Juan Pedro Bandera Rubio

Completa los estudios de Ingeniería de Telecomunicación en la Universidad de Málaga, realizando las especialidades en Electrónica y Comunicaciones, entre 1997 y 2003. En 2003 presenta el PFC "Control Automático de Videoconferencia mediante Realidad Aumentada", calificado con Matrícula de Honor.
Actualmente trabaja como Becario FPU del MECD en el Dpto. de Tecnología Electrónica de la Universidad de Málaga, donde ya disfrutó en 2001/2002 de una Beca-colaboración. Su trabajo ha girado en torno a la robótica autónoma, el procesado de imágenes y la construcción de modelos 3D. Durante este período ha colaborado en dos publicaciones en congresos nacionales (Hispabot 2003 y URSI 2003) y en dos en congresos internacionales (SSPRA 2003–Grecia y MELECON 2004–Croacia).
En los últimos meses se ha encargado de la construcción de un modelo 3D que se utilizará en procesos de captura y análisis de movimientos humanos, como parte de una línea de investigación recientemente iniciada en el Dpto. acerca de modelos de aprendizaje por imitación y refuerzo para robots humanoides.

RESUMEN DEL PROYECTO FIN DE CARRERA

Control Automático de Videoconferencia mediante Realidad Aumentada

1 Introducción

La captura, almacenamiento y transmisión de vídeo digital ha recibido una cada vez mayor atención en las últimas dos décadas. La captura y almacenamiento de vídeo se han concretado en el desarrollo de dispositivos capturadores de imagen que operan directamente en formato digital, así como de soportes con una progresivamente mayor capacidad de almacenamiento como la tecnología DVD. Desafortunadamente, un flujo de vídeo en alta resolución con profundidad de color de 24 o 32 bits va asociado a un enorme volumen de datos, lo que ha dificultado enormemente el problema de la transmisión. Debido a ello, recientemente se han llevado a cabo importantes esfuerzos para generar estándares de compresión que permitan reducir el mencionado volumen de datos con pérdidas de calidad tan pequeñas como sea posible. Con este propósito se creó el *Moving Pictures Expert Group* (MPEG). Los estándares MPEG se basan en, mediante la aplicación de determinados mecanismos de muestreo en el tiempo y el espacio y cuantificación en brillo o color, reducir la redundancia de las secuencias de vídeo para obtener unos factores de compresión bastante altos con unas pérdidas de calidad aceptables. Hasta el momento, los estándares más conocidos de la familia son el MPEG-4 [1], de segunda generación y el MPEG-2 [2] y MPEG-1 [3], de primera generación.

La principal ventaja de los estándares de segunda generación frente a los de primera es que el flujo de vídeo y audio deja de comportarse de forma lineal para descomponerse en objetos, como muestra la Fig. 1. De esta forma se facilitan operaciones de televigilancia o control por visión, pudiéndose detectar y seguir los objetos de interés de una secuencia fácilmente. Por otro lado, la información de la secuencia se hace escalable y puede adaptarse al ancho de banda disponible. Esto hace a estos estándares especialmente interesantes para aplicaciones de baja velocidad de transmisión como transmisión de vídeo por medios inalámbricos o por Internet.

A pesar de las ventajas que supone, es importante notar que el estándar MPEG-4 no especifica cómo se extraen los distintos objetos de una escena [1]. Esta es una tarea que actualmente continúa siendo objeto de una intensa labor de investigación, ya que en la mayoría de los casos la extracción de objetos de un entorno real es un problema extremadamente complejo.

En este proyecto se desarrolla un sistema simple para separar una escena real en objetos en tiempo de ejecución. Para ello, se usarán técnicas de segmentación rápida de imágenes y esquemas de selección de áreas de interés basados en sustracción de fondo, por su solidez y velocidad. Por otro lado, y para permitir movimientos de cámara y envío selectivo sólo de dichos objetos de interés, no se usará como fondo una imagen estática. En lugar de ello, se estimará el fondo a partir de un modelo virtual del entorno, que podrá ser enviado previamente al receptor para la composición de la imagen o no, en función del ancho de banda disponible o de las necesidades concretas de cada aplicación, pues en muchas ocasiones precisamente se querrá sustituir el fondo real por otro distinto, tal y como se ve en el esquema de la Fig. 1.

Actualmente, el sistema desarrollado presenta unas velocidades adecuadas, en equipos de calidad media, para su uso en sistemas de videoconferencia comerciales, con anchos de banda limitados. Es por ello por lo que se ha orientado a este tipo de aplicaciones, aunque las características del método lo hacen igualmente útil para su aplicación en redes de televigilancia o telecontrol, o para comunicaciones en banda ancha donde se quiera ampliar la imagen real con elementos virtuales o reducir redundancias. Además, si bien actualmente las pruebas se han limitado a interiores, más fácilmente modelables en un entorno virtual, nada impide la extensión del sistema a localizaciones en exteriores, siempre y cuando dicho exterior sea lo suficientemente sencillo como para poder ser modelado con precisión.

El sistema será explicado con más detalle en los siguientes apartados. Más concretamente, en el apartado 2 se presenta la técnica propuesta para detección de objetos, basada en la sustracción de fondo. Para la estimación del fondo a sustraer, se emplean, como se ha dicho, técnicas de realidad virtual tal como se explica en el apartado 3. De nuevo, mediante técnicas de este tipo se regenera la escena en el extremo receptor, tal como se explica en el apartado 4. El sistema propuesto se ha probado satisfactoriamente con secuencias de vídeo reales, mostrándose algunos resultados en el apartado 5. Por último, las conclusiones y el trabajo futuro se presentan en el apartado 6.

Fig. 1. Composición de imágenes en codificación orientada a objetos

2. Extracción de objetos

En general, la extracción de objetos de una secuencia de vídeo consiste en separar las entidades de interés de dicha escena de un fondo carente de éste. A tal efecto, el fondo debe presentar ciertas características de homogeneidad que permitan distinguirlo de los mencionados objetos. Evidentemente, exceptuando algunos ejemplos clásicos en que el fondo es homogéneo, como extraer al hombre del tiempo de una pantalla azul, es inmediato constatar que en entornos no controlados no existe ningún rasgo obvio (color, distancia, textura o movimiento) que permita hacer esta distinción. Esta dificultad ha devenido en que la mayor parte de los métodos de extracción de objetos en escenas reales se basen en técnicas de sustracción de fondo. Estas técnicas consisten principalmente en capturar lo que la aplicación debe entender como fondo en ausencia de objetos de interés y sustraerla de cada fotograma de la secuencia. Así, sólo quedarán en la imagen resultante los objetos que se diferencian del fondo. En [4] se presenta un estudio de varias técnicas de este tipo.

Incluso un algoritmo tan simple como la sustracción de fondo no resulta obvio de implementar debido a ruido de captura, cambios de iluminación y sombras, que hacen que el fondo difiera del capturado, así como de errores de separación causados por coincidencias temporales entre los objetos y el fondo. La mayoría de las modificaciones a las técnicas de sustracción de fondo convencionales se basan, pues, en trabajar con un modelo de fondo adaptable que generalmente se calcula por promediado de varios fotogramas consecutivos. En estos casos, los móviles se pierden en el promediado mientras que las diferencias por ruido o cambios de luz se atenúan. En [5] los autores propusieron un nuevo método de sustracción de fondo derivado del llamado Olvido Exponencial, pero enmascarando los móviles para evitar que se incluyan en la estimación de fondo. De esta forma, el algoritmo propuesto era capaz de adaptarse no sólo a cambios en las condiciones de iluminación sino también a la presencia de objetos lentos que tienden a fundirse con el fondo. No obstante, todas las técnicas de este tipo presentan una desventaja común: no son válidas para cuando las cámaras se mueven, ya que en esos casos el fondo cambia de forma brusca y no se dispone de modelo alguno para hacer la sustracción.

Dado que en multitud de aplicaciones puede ser interesante desplazar la cámara para mostrar otras áreas, así como acercar o alejar la vista, en este trabajo se propone un nuevo algoritmo de extracción de fondo que permite trabajar con cámara móvil. Este algoritmo se basa en operar con un modelo virtual del entorno de trabajo, que se construye tal como se indica en el apartado siguiente. Combinando con la cámara un *tracker* como los que incorporan las gafas de realidad virtual (Fig. 2.b), se estima hacia donde está enfocada la cámara (Fig. 2.c) y se renderiza la vista correspondiente (Fig. 2.d). Con esta información, se puede extraer del modelo virtual qué fondo se espera ver en cada instante. Por fin, se sustrae este fondo del fotograma actual (Fig. 2.a), separando así de forma simple todo lo que no se encuentre en el modelo virtual del fondo. El algoritmo propuesto es válido para aplicaciones de videoconferencia por dos motivos: i) la videoconferencia tiene lugar en un entorno determinado que no cambia en el transcurso de ésta; y ii) la tasa de imágenes por segundo en videoconferencia es significativamente menor que en otras aplicaciones, lo que permite su procesado en tiempo real.

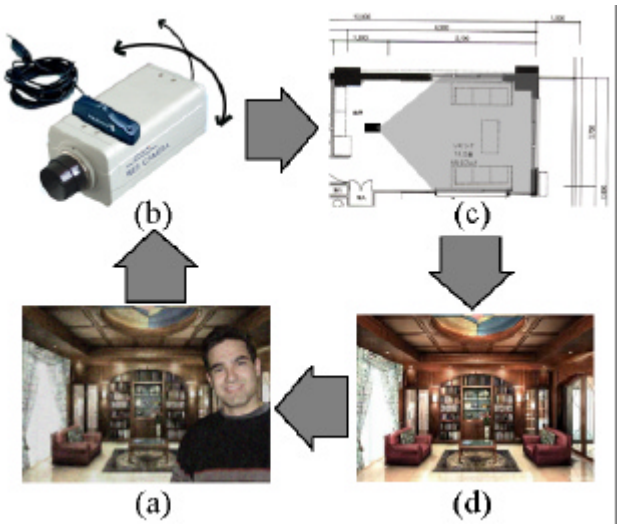

Fig. 2. Esquema de la aplicación: a) entorno de captura; b) cámara con *tracker* para dar información de posición; c) vista estimada sobre el modelo virtual; d) modelo del fondo estimado.

El algoritmo de sustracción de fondo propuesto parte, por tanto, de dos imágenes en color, una real y otra virtual, de las mismas dimensiones (Fig. 3). Dado que la rapidez es básica para poder trabajar en tiempo real, se recomienda almacenar dichas imágenes como arrays en lugar de como matrices. Los pasos de la comparación son los siguientes:

1º) Se diezma la imagen a una fracción de su resolución para filtrar el ruido de captura y disminuir la sensibilidad a pequeñas diferencias con respecto al modelo virtual. A partir de ahora, cuando se hable de 'imagen', nos referimos a 'imagen diezmada'.

2º) Se calcula la diferencia de color píxel a píxel entre la imagen real y la virtual. Para ello:

1. Si los colores están suficientemente saturados, se calculan las componentes r,g normalizadas (r = R / (R+G+B) y g = G / (R+G+B)) y la distancia de color d_i entre los píxeles i de imagen y fondo se calcula utilizando la siguiente fórmula:

$$d_i = |r_{imagen}(i) - r_{fondo}(i)| + |g_{imagen}(i) - g_{fondo}(i)|.$$

   Si esa distancia supera un cierto umbral, entonces los dos píxeles se consideran distintos. En este caso se ha escogido el espacio r,g normalizado por su resistencia a cambios en la iluminación y por su simplicidad, que lo hace especialmente indicado para la implementación de algoritmos que se han de ejecutar en tiempo real, como el desarrollado en este proyecto.

2. Si, en cambio, los colores están poco saturados, se calculan tres distancias de color, una para cada componente RGB. Estas distancias son la diferencia de los valores de esa componente en objeto y fondo. Si cualquiera de estas distancias supera el umbral, los píxeles se consideran distintos.

El nivel de saturación de un color se calcula cambiando la imagen al espacio HSI (tono-saturación-intensidad) y evaluando su componente S [7]. Si tanto la imagen virtual como la real presentan una saturación por debajo de un cierto umbral, se usa el criterio 2. Si no, se usa el 1.

(a) (b)

Fig. 3. a) imagen real; y b) fondo estimado

La Fig. 4 muestra los resultados de la sustracción de fondo usando distintos campos de color sobre la Fig. 3. Puede observarse en dicha figura que hay un cambio de iluminación de la imagen capturada (Fig. 3.a) respecto al fondo estimado (Fig. 3.b). Eso y pequeños errores de orientación en la cámara provoca que la equivalencia no sea perfecta. Es importante contar con estos dos factores a priori, ya que el *tracker* nunca será completamente preciso y la compensación automática de ganancia que incorporan la mayoría de las cámaras provoca que, incluso en condiciones de iluminación constantes, la posición del usuario ante la cámara provoque alteraciones importantes del color. La Fig. 4.a muestra los resultados de la sustracción en el campo RGB. Como puede observarse, este campo es particularmente problemático para el color carne, lo que lo hace inviable para nuestra aplicación de videoconferencia. Es necesario resaltar que esto se debe a que el color RGB es muy sensible a variaciones de intensidad y gradientes, que afectan en particular al color carne. La Fig. 4.b muestra los resultados de la sustracción en el campo HSI. En este caso, el color carne se define correctamente, pero las diferencias en los tonos claros se marcan en exceso y, por tanto, esas áreas no se sustraen correctamente (ver esquina superior derecha). Además,

determinados colores no se distinguen correctamente, tal como se observa en la zona de la camisa a la derecha de la escena. La Fig. 4.c muestra la sustracción con el criterio propuesto. Es interesante notar que no existe ningún tono en particular que resulte problemático. Eso sí, los resultados son considerablemente más ruidosos que en los casos anteriores. Este factor no resulta preocupante ya que en pasos posteriores del algoritmo se elimina el ruido y, mediante un algoritmo de agrupación, se cierran los posibles huecos en los objetos a enviar. Es importante notar que, a pesar de dichos pasos, pueden quedar algunos huecos en los bordes del objeto pero, dado que esto sólo ocurre cuando su color es muy similar al del fondo, no se percibirá significativamente en recepción.

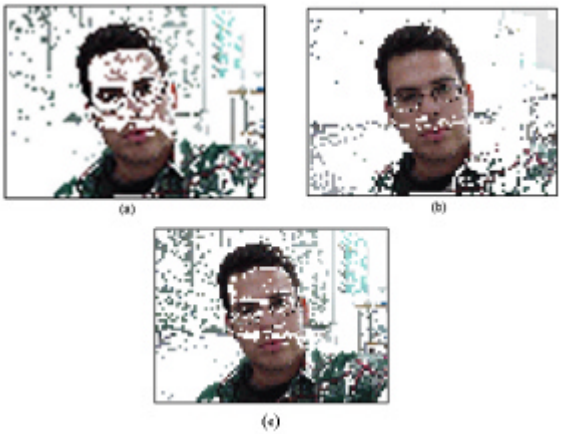

Fig. 4. Sustracción en los campos: a) RGB; b) HSI; y c) propuesto.

3º) A partir del resultado de la comparación anterior, se procede a la agrupación en clases de los píxeles marcados como distintos. Para esto se utiliza un algoritmo de agrupación por mezclado [8] que incluye en una misma clase a todos los píxeles de la imagen diferencia entre los que exista conectividad.

4º) Para cada una de las clases resultantes del paso anterior, se calcula el número de píxeles que la constituyen. A partir de este número, se eliminan todas aquellas clases que no superan un cierto umbral. En aplicaciones de videoconferencia, lo normal es que el interlocutor este a una distancia media o cercana de la cámara, por lo que el área de imagen que ocupa debe ser elevada. Este paso permite eliminar ruido remanente debido a cambios de iluminación y sombras, así como a defectos en el modelo virtual del fondo. Las clases que resultan de este paso de descarte equivalen a los objetos de interés que se van a transmitir, mientras que las zonas descartadas no van a enviarse.

5º) Las clases equivalentes a objetos de interés son sometidas a un proceso de dilatación, que elimina los pequeños huecos que puedan haberse producido por errores en la sustracción de fondo. Como la complejidad en el proceso de dilatación depende de la cantidad de píxeles que se dilaten las clases, se ha mantenido en este proyecto una dilatación de un solo píxel en las imágenes, que puede verse en la Fig. 5. De esta forma, se elimina la mayor parte del ruido de sustracción y se proporciona definición a los objetos detectados, sin provocar una disminución significativa en la cantidad de fotogramas por segundo procesados.

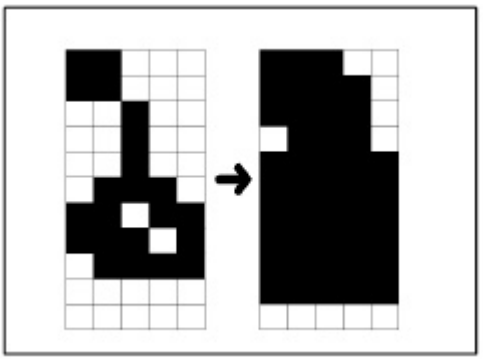

Fig. 5. Proceso de dilatación

Los distintos pasos del proceso pueden observarse en la Fig. 6. La Fig. 6.a muestra una imagen diferencia como la que se obtiene en el paso 2 del algoritmo propuesto. Puede observarse que además de la figura principal, aparecen áreas irregulares debidas a ruido que resultan en cuatro objetos potenciales (Fig. 6.b). Si se descartan los objetos cuya área no supera el umbral prefijado (Fig. 6.c), nos queda una única clase. La Fig. 6.d muestra el objeto final a transmitir, marcándose en un color homogéneo el resto de la imagen que, en recepción, se extraerá de un modelo virtual local tal como se comenta en el apartado 4.

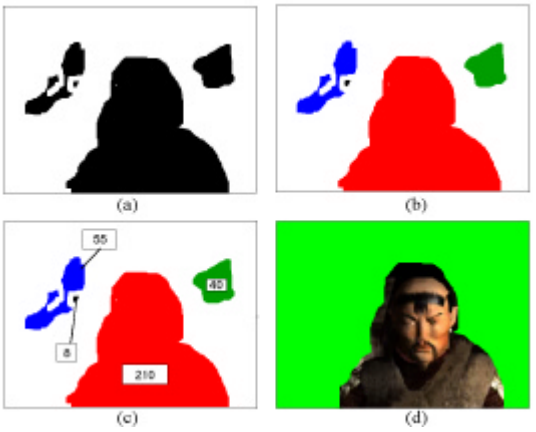

Fig. 6. Sustracción de fondo: a) resultado de la sustracción; b) segmentación de regiones; c) descarte de regiones de áreas inferiores a 80 píxeles; d) fotograma enmascarado.

3. Creación de un fondo virtual

La creación de un modelo de fondo virtual puede hacerse de forma completamente automática o parcialmente supervisada por el usuario. En cualquier caso, para construir dicho modelo es necesario hacer un barrido con la cámara con que se trabaja de la habitación donde se va a efectuar la videoconferencia. Si se conocen los parámetros de dicha cámara con precisión (apertura, deformación, frustrum, resolución, etc), puede extraerse directamente información de profundidad de la secuencia de imágenes resultante. Sin embargo, resulta bastante habitual, sobre todo en cámaras de bajo coste, que los parámetros que suministra el fabricante sean aproximados o insuficientes. Ello dificulta considerablemente la construcción del modelo. Por el contrario, se puede suministrar a priori al sistema un archivo de texto con las medidas aproximadas de la habitación, lo que simplifica y acelera la construcción del modelo. Actualmente, usamos esta segunda opción por simplicidad. El mencionado archivo tiene una estructura muy simple. En una primera línea se indica el número de paredes visibles en el modelo. En la segunda se incluye la altura aproximada del techo. Por último, se incluye una línea por pared a modelar indicando las coordenadas de sus extremos (x1,z1,x2,z2).

Dichas coordenadas son relativas a la cámara, que ocupa la posición (0,0) y apunta a –z cuando el *tracker* está recién inicializado.

Así, a la hora de construir un modelo de este tipo, son necesarias cuatro cosas: i) una cámara de vídeo; ii) un dispositivo de posicionamiento (*tracker*) que permita determinar hacia dónde está enfocada la cámara; iii) fichero con la información de la planta que se va a modelar; y iv) un motor gráfico que permita renderizar la información 3D en un mapa de bits. En este caso hemos escogido el motor Genesis3D [9] por ser abierto, gratuito, flexible y eficaz para una aplicación simple como la propuesta. Los pasos para construir el modelo virtual del entorno son los siguientes:

1º) Se carga la geometría del entorno, a partir del archivo de texto, en el motor gráfico. Esto supone alzar un mapa 3D de las paredes del entorno sin texturas.

2º) Apuntar la cámara al suelo y al techo para capturar sus texturas. A partir de este punto, se procede a girar la cámara para capturar las distintas texturas del entorno, repitiéndose los pasos siguientes para cada captura.

3º) Se calculan mediante trigonometría las aristas que teóricamente deben verse en pantalla (Fig. 7). Para ello, se utiliza la información del mapa de planta y los parámetros de la cámara, y relaciones trigonométricas simples que permiten realizar los cálculos rápidamente. Una vez obtenidas, las columnas de la imagen donde se proyectan las aristas (Fig. 7.b) son usadas en la imagen real para delimitar las texturas que se habrán de mapear en cada pared.

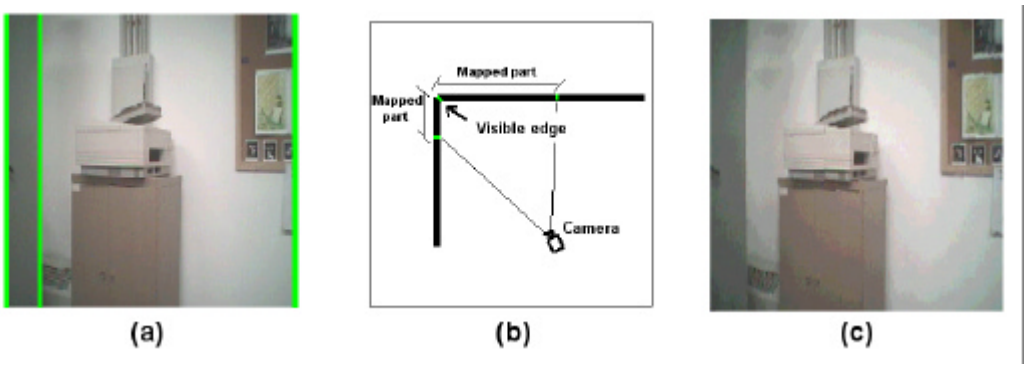

Fig. 7. a) Proyecciones de las aristas virtuales sobre la imagen real; b) Mapa de planta de a) mostrando la cámara, las paredes parcialmente mapeadas y las aristas visibles; y c) Modelo virtual obtenido.

4º) Utilizando el mapa de planta, las distancias de cada arista visible a la cámara pueden obtenerse fácilmente. Hecho esto, se almacena la información en una lista con la estructura de la Fig. 8, donde 'numpad' es el número de paredes del entorno. Las casillas a –1 indican que la pared correspondiente no tiene arista en esa columna de la pantalla, mientras valores distintos a –1 indican que esa pared tiene arista en esa columna y, además, que la distancia de la arista a la cámara (en cm) es el valor contenido en esa casilla. En la Fig. 8, por ejemplo, las paredes 1 y (numpad-1) tendrán teóricamente arista en la columna 1 de la imagen y, además, dicha arista estará situada a 234 cm de la cámara. Es importante notar que cuando una pared tenga arista, usualmente habrá al menos otra pared que tenga arista en esa misma columna para formar una esquina.

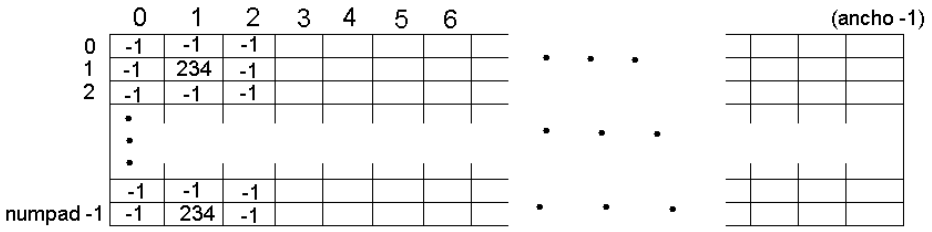

Fig. 8. Estructura de aristas del modelo virtual

5º) A partir de la imagen captada por la cámara y la información de aristas y distancias obtenida en los pasos anteriores, se recortan los fragmentos de imagen que deben pegarse en las diferentes caras. Estos fragmentos son procesados para corregir la deformación debida a la perspectiva. Para ello, se utiliza la técnica de *Affine Texture Mapping* [10] por su rapidez.

6º) Finalmente se agregan los fragmentos obtenidos en el paso 5 sobre la pared correspondiente del modelo. Hay que tener en cuenta que habrá caras que no se verán completamente. En estos casos, se mapea sólo la parte visible.

Durante el proceso, se van mostrando al usuario los resultados para que continúe capturando vistas o concluya el proceso cuando esté satisfecho con el resultado, tal y como se muestra en la Fig. 9.

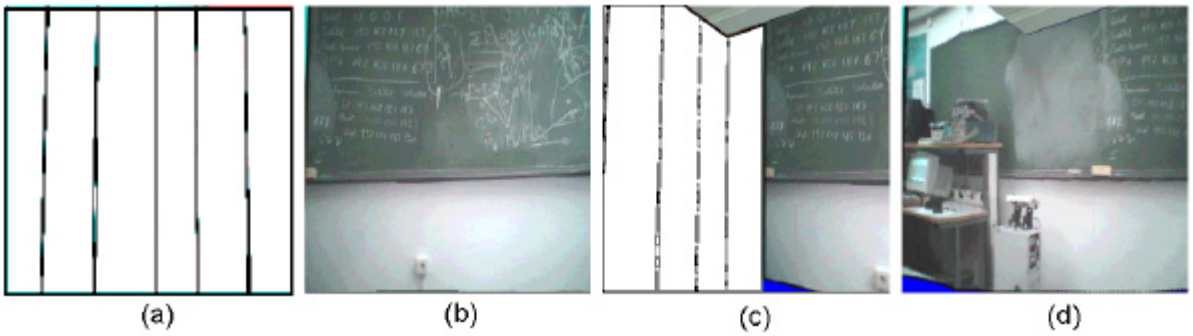

Fig. 9. Mapeado de texturas: a) Vista del modelo virtual sin texturas; b) Texturas reales aplicadas a a); c) Vista del modelo parcialmente mapeado; y d) Vista del modelo completamente mapeado.

4. Composición en recepción

La representación de la imagen completa implica una mezcla del fondo virtual con las siluetas reales obtenidas al aplicar el algoritmo de comparación. En función de las características de la comunicación, esta representación puede tener diferentes características. En el mejor de los casos, el transmisor envía el modelo de su entorno al receptor para que la imagen en recepción sea tan fiel como sea posible. Sin embargo, puede darse el caso de que por cuestiones de tiempo, ancho de banda o incompatibilidad de software en ambos extremos, esto no sea posible. En estos casos, puede recurrirse a utilizar un bitmap cualquiera como fondo fijo en recepción o a hacer uso de cualquier otro fondo virtual ya disponible en dicho extremo. Es importante notar que, de trabajar con un fondo virtual, es necesario acompañar los fotogramas enviados con la información del *tracker* para cada uno de ellos. Así, el

receptor podrá alinear su modelo virtual con la posición de la cámara del transmisor. La etapa de composición consiste en lo siguiente:

1º) Se construye un bitmap igual a la imagen enmascarada, sólo que aquellos píxeles enmascarados, de color homogéneo, que pertenecen al fondo, se convierten en píxeles transparentes. Este bitmap se denomina HUD y se genera con un tipo de representación característico del Genesis3D [9], donde cada píxel incorpora un campo alfa (transparencia) configurable independientemente.

2º) El HUD se superpone a la vista correspondiente del modelo virtual, previamente renderizada por Genesis3D. La información de orientación recibida permite la correspondencia entre el fondo real en transmisión y el modelo virtual de fondo utilizado en recepción.

Así se obtiene el resultado definitivo, como puede verse en la Fig. 10.

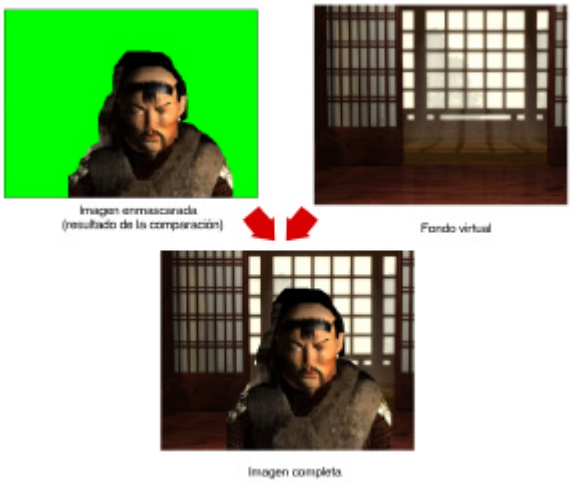

Fig. 10. Resultado del proceso de composición

5. Experimentos

En esta sección se presentan algunos experimentos representativos del trabajo realizado utilizando cámaras de videoconferencia de gama baja y un *tracker* en la configuración de la Fig. 2.a. Las cámaras, al ser de baja calidad, proporcionan imágenes muy ruidosas y distorsionadas, con un marcado efecto 'ojo de pez' que redondea los bordes de las imágenes y curva las aristas horizontales según su posición. Además, el *tracker* presenta también inestabilidades y variaciones debidas fundamentalmente a la presencia de campos electromagnéticos de intensidad variable en el entorno de prueba. Aún en estas condiciones, que podríamos denominar de 'caso peor', el algoritmo propuesto ofrece resultados aceptables, tal y como se muestra a continuación.

La Fig. 11 muestra un primer ejemplo en que la cámara se mantiene fija durante un tiempo después de haberla desplazado tras la inicialización del *tracker*. Este desplazamiento se manifiesta en que el fondo virtual y el real no están perfectamente alineados, tal y como se puede observar en la posición del enchufe en la mitad inferior de la Fig. 11.a y posteriores. Este error se ve intensificado por las imprecisiones en el modelo de paredes y por errores en la apertura de la cámara, ya que se utilizan dos tipos de éstas para comprobar la resistencia del sistema a variaciones hardware. Las Fig. 11.b y 11.c muestran dos fotogramas con y sin usuario respectivamente, así como los objetos extraídos del fondo virtual. Puede observarse que, a pesar de las distorsiones y fallos de alineamiento, el usuario (Fig. 11.b) se extrae casi

correctamente de la escena salvo por dos detalles. El primero, la mitad derecha del borde inferior de la pizarra, que se envía al extremo receptor porque debido al ojo de pez resalta claramente del fondo. El segundo es la región excesivamente iluminada en la mejilla derecha del usuario, que se convierte prácticamente en blanca y, por tanto, es idéntica al fondo. Los problemas de segmentación debidos a que fondo y objeto sean iguales no tienen solución sin usar técnicas cualitativas de alto nivel, pero es importante notar que pasan casi desapercibidos en recepción si se está usando el mismo modelo de fondo que en transmisión, ya que los huecos se rellenan del color al que se parecen. Cuando el usuario se retira de la escena (Fig. 11.c), el borde de la pizarra se deja de transmitir pues aunque la deformación sigue existiendo, el área ocupada por la clase errónea es demasiado pequeña y se descarta como objeto de interés. Sin embargo, se envía una pequeña porción de la silla ya que, como se aprecia al comparar las Fig. 11.a y Fig. 11.c, errores de alineamiento hacen que el área visible de ésta sea ligeramente mayor en la imagen capturada que en la prevista.

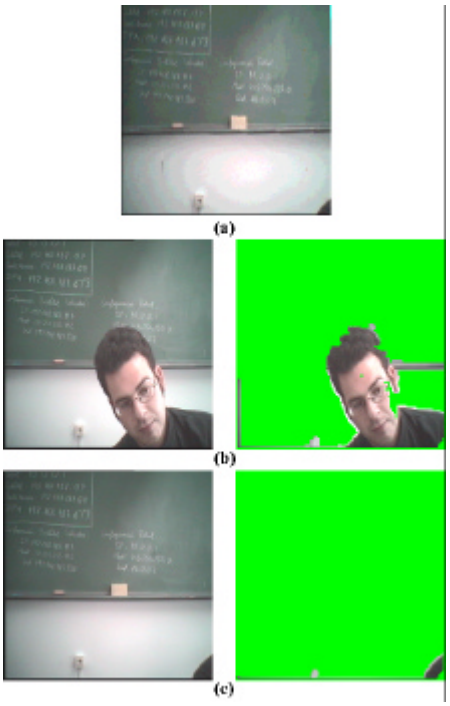

Fig. 11: Camara fija: a) fondo virtual; b) fotogra ma con usuario; c) fotograma sin usuario.

La Fig. 12 muestra un nuevo experimento, esta vez manteniendo el fondo fijo pero girando la cámara significativamente hacia la derecha, tal como se puede apreciar en la diferencia de posición de la silla en la Fig. 12.a. Es importante notar que la silla no se encuentra en el fondo virtual, sino que se incorpora luego, y que debido a la simplicidad del modelo hay un error de mapeo por pegado de texturas en el segundo fondo estimado (Fig. 12.b). Estos errores no aparecen siempre y podrían eliminarse con un modelo de fondo más depurado, pero el objetivo de estas pruebas es, como se ha comentado, evaluar el caso peor. La Fig. 12.c muestra los resultados de la extracción de fondo para los fotogramas de la Fig. 12.a. En el primer caso, la silla se extrae correctamente. En el segundo fotograma, la silla también se extrae sin problemas. No obstante, en la zona donde había un error de mapeado de texturas se extrae un objeto inexistente de área significativa. Es importante remarcar que este objeto no se debe tanto al error de mapeado como a la compensación automática de ganancia de la cámara de vídeo que se utiliza. Esta compensación altera la iluminación de la imagen en función de lo que haya en escena en un momento dado. Naturalmente, cambios bruscos de iluminación provocan diferencias importantes de color con respecto al fondo virtual, creado cuando no había objetos de interés en la imagen. Así, en la textura que se capturó para la región derecha del fondo virtual aparecían sombras en la arista de corte que no aparecen en el nuevo fotograma capturado y que provocan la aparición de esa región extra. Este tipo de

regiones se podrían eliminar por actualización de fondo virtual con una técnica parecida a la que se propuso en [5] o por criterios cualitativos de forma, ya que en regiones erróneas esta es muy errática.

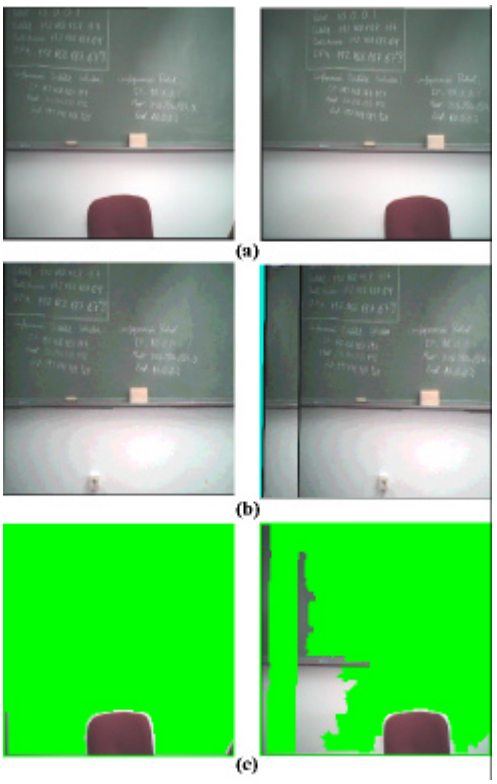

Fig. 12: Cámara móvil; a) fotogramas; b) fondos virtuales correspondientes; c) objetos extraídos.

La Fig. 13 presenta un tercer experimento en que la cámara se gira hasta alcanzar una parte del fondo que no se había mapeado en el modelo virtual, que puede observarse a la derecha de la Fig. 13.a. Cuando se captura un fotograma donde parte del fondo corresponde a regiones no mapeadas (Fig. 13.b), la zona correspondiente a esas regiones se incluye en los objetos a enviar (Fig. 13.c). De nuevo, en este ejemplo se aprecia el efecto de la compensación automática de ganancia en la esquina inferior izquierda de la escena, donde puede observarse como el fondo virtual es mucho más claro que la imagen capturada.

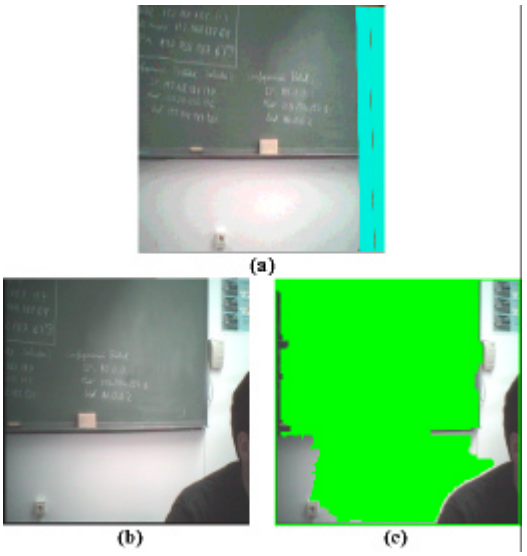

Fig. 13. Mapeado parcial: a) fondo virtual estimado; b) fotograma capturado; c) objetos extraidos.

La Fig. 14 presenta algunos ejemplos de composición de imagen en recepción cuando la cámara está estática (Fig. 14.a y b) y cuando se desplaza significativamente (Fig. 14.c y d). La Fig. 14.b corresponde al usuario extraído en la Fig. 11.b, donde se comentaba que la mejilla derecha no se había segmentado correctamente por ser parecida al fondo. Aquí se observa como, en estos casos, en recepción el efecto no puede apreciarse precisamente porque esa diferencia con el fondo no era significativa. En la Fig. 14.c, por otro lado, el punto de vista de la cámara se ha desplazado hacia abajo. La composición es adecuada en tanto que dicha zona estaba modelada de forma correcta en el fondo virtual disponible. Sin embargo, si giramos la cámara hacia la zona defectuosa por luces y fallos en el modelo de planta que presentamos en la Fig. 12, puede observarse que, si bien corresponde con la escena, la composición presenta zonas con diferencias de color apreciable que corresponden a objetos que no deberían haberse enviado. Tal como se comentó, esto podría corregirse mejorando el modelo de planta, buscando modelos de color más complejos y resistentes a cambios de iluminación o actualizando las texturas en función de las variaciones de luz. Mejorar el modelo de planta le restaría autonomía al sistema, ya que lo complicaría innecesariamente en muchos casos. Modelos de color más complejos implican una carga computacional mucho más elevada y, por tanto, una reducción inaceptable en la tasa de imágenes por segundo. La actualización de fondo es rápida, pero debería enviarse a recepción. Eso sí, a una velocidad mucho más baja que los objetos.

De cualquier forma, hay que resaltar el hecho de que sea cual sea la naturaleza de los errores, el solape de los objetos erróneamente detectados sobre el fondo es, en la mayoría de los casos, apenas perceptible. Así, estos errores sólo implican una disminución de la eficiencia del sistema, pues no se ha eliminado de la escena toda la información redundante. En este sentido, es preferible ser permisivo con el envío de áreas erróneas, prefiriéndose esto a aumentar las restricciones hasta el punto de que se dejen de enviar zonas de interés.

Por último, es importante señalar que si el receptor no dispone por cualquier motivo del hardware o software necesario para manejar un modelo virtual o, simplemente, no ha recibido éste, el sistema sigue siendo válido y los objetos pueden componerse sobre un bitmap fijo o sobre un fondo virtual alternativo disponible en recepción. En transmisión, no obstante, el modelo seguiría siendo necesario para realizar la extracción de objetos.

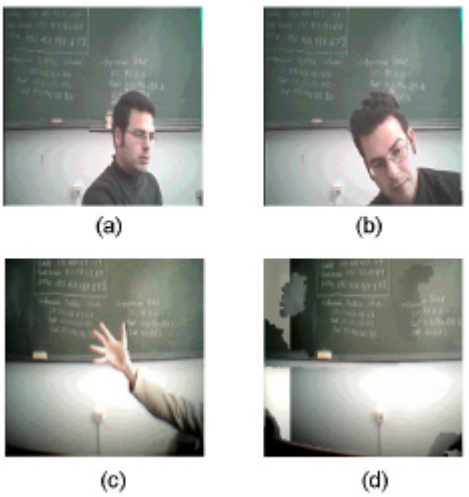

Fig. 14. a-b) composición con cámara fija; c-d) composición con cámara en movimiento.

6. Conclusiones y trabajo futuro

En este proyecto se ha elaborado un sistema de extracción de objetos de un fondo real, para una cámara móvil y entornos controlados, basado en crear un modelo virtual sencillo de dichos entornos. El sistema se ha probado con secuencias reales de vídeo utilizando una cámara de videoconferencia de gama baja y un *tracker* comercial y proporciona una reducción media del 50-60% en el tamaño, al comprimir con MPEG-4, respecto a la compresión de esas mismas secuencias no procesadas (Fig. 15). Las ventajas más relevantes del sistema propuesto son su bajo coste y su versatilidad. Además, el sistema permite procesar el vídeo en tiempo real para la tasa de imágenes típica de una videoconferencia con un PC estándar. Los inconvenientes más importantes detectados son la sensibilidad a la distorsión tipo ojo de pez de las cámaras más económicas y a la compensación automática de ganancia. El trabajo futuro se centrará en tres puntos básicos: i) mejorar las características de la segmentación añadiendo técnicas cualitativas; ii) eliminar en la medida de lo posible la sensibilidad a la compensación automática de ganancia mediante actualización del fondo, extendiendo a cámara móvil los métodos que propusimos en [5]; y iii) incorporar técnicas de reconocimiento automático de objetos que identifiquen en la medida de lo posible los objetos de la escena, como la silla de la Fig. 12, y, en lugar de enviarlos hasta que desaparezcan de ésta, incluya en el modelo de fondo la silla virtual correspondiente. Este nuevo objeto y su ubicación se enviarían a recepción y se combinarían con la secuencia de vídeo recibida. Por último, sería también deseable automatizar completamente el sistema de cálculo de planta de la habitación. Ya se han hecho esfuerzos a este respecto utilizando dispositivos de cálculo de distancias, pero en un futuro lo idóneo sería usar únicamente información de vídeo.

	Tamaño original	MPEG 4	MPEG 4 + Algoritmo propuesto
Secuencia 1	446 MB	6,34 MB	2.6 MB
Secuencia 2	926 MB	12,88 MB	5.9 MB
Secuencia 3	1311 MB	18,35 MB	14.5 MB
Secuencia 4	310 MB	4,4 MB	1,1 MB
Secuencia 5	705 MB	9.9 MB	9.8 MB

Fig. 15. Resultados de la compresión de distintas secuencias reales de vídeo, antes y después de ser procesadas por el algoritmo de sustracción de fondo desarrollado.

Referencias

[1] R. Koenen, *MPEG-4 - Multimedia for our time,* IEEE Spectrum, Vol. 36, No. 2, pp. 26-33, 1999

[2] ISO/IEC 13818-1: *Generic coding of moving pictures and associated audio: Systems.* (MPEG-2 Systems)

[3] *Introduction to MPEG*: http://www.faqs.org/ faqs/compression-faq/part2/section-2.html

[4] S.J. McKenna, S. Jabri, Z. Duric, A. Rosenfeld and H. Wechsler, *Tracking Groups of People*, Computer Vision and Image Understanding 80, 42–56 (2000)

[5] J. A. Rodríguez, C. Urdiales, P. Camacho, F. Sandoval, *Detección Jerárquica de Móviles sobre Geometrías de Fóvea Adaptativa*, Revista Electrónica de Visión por Computador, 3. ISSN:1575-5258 (2002)

[6] Gevers, T. and Groen, F. C. A., "Segmentation of Color Images" , 7th Scandinavian Conference on Image Analysis, 1991.

[7] Lin X. and Chen S., "Color image segmentation using modified HSI system for road following", Proc. IEEE Conf. on Robotics and Automation, Sacramento, California, pp. 1998-2003, 1991.

[8] Pitas, I., *Digital image processing algorithms*, Prentice Hall: New York, 1993.

[9] *Genesis3D Open Source Engine*: http://www.genesis3d.com/

[10] *Texture mapping* : http://www.geocities.com/SiliconValley/Horizon/6933/texture.html

A. AUTOR

Nombre: Juan Pedro Bandera Rubio

Número de asociado: 15030

C. PROYECTO

Título: Control Automático de Videoconferencia mediante Realidad Aumentada

Fecha de lectura: 16 de Junio de 2003

Calificación: Matrícula de Honor

Departamento: Tecnología Electrónica

TUTOR

Nombre: Cristina Urdiales García

D. PUBLICACIONES RELACIONADAS

- J.M. Pérez, J.P. Bandera, C. Urdiales y F. Sandoval, "Agente autónomo de bajo coste para la exploración de conductos teleoperados mediante realidad virtual", I Seminario Nacional Hispabot (HISPABOT'03), Alcalá de Henares (Madrid), abril de 2003.

- C. de Trazegnies, J.P. Bandera, C. Urdiales y F. Sandoval, "A real 3D object recognition algorithm based on virtual training", IASTED International Conference on Signal Processing, Pattern Recognition, and Applications (SPPRA 2003), Rhodes (Grecia), junio-julio 2003.

- J.M. Pérez, J.P. Bandera, A. Bandera y F. Sandoval, "Algoritmo de agrupación de segmentos en mapas métricos basado en fusión de clases en el espacio de Hough", Unión Científica Internacional de Radio, XVIII Simposium Nacional URSI´2003, La Coruña, Septiembre de 2003.

- J.P. Bandera, C. Urdiales y F. Sandoval, "Selective Video Transmission by means of Virtual Reality Based Object Extraction", aceptado en: MELECON 2004, Dubrovnik (Croacia), mayo 2004.

E. OTRA INFORMACIÓN

- Conferencia *"Aplicación práctica, microbot explorador de conductos"*, ofrecida dentro del ciclo *"Conferencias sobre microbótica"* organizado por la Rama de Estudiantes del IEEE de Málaga, Diciembre de 2002.

- Referenciado en la bibliografía de la asignatura "Control Electrónico Digital", curso 2003-2004, coordinada por D. Felipe Espinosa Zapata, del Departamento de Electrónica, e impartida en la Universidad de Alcalá de Henares (Madrid).