

URBANMIX

PORTALÉS RICART, Cristina. Laboratorio de Luz, BB. AA., Universidad Politécnica de Valencia, camino de Vera s/n 46022, Valencia. criporri@doctor.upv.es

GINER MARTÍNEZ, Francisco. Laboratorio de Luz, BB. AA., Universidad Politécnica de Valencia, camino de Vera s/n 46022, Valencia. fginerm@pin.upv.es

SANMARTÍN PIQUER, Francisco Javier. Laboratorio de Luz, BB. AA., Universidad Politécnica de Valencia, camino de Vera s/n 46022, Valencia. frasanpi@pin.upv.es

Palabras clave: realidad aumentada, tiempo real, espacio abierto.

1. INTRODUCCIÓN

“Realidad aumentada”, se puede definir como la amplificación de la capacidad sensorial de la percepción del mundo que nos rodea mediante el uso de dispositivos que superponen elementos virtuales a la imagen real.

La aplicación Urbanmix, nos posibilita este concepto a escala urbana y en exteriores usando un sistema propio de realidad aumentada, que se basa en modelos 3D y el programa Max/MSP Jitter [2] de edición de video en tiempo real.

A través de la aplicación Urbanmix, podemos acceder a una visión de la ciudad o espacio en el que nos encontremos, ampliada con algunos edificios o monumentos provenientes de otras. Con ello, y utilizando escalas 1:1 en el sistema, podemos realizar fácilmente comparaciones reales y así tomar verdadera conciencia de la magnitud de los edificios que visualizamos.

Ante la gran dimensión de algunos de los edificios que podemos introducir en el sistema, es necesario ubicarlos a una distancia lo suficientemente grande como para que el sistema pueda tomar las referencias necesarias para ubicarlo correctamente y su visualización continúe siendo creíble; por la falta de espacio en las ciudades lo normal es que algunos edificios reales se superpongan generalmente a los virtuales, por una superposición de canales de video, con lo que nos encontramos con un problema de oclusión de formas reales a virtuales que no se corresponde con lo habitual en estos sistemas, oclusión de formas virtuales a reales. El problema se resuelve con una máscara tridimensional de video que actualiza sus datos en tiempo real, que permite al usuario moverse a su antojo dentro del área acotada en donde se desarrolle la aplicación, y continuar viendo la inmersión del edificio o monumento escogido en el espacio precargado desde cualquier punto de vista.

2. ANTECEDENTES

Urbanmix es un proyecto en continuo desarrollo. Nuestro primer prototipo [11] estaba pensado para su ejecución dentro de un único espacio, el jardín de la escuela de BB.AA. de la Universidad Politécnica de Valencia. Actualmente, pensamos que el proyecto puede ser fácilmente trasladable a cualquier espacio abierto que sea susceptible de ser medido y modelado. Esta tarea se facilita enormemente gracias a la herramienta de libre distribución Google Earth (ver apartado 3.3.), con la que es posible adquirir la planta y ubicación relativa de los necesarios modelos 3D de casi cualquier ciudad del mundo; pero también, por el empleo de una serie de pegatinas/marcas

(fácilmente transportables) para realizar cálculos de la posición del usuario dentro de la aplicación mediante técnicas de visión por computador (ver apartado 3.2.).

3. METODOLOGÍA

Para una correcta integración de los edificios reales con los virtuales, es necesario realizar los siguientes pasos:

- En primer lugar, se debe definir el sistema de realidad aumentada, es decir, los dispositivos que intervendrán en la aplicación.
- Después, se definirá y modelará el escenario virtual, que se superpondrá al real, y los objetos 3D que se incluirán en el sistema. Para resolver el problema de la oclusión, el sistema aplica una máscara sustractiva de video entre ambos modelados; la oclusión se da cuando un objeto real se interpone entre el objeto virtual y el punto de vista del usuario.
- Finalmente, el sistema debe ser capaz de calcular en tiempo real la posición y orientación del usuario que, en este caso, coincide con el centro de proyección de la cámara.

A continuación se explican todos estos pasos con detalle.

3.1. Sistema de realidad aumentada

En este tipo de aplicaciones, se necesita de un dispositivo que permita al usuario percibir la nueva realidad creada. Estos dispositivos se denominan *interfaces*, y a través de ellos se muestran superpuestas la realidad y la ficción. Generalmente, se suelen utilizar unas gafas especiales, conocidas como HMD (Head Mounted Display), aunque también podemos encontrar diversas aplicaciones en las que se hace uso de pantallas (bien de ordenadores portátiles o de tablet pc). Ambos dispositivos tienen sus ventajas e inconvenientes. Una de las grandes ventajas que ofrece una pantalla, es que puede ser visualizada al mismo tiempo por varios usuarios, permitiendo un entorno colaborativo. Sin embargo, al tener lugar nuestra aplicación en un espacio abierto, optamos por la elección de un HMD, principalmente porque una pantalla de ordenador se vería afectada por los reflejos de la luz solar, impidiendo una correcta percepción de la nueva realidad. Además, las HMD son extremadamente ligeras en comparación con las pantallas, y ofrecen también una inmersión más creíble del usuario dentro de la aplicación.

Otro dispositivo indispensable en nuestra aplicación es una cámara, cuyo centro de proyección, en lo posible, se debe hacer coincidir con el punto de vista del espectador; es por ello que la cámara deberá situarse lo más cerca posible de los ojos del usuario. Esta cámara se encarga de registrar nuestro entorno en tiempo real, para posteriormente la imagen-vídeo ser procesada por el sistema, el cual nos ofrecerá (a través del HMD) dicha imagen-vídeo superpuesta con los objetos virtuales creados por ordenador. La resolución de la cámara deberá ser óptima, ni demasiado alta (para permitir el proceso en tiempo real) ni demasiado baja (para no ofrecer una imagen demasiado pobre). En nuestro caso utilizamos una Fire-i de Unibrain [15] con una resolución de 640 x 480 píxeles.

Otro factor a tener en cuenta es la posición y orientación del usuario dentro de nuestra aplicación, que se llevará a cabo en tiempo real mediante sensores. En nuestro caso, hemos utilizado la combinación de un sensor visual y uno inercial (ver apartado

3.4.). El sensor visual es la cámara anteriormente descrita; como sensor inercial hemos utilizado el MT9 de Xsens [14].

Finalmente, el usuario deberá cargar consigo un ordenador portátil en una cómoda mochila. Este se encargará de procesar toda la información recibida en tiempo real y realizar las operaciones necesarias para crear el nuevo entorno que verá el usuario.

En la siguiente figura pueden apreciarse todos los dispositivos anteriormente descritos.



Figura 1. Dispositivos de realidad aumentada utilizados en Urbanmix.

3.2. Resolución de la oclusión

Para alcanzar una integración apropiada, debe ser resuelto el problema de la oclusión. Este problema se presenta cuando el canal en el que aparecen los objetos virtuales se superpone al canal de video de la cámara, entonces los objetos virtuales aparecen superpuestos a la realidad.

Si pretendemos que los objetos virtuales se integren en una trama urbana debemos poder escoger qué parte de los objetos virtuales ha de quedar visible, y cual ocultar para que dejen ver a través la realidad que se supone que ha de presentarse en un plano más cercano a ellos.

Diversas metodologías se han probado para resolver este sistema [3; 4; 9]. En Urbanmix, este problema ha sido solucionado con una aplicación realizada con el programa de edición de video en tiempo real Max/MSP/Jitter [2], donde un modelo virtual de la ciudad funciona como una máscara 3D que oculta partes del modelo virtual del edificio integrado según la posición del usuario. La oclusión es solucionada dando al modelo 3D de la ciudad algunas características especiales, de manera tal que aparece transparente pero oculta aquellas partes del modelado posterior a las que se superpone desde el punto de vista del usuario.

Por lo tanto, la imagen video aparece donde el modelo 3D de la ciudad debería estar visible, ocultando solamente esas partes del edificio virtual que son ocluidos por los verdaderos. Finalmente, el modelo 3D y la imagen video se renderizan con OpenGL mediante el objeto *jit.gl.render* y se muestran en la ventana de la salida llamada *jit.window*.

En la siguiente figura podemos ver el funcionamiento de la oclusión. Como ejemplo se ha modelado el entorno de la Puerta del Sol de Madrid, que es donde se desarrollará la aplicación; como objeto virtual perteneciente a otra ciudad, se ha modelado el edificio Chrysler, de Nueva York. Para entender mejor el proceso hemos superpuesto estos modelados a una imagen que no se corresponde con los modelados cargados, esta

imagen la recogería la cámara web. En segundo lugar, vemos el resultado al aplicar el efecto transparencia y asignar la máscara sustractiva, quedando visible la parte del Chrysler que no quedaría oculta por el modelado.

En último lugar, vemos el efecto que se produce cuando hacemos coincidir el modelado virtual de la ciudad (transparente también) con la imagen recogida por la cámara web.



Figura 2. Funcionamiento de la oclusión.

3.3. Modelado 3D

Como ya hemos apuntado en el anterior apartado de la oclusión, para que el sistema funcione, es necesario contar con un par de modelados que han de funcionar en canales distintos, el del entorno en el que nos movamos, generalmente urbano, y que ha de funcionar como máscara, y el del edificio o monumento que queramos introducir en él.

El espacio urbano en el que vayamos a hacer funcionar el sistema, necesario para poder contemplar a escala real los modelos introducidos, se ha modelar tridimensionalmente. Para ello partimos de vistas aéreas de esa parte de la ciudad, que podemos encontrar, por ejemplo, en la aplicación de libre distribución Google Earth.

Teniendo en cuenta que este modelado no ha de tener demasiado detalle, sino que más bien, su naturaleza ha de ser sencilla y su acabado limpio para que el efecto de

sustracción sea efectivo y rotundo, la cantidad de polígonos con la que lo construimos no es excesiva.



Figura 3. Creación del modelado urbano desde vista aérea

El proceso es sencillo, desde una imagen aérea con suficiente calidad, tomamos las dimensiones bidimensionales y los datos de ubicación relativa en planta de cada volumen, y con técnicas fotogramétricas y/o topográficas [6; 8], le asignamos a cada uno de ellos, su altura y las peculiaridades específicas que el sistema necesita para funcionar; se ha de cuidar en específico el modelado del contorno superior, ya que es generalmente el que limita con la visión del edificio o monumento que aumenta la realidad al introducirse en el sistema.

El proceso se puede realizar en principio con cualquier programa de creación 3d, siempre que el resultado final pueda exportarse al formato nativo de opengl, con extensión *.obj.

En cuanto al edificio que introducimos en el sistema, hemos de recurrir en principio a un nivel de acabado mucho mayor y a una texturización del mismo con fotografías reales del edificio para conseguir un efecto mayor de realidad. Este modelado generalmente es mucho más detallado, y contempla por tanto una cantidad mayor de polígonos pues ha de soportar con garantías la visualización del usuario-espectador y la comparación que se realiza con la realidad al ubicarlo en el mismo plano de visión en el que están los edificios reales. A partir de la información que se pueda reunir sobre el edificio, se realiza un modelado exhaustivo, en el que son muy importantes las proporciones y su altura.

Por otro lado, su textura no puede ser de origen sintético, pues se evidencia inmediatamente y se percibe como una construcción digital, la solución más efectiva es cubrir el modelado entero con fotografías del edificio original, consiguiendo así un efecto mucho más creíble, las fuentes de obtención de éstas, pueden ir desde la toma directa si es posible, hasta la utilización de imágenes tomadas de Internet, ya que la resolución final de visualización es la de pantalla, 72 ppp.

Es preciso recalcar que todos los programas de creación 3d, poseen un apartado de texturización, comúnmente llamado mapeado, que introduce sobre la imagen que vayamos a utilizar, los efectos necesarios para



integrarla en la imagen final; al exportar el objeto al formato *.obj, estos efectos programados se pierden, ya que generalmente, se exporta simplemente la información volumétrica y el enlace a la imagen original que ha de servir de textura. Por ello la integración y adecuación de estas imágenes se ha de realizar en un programa externo de edición de las mismas, como el Adobe Photoshop o el Corel Photopaint, de esta forma, el modelo *.obj, se encarga de ubicar estas imágenes, ya preparadas, en el modelo tridimensional.

Una vez obtenidos los dos modelados, se ubican en el espacio, el primero, correspondiente al entorno real, y del cual se utiliza únicamente su geometría, se sitúa con respecto a la ubicación del usuario-espectador, exactamente en las coordenadas relativas que coinciden con las de los volúmenes reales, buscando así una superposición de formas lo más exacta posible.

El segundo se puede ubicar en donde queramos, pero siendo conscientes, como decíamos anteriormente, de que lo interesante es que se contemple todo su volumen dentro del mismo campo visual que nos posibilita ver los edificios reales cercanos. De lo contrario, no existiría esa comparación que hace al sistema novedoso y útil.

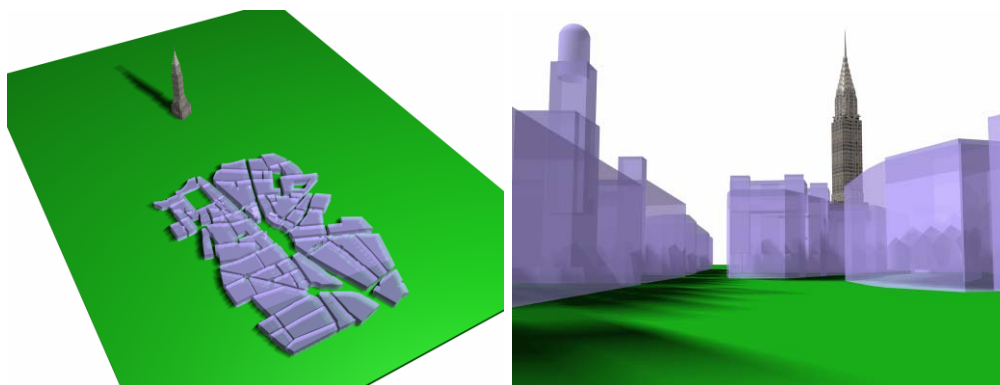


Figura 3. Modelos 3D de los edificios pertenecientes al entorno real y modelo del edificio Chrysler.

3.4. Orientación y posicionamiento del usuario

Para una correcta ubicación del usuario dentro de una aplicación de realidad aumentada, es necesario calcular un total de seis parámetros, generalmente denominados como “parámetros de orientación externa”, en contraposición a los “parámetros de orientación interna” (distancia focal y punto principal de la cámara). Estos seis parámetros son: tres rotaciones u “orientaciones”, que son el acimut, cabeceo, y balanceo (α , ν , κ); y tres coordenadas o “posiciones”, que son la (X, Y, Z) del usuario dentro del sistema 3D establecido. Como ya se ha apuntado en el apartado 3.1., la orientación y posición del usuario dentro de nuestra aplicación se realiza mediante la combinación de dos sensores, uno visual y otro inercial (comúnmente denominado IMU: Inertial Motion Unit). Ambos sensores son de características diferentes, pero si se combinan adecuadamente, pueden ofrecernos la precisión necesaria para nuestra aplicación. Dichas características o particularidades son:

- El sensor inercial es un tipo de sensor “directo”, es decir, registra automáticamente y de forma continua una serie de parámetros sin necesidad de realizar cálculos. Los parámetros que registra el MT9 son: la aceleración, variación del giro y campo magnético local en tres direcciones ortogonales, y la

- temperatura ambiente. A partir de estos datos, directamente se obtiene la orientación del usuario, que puede venir expresada como una matriz de rotación.
- Por el contrario, la cámara es un sensor de tipo “indirecto”, es decir, se necesitan de una serie de operaciones matemáticas para poder calcular los parámetros que necesitamos a partir de un previo análisis de la imagen recogida. Específicamente, la cámara deberá de reconocer una serie de puntos repartidos por el espacio; estos puntos se denominan “puntos de apoyo” y tienen la peculiaridad que sus coordenadas tridimensionales son conocidas. Estos puntos pueden hacerse visibles mediante pegatinas o elementos reflectantes, de tal modo que la cámara los pueda registrar y reconocer mediante técnicas de visión por computador (en nuestro caso, tracking de color). A partir de ellos, y mediante una serie de operaciones matemáticas, obtendremos finalmente la posición del usuario.

En la siguiente figura se puede ver esquematizado el funcionamiento de nuestro sistema, tal y como hemos explicado:

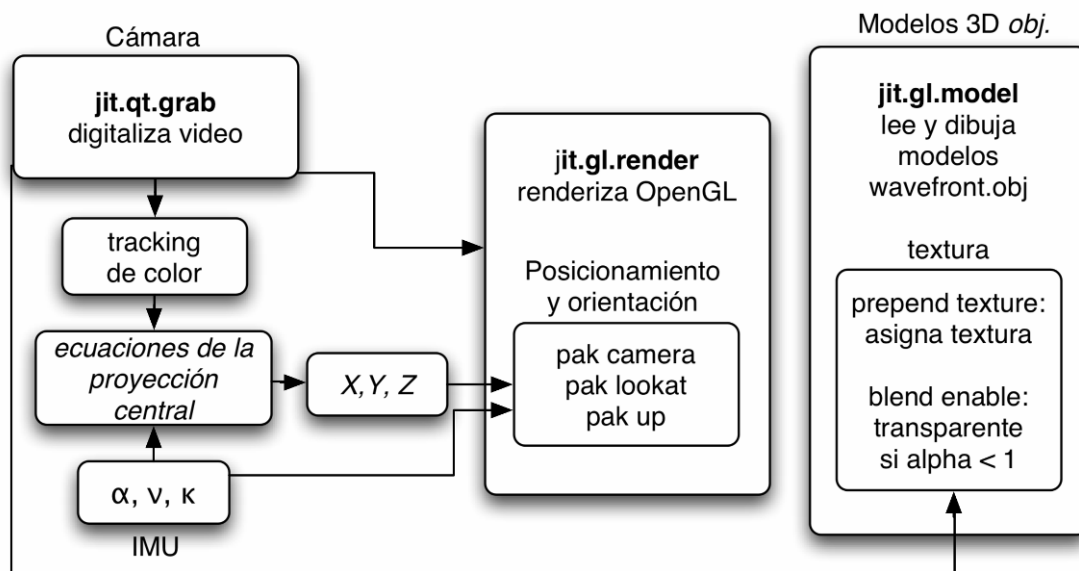


Figura 4. Esquema del funcionamiento de nuestro sistema dentro del programa Max/MSP Jitter.

Realmente, tanto la posición como orientación del usuario podría realizarse con el único empleo de un tracking visual [1; 5; 9; 10; 12]; el problema que surge en este caso, es que el número mínimo de puntos de apoyo que debería registrar la cámara simultáneamente son seis [8], hecho que ralentizaría considerablemente el proceso de adquisición y procesado de los datos y, por lo tanto, dificultaría considerablemente la ejecución de la aplicación en tiempo real. En Urbanmix, al encargarse el IMU de adquirir de forma directa la matriz de rotación, la cámara únicamente deberá registrar simultáneamente un mínimo de dos puntos, de acuerdo con las ecuaciones de la proyección central [6; 8; 13], para calcular la posición.

Respecto al tracking o rastreo de color, hemos utilizado el objeto *tap.jit.colortrack*, que detecta colores específicos de una imagen-vídeo y devuelve información numérica de ellos. De esta manera obtendremos en cada instante información que proporciona las coordenadas de los centros de los objetos rastreados que, como ya hemos dicho, son

una especie de pegatinas circulares de colores diferentes y que contrasten con el entorno.



Figura 5. Imagen de la Puerta del Sol con las pegatinas;
Objetos registrados por el tracking de color.

4. CONCLUSIONES

En esta comunicación se ha presentado Urbanmix, un sistema de realidad aumentada de creación propia.

En este, el usuario se ve inmerso, mediante un dispositivo HMD, en un entorno aumentado; en particular, el espacio donde se desarrolla la aplicación puede ser cualquier ciudad o entorno susceptible de ser modelado; como ejemplo, hemos tratado de recrear nuestra aplicación en la plaza de la Puerta del Sol de Madrid.

El usuario percibe su entorno real mezclado con una serie de objetos virtuales creados por ordenador; estos objetos son edificios de otra ciudad diferente, siguiendo nuestro ejemplo, el edificio Chrysler de Nueva York, y que están dotados de textura foto-realística para aumentar la sensación de credibilidad del sistema en su conjunto. De este modo, el usuario percibe una fusión de ciudades (Madrid-NY), alterando su percepción del entorno y viéndose inmerso en una nueva realidad.

Este proyecto se ha realizado con el programa Max/MSP Jitter que, en principio, está pensado para gestionar en tiempo real imágenes-video y objetos 3D, pero que nosotros hemos adaptado para lograr una combinación y/o mezcla de la realidad con la ficción.

Para ello, hemos creado un *patch* que resuelve el problema de la oclusión mediante el empleo de modelos 3D de los objetos pertenecientes al entorno real (Puerta del Sol de Madrid) y máscaras de vídeo; además, la posición y orientación del usuario dentro de Urbanmix, se ha realizado también mediante un *patch* que combina un tracking visual con uno inercial, todo ello en tiempo real.

5. REFERENCIAS

- [1] ARToolKit, <http://www.hitl.washington.edu/artoolkit>
- [2] BERNSTEIN, J., CLAYTON, J. K., DOBRIAN, CH., DUBOIS, R. L., JONES, R., TAYLOR, G. (2002-2004) *Jitter Tutorials and Topics*. Cycling '74.
- [3] FISCHER, J., BARTZ, D., STRAßER, W. (2004). Occlusion Handling for Medical Augmented Reality using a Volumetric Phantom Model. *Proceedings of the ACM Symposium on Virtual Reality Software and Technology (VRST'04)* (Hong Kong, 10-12 Noviembre).

- [4] FISCHER, J., REGENBRECHT, H., BARATOFF, G. (2003). Detecting Dynamic Occlusion in front of Static Backgrounds for AR Scenes. *Proceedings of the workshop on Virtual environments (EGVE '03)* (Zurich, Switzerland, 22-23 Mayo, 2003). ACM Press, New York, NY, pp. 153-161.
- [5] KOLLER, D., KLINKER, G., ROSE, E., BREEN, D., WHITAKER, R., AND TUCERYAN, M. (1997). Real-time vision-based camera tracking for augmented reality applications. *Proceedings of the ACM Symposium on Virtual Reality Software and Technology (VRST'97)* (Lausanne, Switzerland, 15-17 Septiembre, 1997). ACM Press, New York, NY, pp. 87-94
- [6] KRAUS, K. (1997). *Photogrammetry Volume 2, Advanced Methods and Applications*. Ferdinand Duenmiller (4th edition). Vienna.
- [7] LEE, G. A., BILLINGHURST, M., KIM, G. J. (2004). Occlusion based Interaction Methods for Tangible Augmented Reality Environments. *Proceedings of the 2004 ACM SIGGRAPH international Conference on Virtual Reality continuum and its applications in industry (VRCAI'04)* (Singapore, 16-18 Junio, 2004). ACM Press, New York, NY, pp. 419-426.
- [8] LERMA, J. L. (2002). *Fotogrametría Moderna: Analítica y Digital*. Universidad Politécnica de Valencia.
- [9] NEUMANN, U., YOU, S. (1999). Natural Feature Tracking for Augmented Reality. *IEEE Transactions on Multimedia*. Vol. 1, no. 1, Marzo, pp. 53-64.
- [10] PIEKARSKI, W., THOMAS, B. H. (2002). Using ARToolKit for 3D Hand Position Tracking in Mobile Outdoor Environments. *Proceedings CD of 1st International Augmented Reality Toolkit Workshop*. (Darmstadt, Germany, 29 Septiembre, 2002).
- [11] PORTALÉS, C., GINER, F., SANMARTÍN, F. J. (2005). Back to the 70's. *Proceedings of the 2005 ACM SIGCHI International Conference on Advances in Computer Entertainment Technology (ACE'05)*. (Valencia, España, 15-17 Junio, 2005). ACM Press, New York, NY, pp. 209-212.
- [12] PRINCE, S. J. D., XU, K., CHEOK, A. D. (2002). Augmented Reality Camera Tracking with Homographies. In *IEEE Computer Graphics and Applications*. Vol. 22, no. 6, Noviembre/Diciembre, pp. 39-45.
- [13] RÖNNHOLM, P., HYYPPÄ, H., PÖNTINEN, P., HAGGRÉN, H. (2003). Interactive relative orientation between terrestrial images and airborne laser scanning data. In *Proceedings of the ISPRS working group III/3 workshop, Vol. XXXIV-3/W13* (Dresden, Alemania, 8-10 Octubre). Editors: H.-G. Maas, G. Vosselman, A. Streilein.
- [14] Unibrain, <http://www.unibrain.com>
- [15] Xsens, <http://www.xsens.com>