consiste en utilizar marcadores visuales, y obtener la pose de la cámara respecto a estos marcadores usando técnicas de visión artificial.

Para que esta técnica funcione, es preciso realizar una calibración previa de la cámara. En nuestro caso, hemos empleado la biblioteca OpenCV, junto con el módulo de marcadores ARUCO para realizar la calibración.

La precisión y rapidez en el cálculo de la pose con este sistema se ve afectada por la resolución de la imagen y la proporción de ésta que está ocupada por los marcadores: a más ocupación, más precisión, pero también más retardo; pasado cierto umbral, el incremento de la precisión no compensa el retardo en el cálculo. Experimentalmente hemos comprobado que una imagen cuadrada de 1024 píxeles de lado es suficiente para calcular la pose usando marcadores de 28 centímetros de lado, visualizados a varios metros de distancia.

En el cálculo de la orientación se llega a producir ruido considerable, que aumenta con la distancia a los marcadores. En cambio, los valores de posición son bastante precisos.

© 2018 The Author(s)
Eurographics Proceedings © 2018 The Eurographics Association.

Para generar el mapa de marcadores y así integrarlos todos en el mismo espacio, hemos desarrollado también una extensión para Blender, que permite colocarlos en el espacio 3D y exportarlos a un fichero para posteriormente imprimirlos.

3.2. Generación de la imagen

Para la generación de las imágenes a proyectar hemos utilizado la biblioteca Qt compilada con soporte para aceleración hardware en dispositivos Raspberry Pi. Esto además permite utilizar directamente el framebuffer, sin necesidad de pasar por el entorno X11 para el acceso al hardware gráfico.

Partiendo de la escena 3D a proyectar que está almacenada en el dispositivo, calculamos la proyección bidimensional siguiendo los pasos del pipeline clásico de visualización. Utilizamos la información de la pose capturada (sección 3.1) para calcular la matriz de modelado que se aplica a la escena, mientras que para la matriz de proyección es necesario conocer las propiedades ópticas del videoproyector si queremos que las dimensiones de la imagen proyectada se correspondan con la realidad.

4. Resultados y conclusiones

Fruto del trabajo realizado, obtuvimos un prototipo funcional que cumple con los objetivos iniciales del proyecto. No obstante, la experiencia no fue totalmente satisfactoria debido a los problemas encontrados, algunos de los cuales no pudimos resolver al estar condicionados por el hardware:

- Hemos tenido que hacer los marcadores bastante grandes, con lo que producen cierta incomodidad a la vista.
- No ha sido posible utilizar el acelerómetro para calcular la pose debido al excesivo ruido en las mediciones. No obstante, sí hemos podido utilizar sus lecturas para calcular la orientación, porque este ruido no afecta al filtro complementario. Por otra parte, hemos comprobado que el error acumulativo en el giroscopio calibrado y sin filtro complementario tarda varios minutos en ser apreciable.
- El fabricante del videoproyector que elegimos no publica los datos sobre la óptica del mismo, y esto nos dio bastantes problemas. En particular, pudimos apreciar que el frustum de proyección no es simétrico respecto al eje horizontal, lo que hace que las imágenes generadas no se correspondan totalmente con el mundo real. Este efecto es apreciable sobre todo en rotaciones alrededor del eje Z del dispositivo (profundidad), en las que la imagen rota como debería, pero no queda fija en un punto.

5. Trabajo futuro

A continuación se enumeran algunas funcionalidades adicionales o mejoras a algunas existentes que han surgido a lo largo del desarrollo del proyecto:

- Creación de una interfaz de usuario para lanzar aplicaciones o herramientas de configuración.
- Utilización de técnicas de *Motion Tracking* a partir de la imagen de la cámara del dispositivo. Esto hará que se reduzca el número de marcadores necesarios, ya que podremos calcular los movimientos relativos de la cámara, reduciendo así el uso de los marcadores al cálculo de la posición absoluta.
- Mejor integración del posicionamiento obtenido de la IMU y la cámara, de forma que la orientación obtenida con la cámara pueda corregir la del IMU mediante un filtro. En la implementación actual podemos seleccionar una u otra, pero no las dos al mismo tiempo.
- Simplificación del modelado de los datos que se proyectan: actualmente el mapa de marcadores se proporciona en el archivo generado por la extensión para Blender y los datos a proyectar en el código de la aplicación, creando los objetos manualmente. Una solución sería crear un formato de fichero para proporcionar todos los datos de una vez.

6. Agradecimientos

Este trabajo ha sido parcialmente subvencionado por la Unión Europea (Fondos FEDER), el Ministerio de Economía y Competitividad de España (proyecto TIN2014-58218-R) y el Ministerio de Economía, Industria y Competitividad de España (proyecto TIN2017-84968-R).

References

[BR05] BIMBER O., RASKAR R.: *Spatial Augmented Reality: Merging Real and Virtual Worlds*. AK Peters, 2005. 1

[Inv13] INVENSENSE INC.: Mpu-6000 and mpu-6050 product specification revision 3.4, 2013. URL: https://www.invensense.com/wp-content/uploads/2015/02/MPU-6000-Datasheet1.pdf. 2

[Mic18] MICROSOFT CORP.: Microsoft HoloLens, 2018. URL: https://www.microsoft.com/es-es/hololens. 1

[Ras] RASPBERRY PI FOUNDATION: Raspberry Pi Documentation - Camera module. URL: https://www.raspberrypi.org/documentation/hardware/camera/README.md. 2

[RBvB*04] RASKAR R., BEARDSLEY P., VAN BAAR J., WANG Y., DIETZ P., LEE J., LEIGH D., WILLWACHER T.: RFIG lamps: interacting with a self-describing world via photosensing wireless tags and projectors. *ACM Transactions on Graphics 23*, 3 (2004), 406–415. 1

[SRK*09] SCHÖNING J., ROHS M., KRATZ S., LÖCHTEFELD M., KRÜGER A.: Map torchlight: a mobile augmented reality camera projector unit. In *CHI 2009* (2009). 1

[STM14] STMICROELECTRONICS: Application note AN4508: Parameters and calibration of a low-g 3-axis accelerometer, 2014. URL: http://www.st.com/content/ccc/resource/technical/document/application_note/a0/f0/a0/62/3b/69/47/66/DM00119044.pdf/files/DM00119044.pdf/jcr:content/translations/en.DM00119044.pdf. 3

© 2018 The Author(s)
Eurographics Proceedings © 2018 The Eurographics Association.