Introducción al Modelado y Visualización Basado en Imágenes †

José Ribelles ‡

Departamento de Lenguajes y Sistemas Informáticos
Universitat Jaume I, 12080, Castellón, Spain
ribelles@uji.es

Resumen
El Modelado y Visualización Basado en Imágenes reúne a un conjunto de técnicas que tratan de acelerar los procesos de modelado y visualización de una escena mediante el uso de imágenes. En la etapa de modelado, ciertos algoritmos específicos utilizan las imágenes para obtener las medidas, posición y orientación de los objetos. En el proceso de visualización, las entidades complejas formadas por multitud de elementos que definen al detalle tanto su geometría como su apariencia se sustituyen por imágenes y se reutilizan mientras conserven la apariencia visual. Este trabajo resume algunas de las técnicas más interesantes presentadas en la literatura para el modelado y visualización basado en imágenes. El objetivo es elaborar una introducción a ese conjunto de técnicas que tratan de la visualización, la recuperación de geometría, la transformación de vistas, la construcción de panoramas y mosaicos y, finalmente, su aplicación al modelado y visualización de arquitectura.

1. Introducción

Uno de los principales objetivos de la Informática Gráfica es generar imágenes sintéticas con calidad fotorrealista. Cada vez son más los campos que requieren este tipo de imágenes y en muchas ocasiones ocurre que la escena a visualizar ha de ser una réplica lo más exacta posible de un modelo real. El *Trazado de rayos* y la *Radiosidad* son técnicas complejas que intentan simular el comportamiento de la luz desde que sale de sus fuentes hasta que llega al observador y son capaces de generar imágenes que muestran un gran realismo. Sin embargo, los algoritmos de visualización tienen un coste computacional muy caro y, hoy por hoy, son inviables para aplicaciones donde se requiere una alta tasa de imágenes por segundo como, por ejemplo, realidad virtual o juegos de ordenador. Por tanto, estas aplicaciones se han de conformar con utilizar imágenes poco realistas con una calidad limitada por el hardware gráfico actual. Por ejemplo, en la figura 1 se muestra una imagen generada mediante la técnica de trazado de rayos. Esta imagen de 1600x1200 pixeles consumió aproximadamente 75 minutos de cpu para generarse [14, 16] en una estación de trabajo con un procesador Pentium-II a 450Mhz y 386MB de memoria RAM.

Figura 1: *Imagen generada utilizando la técnica de Trazado de Rayos.*

El modelado y la visualización basada en imágenes

† Este trabajo ha sido parcialmente financiado por el proyecto TIC99-0510-C02-01 (CICYT, Ministerio de Educación y Ciencia).
‡ Mientras realizaba una estancia en el DSIC de la Universidad Politécnica de Valencia.

(IBMR) se presenta como alternativa a estas técnicas. IBMR parte de fotografías del modelo real y aprovecha el realismo que estas proporcionan. Además, consigue que el coste de visualización de la escena no dependa de su complejidad, manteniendo la tasa de imágenes por segundo requerida por la aplicación. Una fotografía captura el mundo real, es muy fácil de hacer y la imagen obtenida tiene calidad fotorrealista por sí misma. Sin embargo, una fotografía muestra una escena estática. ¿Cómo añadir dinamismo a esa escena?

- Para que el observador pueda mirar en diferentes direcciones se pueden tomar fotografías en todas las direcciones y disponerlas en un panorama (ver figura 2) o mosaico (ver figura 3). De esta forma, es posible mostrar al usuario justo la parte de la escena hacia la que dirige su vista.
- Para que el observador pueda moverse por la escena se pueden tomar diferentes fotografías desde diferentes puntos de vista (ver figura 4) y mostrar la más adecuada dependiendo de su posición.

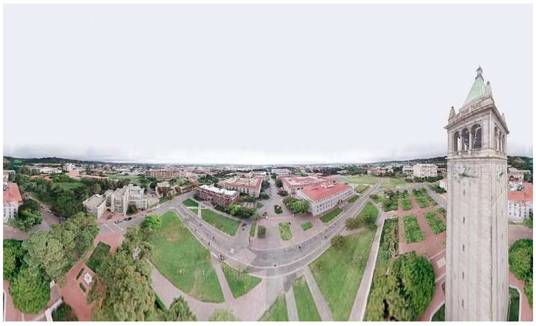

Figura 2: *Panorama de 360 grados* [2].

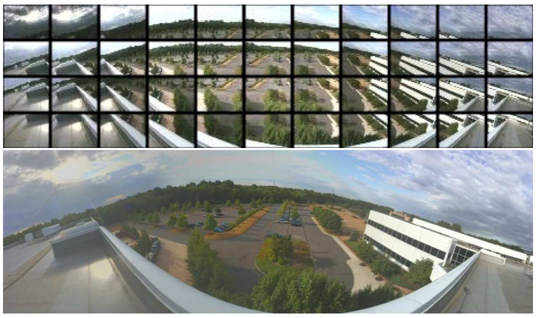

Figura 3: *Ejemplo de mosaico* [30]. *Arriba, conjunto de fotografías; abajo, la composición.*

Es evidente la necesidad de una gran cantidad de imágenes sólo para dotar al observador de cierta movilidad. Para reducir esta cantidad, IBMR requiere de un modelo tridimensional de la escena y utiliza modelo e imágenes para generar nuevas vistas. Mediante las técnicas de modelado tradicional, un usuario ha de realizar un esfuerzo muy alto para crear un modelo tridimensional de la escena, y eso

Figura 4: *Fotografías de un mismo objeto tomadas desde puntos de vista diferentes* [2].

suponiendo que dispone de la información necesaria como podrían ser los planos en el caso de tratarse de un modelo arquitectónico. Por ejemplo, la escena que se muestra en la figura 1 fue modelada utilizando principalmente la técnica de *Geometría Sólido Constructiva (CSG)* y el tiempo de modelado fue de varias semanas. IBMR propone, para disminuir el coste del proceso de modelado, utilizar las propias imágenes como guías de referencia para obtener las medidas, posición y orientación de los objetos. Además, disponer de un modelo 3D permitiría que un usuario pudiese editarlo, es decir, modificar, añadir o eliminar los elementos de la escena. En contra de lo que se pueda pensar, el modelo obtenido no necesita ser tan detallado (ver figura 5) como los modelos tradicionales (en la escena de la figura 1 se utilizaron 6.451 primitivas geométricas y 2.560 operaciones CSG). La fidelidad de esta representación geométrica respecto al mundo real depende de la calidad de las técnicas que se utilicen para obtener la geometría. Este es un problema típico de la Visión por Ordenador y lleva estudiándose desde prácticamente sus orígenes con lo que son muy diversas las soluciones presentadas en la literatura [25].

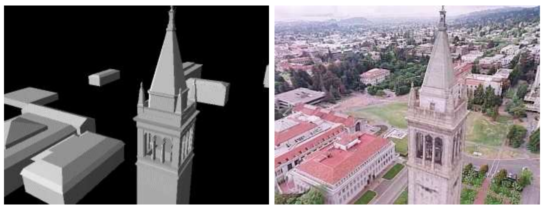

Figura 5: *Ejemplo de geometría obtenida a partir de imágenes* [2]. *A la izquierda una vista del modelo geométrico, y a la derecha la imagen final: geometría + imágenes.*

El objetivo de este informe es dar a conocer las diversas técnicas relacionadas con IBMR y su aplicación al modelado y visualización de arquitectura. En la sección 2 se presentan las técnicas básicas para la visualización basada en imágenes. En la sección 3 se introducen los métodos existentes para la obtención de geometría a partir de imágenes. Más tarde, en la sección 4, se explica la técnica conocida como *Transformación de vistas* que permite generar imágenes correspondientes a vistas intermedias entre dos dadas. A continuación, en la sección 5, se trata la creación de panoramas y mosaicos. Finalmente, en la sección 6 se muestra la aplicación de IBMR a la arquitectura. Quedan fuera de este informe las técnicas denominadas visualización con campos de luz e iluminación basada en imágenes (ver figura 6).

Figura 6: *Imagen generada mediante la técnica de Iluminación basada en Imágenes* [2].

2. Visualización basada en imágenes

Cada vez son más las aplicaciones que requieren de interacción con el usuario. Al mismo tiempo, los avances tecnológicos han hecho posible que estas aplicaciones incorporen modelos con alto grado de detalle lo que dificulta la deseada visualización interactiva debido a que con gran facilidad se sobrepasa la capacidad del hardware gráfico actual. En algunas aplicaciones este problema llega a ser crítico, como ocurre en el campo de la Realidad Virtual, donde es necesario además mantener una alta tasa de imágenes por segundo. Las soluciones presentadas en la literatura a este problema se pueden clasificar en tres grupos principales:

- Eliminación de partes ocultas [10, 31]. Esta técnica intenta evitar que se dibujen objetos no visibles debido a oclusiones producidas por otros objetos de la misma escena.
- Modelado multirresolución [17]. Esta técnica modela los objetos de la escena con diferentes niveles de detalle y determina con qué detalle se ha de visualizar cada objeto en base a un conjunto de consideraciones como, por ejemplo, su contribución visual en la imagen final.
- Visualización basada en imágenes. Esta técnica permite crear nuevas vistas de la escena reutilizando partes de imágenes calculadas previamente.

Los dos últimos grupos de técnicas tratan de resolver el problema mediante la sustitución de un objeto por alguna entidad que conserve las características visuales importantes pero que sea más rápido de dibujar. A estas entidades se les denomina impostores [6]. El impostor en el modelado multirresolución es una representación geométrica de menor detalle, mientras que en la visualización basada en imágenes es una imagen la que realiza dicha función. A continuación se revisan algunos de los trabajos que utilizan esta última técnica.

2.1. Image caching

En una imagen, no todas las partes deben ser refrescadas con la misma frecuencia. En general, cuando el observador se mueve por la escena, los objetos muy lejanos parecerán estar parados, los que ya no se encuentran tan lejanos se moverán pero sin cambiar mucho su apariencia y sólo los cercanos sufrirán cambios importantes. Así, la escena se divide en fragmentos, cada fragmento genera por separado y con una frecuencia de refresco diferente un trozo de la imagen final, y todos se unen en un proceso de composición posterior.

Regan y Pose [15] introdujeron esta idea en un sistema de visualización donde la escena era fragmentada utilizando el criterio de la distancia al observador. Cada fragmento de la escena se genera con una frecuencia de refresco adecuada y almacena una información de profundidad que es constante para todos los pixeles que lo forman. Un hardware específico se encarga de juntarlos para componer la imagen final cumpliendo con la tasa de imágenes requerida por la aplicación.

Schaufler y Stürzlinger [18], y Shade et al. [22] mejoraron esta idea estableciendo una jerarquía de la escena (árboles BSP, Kd). De esta forma, cada nodo contiene una imagen que representa la vista que sus hijos almacenan. A la hora de generar una imagen de una nueva vista, un algoritmo recorre la jerarquía y decide si utilizar la imagen almacenada en el nodo o recorrer recursivamente sus hijos. Si la imagen contenida en un nodo es actualizada, también lo es la de su nodo padre. De esta forma, se pretende que sólo sea necesario llegar hasta las hojas en ciertas ocasiones y para sólo cierta parte de la escena.

2.2. Warping pixels

En 1996, Toborg y Kajiya [27] presentaron la arquitectura *Talisman* para manipular de forma eficiente la primitiva *capa*. Durante décadas, la visualización de escenas de dibujos animados o de efectos especiales en películas ha sido dividida en capas que son actualizadas independientemente y compuestas en una imagen final. Dicha arquitectura combina un módulo *render* que se encarga de generar las capas y un módulo *warp/compositer* que se encarga de combinar las capas en tiempo real después de transformar cada una (*warping*) para obtener la mejor aproximación a la imagen final deseada. Sólo cuando no es posible obtener la aproximación deseada se llama al módulo *render* para generar de nuevo la capa. El módulo *render* opera bajo demanda y a una frecuencia menor que la tasa de imágenes por segundo final. Lengyel y Snyder [11] presentan cómo aumentar las prestaciones de la arquitectura *Talisman* entre tres y diez veces más. En la figura 7 se muestra un ejemplo de imagen generada por la composición de capas.

2.3. Layered warping

Esta técnica [19] se basa en la idea de que las partes de la imagen que se encuentran a cierta distancia del observador se

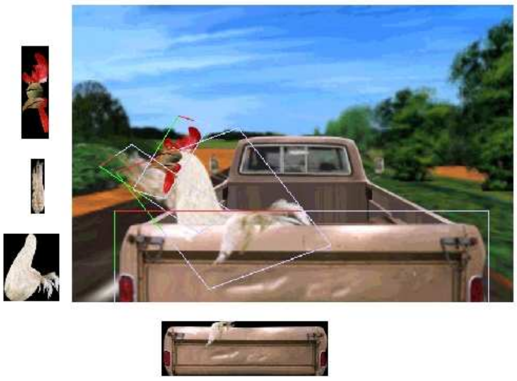

Figura 7: *Imagen generada por la composición de 80 capas* [11]*. Algunas de estas capas se muestran de forma separada y recuadradas en la imagen final.*

mueven de manera uniforme. En concreto consiste en representar a un objeto mediante un conjunto de polígonos transparentes donde cada polígono aporta aquellas partes del objeto que están a una distancia similar del observador. Cada polígono es una capa que puede imaginarse como una rebanada del objeto a cierta distancia del observador (ver figura 8). A más rebanadas, mejor representación. Cada una de ellas almacena información de color y profundidad. En cada capa, sólo se dibujan aquellas zonas donde la profundidad es cercana a la distancia desde la capa al punto de vista. A la hora de visualizar el objeto, éste es dibujado como una pila piramidal donde el punto de vista es el vértice. Los posibles agujeros que puedan aparecer como consecuencia de mover el punto de vista se evitan solapando en parte los rangos de profundidad de las capas consecutivas.

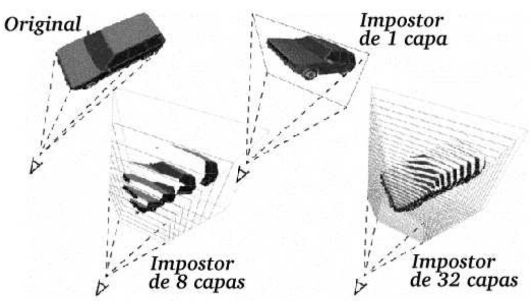

Figura 8: *Ejemplo de Layered Warping* [19]*. Sustitución de un objeto (arriba-izquierda) por un impostor (arriba-derecha) o un impostor de capas (abajo).*

3. Obtención de geometría a partir de imágenes

La reconstrucción de modelos tridimensionales a partir de imágenes ha sido desde siempre uno de los temas más importantes en el campo de la visión por ordenador [9]. Aunque no existe una técnica general que permita obtener modelos a partir de imágenes, diversas líneas de investigación han proporcionado métodos que pueden ser aplicados al modelado y visualización basado en imágenes. A continuación se revisan las líneas más relevantes.

3.1. Calibración de la cámara

La extracción de geometría a partir de imágenes es un proceso más simple cuando dichas imágenes son tomadas con cámaras calibradas. Una cámara está calibrada [8, 9] cuando se conocen sus parámetros tanto intrínsecos como extrínsecos. El conocimiento de los parámetros intrínsecos permite obtener las dimensiones de los objetos de la escena y son las coordenadas del centro de proyección en la imagen, la longitud focal, la relación ancho/alto de la lente (normalmente 1) y el ángulo del eje óptico (normalmente 90 grados). El conocimiento de los parámetros extrínsecos permite obtener la posición y orientación de los objetos en coordenadas del mundo real, y son la posición y la dirección de la cámara.

3.2. Estructura a partir de movimiento

Dada la proyección de un punto, se sabe que su posición en el espacio está restringida a pertenecer a la recta que pasa por el centro de proyecciones y su proyección. Si en lugar de una sola imagen se dispone de un conjunto de imágenes donde ese punto aparece proyectado, es posible deducir su posición en el espacio matemáticamente. Desde 1913 se han desarrollado métodos basados en esta técnica y son revisados en el libro de Faugeras [9]. Estos métodos tienen el inconveniente de ser muy sensibles al ruido cuando la nueva posición de la cámara varía muy poco de la anterior. Por otra parte, el modelo recuperado es una nube de puntos o segmentos (ver figura 9), el cual no puede ser visualizado como un modelo sólido. Los métodos basados en esta técnica sólo son útiles para IBMR si se conoce cómo unir esa nube de información.

3.3. Volumen a partir de siluetas

A partir de la silueta del objeto en la imagen y del centro de proyección de la cámara se obtiene una región del espacio infinita y en forma de cono en la cual se encuentra el objeto. Utilizando diferentes imágenes se puede calcular la intersección de las distintas regiones y obtener el volumen englobante del objeto. En la figura 10 se muestra un ejemplo donde sólo han hecho falta tres vistas para recuperar la geometría del objeto. En la figura 11 se muestra otro ejemplo donde la geometría es reconstruida a partir de una secuencia continua de imágenes.

3.4. Correspondencia estéreo

La visión estereoscópica es el conjunto de técnicas que tratan la recuperación de la información tridimensional a partir

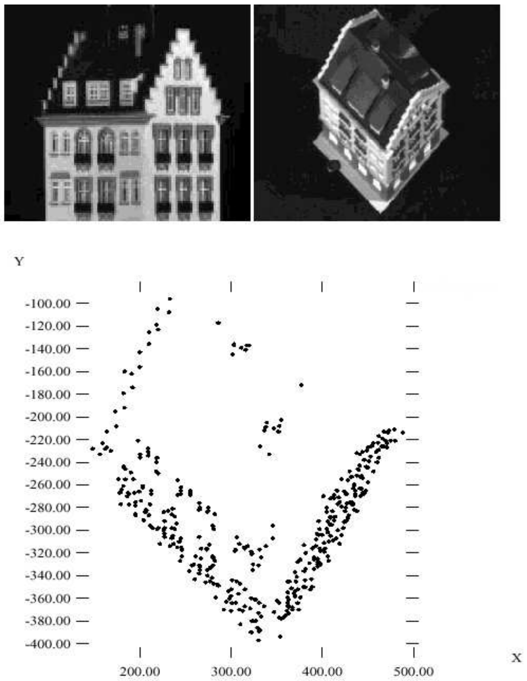

Figura 9: *Puntos obtenidos automáticamente a partir de una secuencia de imágenes de una casa girando [28]. Arriba se pueden ver dos imágenes de esa secuencia y abajo los puntos recuperados.*

de dos o más vistas de una escena. Si un objeto es observado desde dos puntos de vista diferentes la posición de las proyecciones en cada imagen de un mismo punto de la escena 3D serán diferentes. Hay dos conceptos característicos de los métodos estéreo: *baseline* y disparidad. La *baseline* es la distancia entre las posiciones de las cámaras desde las que se han obtenido las dos imágenes. La disparidad es la diferencia entre las proyecciones del mismo punto 3D en ambas imágenes. Así, si la *baseline* es pequeña las imágenes son muy similares y las profunidades obtenidas son muy sensibles al ruido. Si, por contra, la *baseline* es grande produce altas disparidades, oclusiones importantes y es más difícil establecer la correspondencia (encontrar de forma automática las dos proyecciones de un mismo punto 3D en ambas imágenes). En la figura 12 se muestra un ejemplo de escena donde la profundidad de sus elementos se ha obtenido a partir de dos imágenes.

3.5. Escáner de rango

Los escáners de rango [1] miden directamente la profundidad de varios puntos de la escena. Los primeros que aparecieron eran algo lentos y caros. Aunque su tecnología ha evolucionado, no es posible utilizarlos en escenas de exterior (edificios, monumentos, etc.). Además, es necesario de algoritmos que combinen los resultados obtenidos desde diferentes puntos de vista y, para aplicaciones de Informática Gráfica, sigue siendo necesario realizar fotografías con el fin de utilizarlas como texturas a la hora de visualizar el modelo 3D obtenido. En la figura 13 se muestra un ejemplo realizado con esta tecnología.

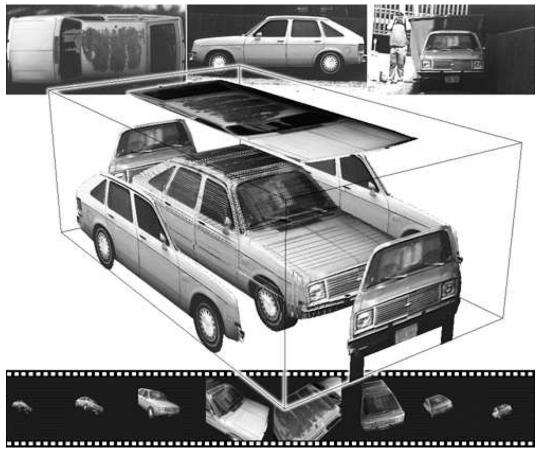

Figura 10: *Ejemplo de volumen obtenido a partir de siluetas [3]. Arriba las tres imágenes. En el centro se observa como se colocan estas imágenes en una caja para hacer el barrido y obtener el volumen. Abajo se muestran distintas vistas del objeto reconstruido.*

Figura 11: *Ejemplo de modelo obtenido a partir de siluetas [23]. A la izquierda una de las imágenes utilizada como dato de entrada, en el centro su silueta y a la derecha el modelo 3D recuperado.*

4. Transformación de vistas

En 1996, Seitz y Dyer [21] presentaron un trabajo que permite, a partir de dos imágenes, obtener nuevas vistas de la escena mediante técnicas de *morphing* asegurando una transición realista. Es decir, las nuevas imágenes obtenidas son las que se verían si realmente la cámara se moviera físicamente entre las posiciones de las dos imágenes dadas (ver figura 14). Se preserva la forma del objeto y se produce la ilusión de que el objeto se mueve (rotación y traslación 3D) rígidamente entre las posiciones de las dos imágenes. Para ello, sólo hace uso de las dos imágenes, de sus respectivas matrices de proyección y de la correspondencia entre pixeles

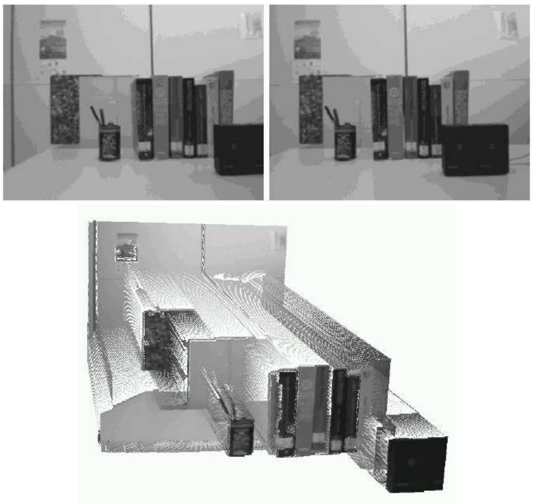

Figura 12: *Ejemplo de información 3D obtenida a partir de un método estéreo* [13]*. Arriba el par estéreo. Abajo una vista con la información 3D recuperada.*

Figura 13: *Ejemplo de modelo 3D obtenido a partir de un escáner de rango* [20]*. En la izquierda una fotografía del modelo. A la derecha el modelo 3D con la imagen utilizada como textura.*

de ambas imágenes. Esta correspondencia se realiza mediante la combinación de la interacción de un usuario y métodos de interpolación automática propuestos en técnicas de *morphing*. No se requiere ningún tipo de información 3D sobre la escena.

El método que presentan funciona mediante una predeformación de las dos imágenes (*prewarping*), cálculo de la transición entre estas imágenes predeformadas (*morphing*), y entonces una postdeformación (*postwarping*) de cada una de las imágenes producidas en el paso anterior. La figura 15 muestra un ejemplo realizado con esta técnica aplicada a caras humanas.

Figura 14: *Transformación de vistas entre dos imágenes de un objeto tomadas desde dos puntos de vista diferentes* [21]*. El resultado (imagen intermedia) produce la ilusión de mover físicamente una cámara virtual.*

Figura 15: *Transformación de vistas aplicado a caras humanas* [21]*. A la izquierda y a la derecha se encuentran las imágenes originales y entre ellas las nuevas vistas.*

4.1. Vistas paralelas

Este caso ocurre cuando la cámara se mueve en una dirección paralela al plano de la imagen aunque haya una variación del zoom (ver figura 16). Entonces, la matriz de proyección de cualquier vista intermedia es el resultado de la interpolación lineal entre las matrices de proyección de las imágenes de referencia I_0 e I_1. El color de un punto en la imagen intermedia se calcula promediando el color de los pixeles en las imágenes de referencia.

4.2. Vistas no paralelas

Este caso ocurre cuando hay un cambio del punto de vista tal y como se muestra en la figura 17. Para resolverlo, Seitz y Dyer proponen utilizar la reproyección de imágenes [29] para convertir este problema en el caso que se ha mostrado en la sección anterior (vistas paralelas). Así, dividen el proceso en tres partes:

1. Predeformación: si T_0 y T_1 son, respectivamente, las matrices que especifican la posición y orientación de las imágenes de referencia I_0 e I_1 en coordenadas del mun-

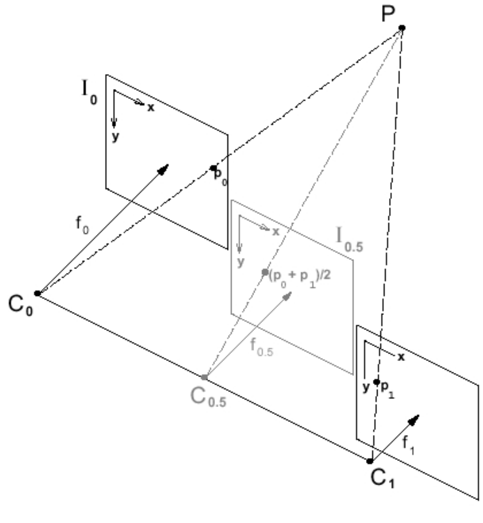

Figura 16: *Transformación de vistas paralelas* [21]. *La interpolación lineal de los pixeles correspondientes en vistas paralelas I_0 e I_1 crea la imagen $I_{0.5}$ que representa otra vista paralela de la misma escena.*

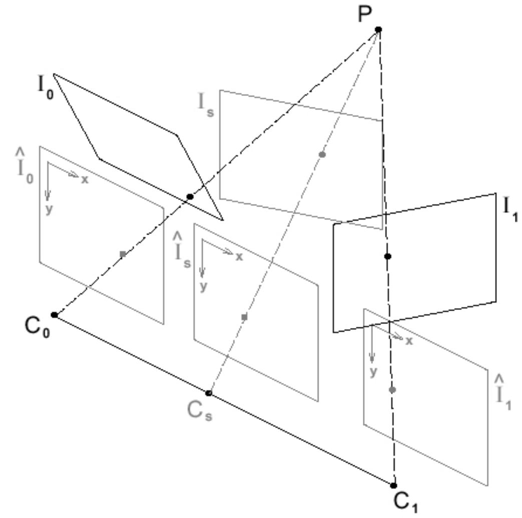

Figura 17: *Transformación de vistas no paralelas* [21]. *Las imágenes originales I_0 e I_1 son predeformadas para crear las vistas paralelas $\hat{I}_0$ e $\hat{I}_1$. $\hat{I}_s$ se crea mediante interpolación de las imágenes predeformadas. Finalmente, $\hat{I}_s$ es postdeformada en I_s.*

do real, $\hat{I}_0$ e $\hat{I}_1$ se obtienen como consecuencia de aplicar T_0^{-1} y T_1^{-1} a I_0 e I_1 respectivamente. Tras esta transformación se ha convertido este caso en el de vistas paralelas.
2. Transformación: crea $\hat{I}_s$ mediante la interpolación lineal de las posiciones y colores de los puntos en $\hat{I}_0$ e $\hat{I}_1$, y T_s mediante la interpolación lineal de T_0 y T_1.
3. Postdeformación: obtiene I_s mediante la aplicación de la transformación T_s al plano de la imagen $\hat{I}_s$.

4.3. Cambios en la visibilidad

Los cambios en la visibilidad ocurren cuando partes que están ocultas pasan a ser visibles o viceversa y dan lugar a problemas de dos tipos: dobleces y agujeros. Una doblez ocurre cuando algo que es visible en una de las imágenes originales está oculto en la imagen intermedia. El caso contrario, cuando algo oculto en la imagen original se hace visible en la imagen intermedia, da lugar a un agujero.

Las dobleces se resuelven utilizando técnicas basadas en Z-buffer. Si no hay disponible información de profundidad, ésta se sustituye por la disparidad que, en vistas paralelas, es inversamente proporcional a la profundidad.

Respecto a los agujeros, éstos se rellenan utilizando un color de fondo, interpolando con los pixeles vecinos o tomando más imágenes que cubran esas superficies.

5. Panoramas y mosaicos

En esta sección se aborda la construcción de panoramas y mosaicos [24, 26]. Tres son los pasos necesarios: toma de fotografías, registro y composición.

5.1. Toma de fotografías

La toma de fotografías puede hacerse disparando varias fotografías o bien desde el mismo punto de vista pero con diferentes direcciones (ver figura 18(a)), o bien moviendo la cámara si se fotografía una superficie plana. En cualquier caso, es necesario que exista una coincidencia en los campos de visión, entre un 20% y un 50%, para poder hacer un buen registro (explicado en la siguiente sección). A la hora de realizar las fotos es conveniente evitar lentes de ojo de pez debido a la gran distorsión que introducen, aunque sí que es conveniente utilizar lentes de gran angular. También es interesante que entre foto y foto no pase mucho tiempo para evitar que las condiciones de luz cambien y que los objetos se muevan (sobre todo si se fotografían exteriores). Es conveniente utilizar siempre la misma apertura de diafragma y tiempo de exposición a fin de obtener la misma iluminación en zonas comunes de varias fotografías. Sin embargo, no hay ningún problema en variar el zoom.

5.2. Registro

Dadas dos imágenes I_1 e I_2, el registro es el proceso por el cual se establece la correspondencia de al menos cuatro

(a) Cinco fotografías tomadas desde el mismo punto de vista pero girando la dirección de la vista.

(b) Registro de dos imágenes. Se han seleccionado cuatro pixeles en la imagen de la izquierda y sus correspondientes en la imagen de la derecha.

(c) Resultado de la composición de las imágenes que se muestran en la figura 18(b).

(d) Resultado de la composición de las imágenes que se muestran en la figura 18(a).

Figura 18: *Ejemplo de Panorama.*

pixeles de I_1 con otros tantos de I_2 para poder calcular la transformación perspectiva que los llevaría de una imagen a la otra.

Para crear un panorama, la correspondencia entre pixeles (ver figura 18(b)) se suele realizar manualmente o de forma semiautomática. Si se realiza de forma manual, el usuario ha de seleccionar los pixeles en ambas imágenes. Si se realiza de forma semiautomática, el usuario selecciona para cada pixel en I_1 una zona aproximada en I_2 (mediante una ventana rectangular) y, utilizando un método de correlación, se obtiene el punto concreto. Una vez establecida la correspondencia se procede con el cálculo de la matriz de transformación perspectiva.

5.2.1. Correlación

La correlación es la técnica que mide la similitud entre el pixel $p \in I_1$ y el pixel $p' \in I_2$. Para ello, se define la ventana de correlación centrada en p como $W(p)$. Sea N el número de pixeles de la ventana y p_i, $i = [1..N]$, los pixeles del vecindario de p.

Existen diversos criterios. El criterio ZNCC (*Zero-mean Normalized Cross-Correlation*) consiste en maximizar el coseno de dos vectores de intensidad,

$$C_{ZNCC}(p,p') = \frac{\sum_{\forall p_i \in W(p)}((I_1(p_i) - \overline{I_1(p)})(I_2(p'_i) - \overline{I_2(p')}))}{N\sigma_1(p)\sigma_2(p')} \tag{1}$$

de modo que si el resultado es 1 ambos vectores son idénticos mientras que si es -1 son totalmente opuestos.

5.2.2. Transformación perspectiva

Esta transformación 2D convierte una imagen en otra utilizando ocho parámetros y viene dada por la siguiente relación lineal:

$$x = \frac{au+bv+c}{gu+hv+1}, y = \frac{du+ev+f}{gu+hv+1} \tag{2}$$

donde (x,y) es el pixel en la imagen destino y (u,v) es el pixel de la imagen fuente. Escrito en forma matricial homogénea:

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & 1 \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \tag{3}$$

El cálculo de esta matriz de coeficientes se determina a partir de cuatro puntos que forman un cuadrilátero en la imagen fuente y sus correspondientes en la imagen destino. Estos puntos son los especificados anteriormente (ver figura 18(b)). Así, son ocho las ecuaciones que se disponen:

$$x_k = \frac{au_k+bv_k+c}{gu_k+hv_k+1}, au_k+bv_k+c-gu_kx_k-hv_kx_k = x_k \tag{4}$$

$$y_k = \frac{du_k+ev_k+f}{gu_k+hv_k+1}, du_k+ev_k+f-gu_ky_k-hv_ky_k = y_k \tag{5}$$

para $k = 0,1,2,3$. Esto puede escribirse como un sistema de 8x8:

$$M = \begin{bmatrix} u_0 & v_0 & 1 & 0 & 0 & 0 & -u_0x_0 & -v_0x_0 \\ u_1 & v_1 & 1 & 0 & 0 & 0 & -u_1x_1 & -v_1x_1 \\ u_2 & v_2 & 1 & 0 & 0 & 0 & -u_2x_2 & -v_2x_2 \\ u_3 & v_3 & 1 & 0 & 0 & 0 & -u_3x_3 & -v_3x_3 \\ 0 & 0 & 0 & u_0 & v_0 & 1 & -u_0y_0 & -v_0y_0 \\ 0 & 0 & 0 & u_1 & v_1 & 1 & -u_1y_1 & -v_1y_1 \\ 0 & 0 & 0 & u_2 & v_2 & 1 & -u_2y_2 & -v_2y_2 \\ 0 & 0 & 0 & u_3 & v_3 & 1 & -u_3y_3 & -v_3y_3 \end{bmatrix} \tag{6}$$

$$M \begin{bmatrix} a \\ b \\ c \\ d \\ e \\ f \\ g \\ h \end{bmatrix} = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ y_0 \\ y_1 \\ y_2 \\ y_3 \end{bmatrix} \tag{7}$$

5.3. Composición

Una vez obtenida la matriz de transformación perspectiva se procede con la composición de las dos imágenes para formar una sola (ver figuras 18(c) y 18(d)). Esta tarea requiere de un remuestreo 2D para obtener un buen acabado. Una forma computacionalmente barata de conseguirlo es mediante un filtro bilineal. Este tipo de filtro decide el color de un pixel mediante la interpolación lineal en x y en y de los colores establecidos en las cuatro esquinas del pixel.

Para componer panoramas cilíndricos se transforman las coordenadas (x,y,z) en una superficie cilíndrica (θ,h) donde:

$\theta = tan^{-1}(x/z)$

$h = y/(x^2+z^2)^{1/2}$

La coordenada z se suele sustituir por la longitud focal de la lente [26] y será la misma para todos los puntos de la imagen. Una vez convertidas las imágenes, la composición del panorama se consigue moviendo una imagen sobre otra hasta hacer coincidir las partes comunes.

6. Aplicación de IBMR a la Arquitectura

El modelado de objetos 3D a partir de imágenes (ver sección 3) pasa por resolver ciertos problemas que la visión por ordenador investiga desde hace años. Sin embargo, desde el punto de vista de la informática gráfica, estos problemas pueden ser evitados involucrando a un usuario para que ciertas tareas se hagan de forma manual. Por ejemplo, el problema de la correspondencia entre imágenes obtenidas desde diferentes puntos de vista se soluciona si un usuario realiza manualmente parte de esta tarea. Esto no es factible en visión por ordenador ya que precisamente lo que persigue es imitar las capacidades humanas.

Debevec et al. [4] presentaron un método para modelar objetos arquitectónicos a partir de imágenes. Este método se basa en tres técnicas principales: modelado fotogramétrico, texturación dependiente de la vista y modelado basado en estéreo. El modelado fotogramétrico se utiliza para crear un modelo básico de la escena que es utilizado posteriormente por el método estéreo para restringirlo. La texturación dependiente de la vista se utiliza para visualizar el modelo. El modelado basado en estéreo se utiliza para obtener un modelo más detallado del anteriormente obtenido con el modelado fotogramétrico. Otros trabajos realizados sobre la aplicación a la arquitectura son [7, 12].

6.1. Modelado fotogramétrico

El objetivo de esta etapa es obtener un modelo 3D simple. Este modelo estará formado por bloques. Estos bloques no son más que primitivas poliédricas paramétricas que se combinan formando una estructura jerárquica. Este tipo de modelado presenta una serie de ventajas claras: en primer lugar, la mayor parte de los modelos arquitectónicos se pueden representar mediante primitivas geométricas, los bloques definen líneas paralelas y ángulos rectos, lo cual también ocurre en los elementos arquitectónicos, en segundo lugar, las primitivas geométricas hacen que el usuario se centre en la forma general y no en el detalle y, por último, se reduce enormemente el número de parámetros que el algoritmo de reconstrucción necesitaría para recuperar el modelo 3D.

El usuario parte de un conjunto pequeño de imágenes y ha de indicar una serie de correspondencias mediante aristas. Los bloques, tamaño y posición, son calculados de forma automática a partir de esta información. Como ayuda, el usuario puede imponer restricciones a los bloques como, por ejemplo, yacer encima o ser idéntico a otro. En la figura 19 se muestra un ejemplo realizado con esta técnica, a partir de una sola imagen y unas setenta aristas marcadas se recuperan 23 bloques en aproximadamente dos horas de trabajo.

Figura 19: *Ejemplo de modelado fotogramétrico [4]. De izquierda a derecha: imagen de entrada con aristas marcadas en verde; modelo recuperado utilizando sólo la fotografía de entrada y haciendo uso de la simetría de la torre; comprobación del modelo obtenido proyectándolo en la imagen original previamente obtenida la posición de la cámara; imagen sintética generada desde un nuevo punto de vista.*

6.2. Texturación dependiente de la vista

En esta etapa se proyectan las imágenes sobre el modelo (ver figura 20). El método propuesto [5] mezcla la proyección de varias imágenes. Si un pixel en la imagen final puede ser coloreado utilizando varias de las imágenes podría utilizarse aquella que se obtuvo desde la posición de la cámara más cercana a la posición de la cámara actual. Sin embargo esto produce efectos visuales extraños debido a los materiales especulares o geometría no modelada. Para evitarlo se utilizan pesos (ver figura 21).

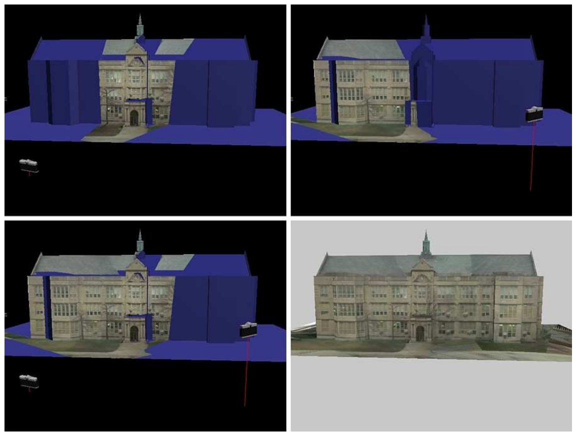

Figura 20: *Proceso de texturación [5]. Las dos figuras superiores muestran dos imágenes proyectadas sobre el modelo. La de abajo a la izquierda muestra la composición final de esas dos imágenes utilizando la texturación dependiente de la vista. La de abajo a la derecha muestra el resultado de componer un total de 12 imágenes.*

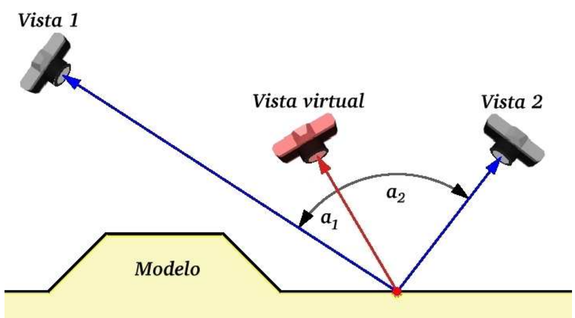

Figura 21: *Esquema para establecer los pesos en la texturación dependiente de la vista. El pixel en la vista virtual se calcula a partir de los pixeles correspondientes en las vistas 1 y 2. Los pesos son inversamente proporcionales al valor de los ángulos a_1 y a_2.*

6.3. Modelado basado en estéreo

En general, la escena con la que se está trabajando hasta el momento carece de detalle como, por ejemplo, cornisas, ventanas, etc. Para recuperar de forma automática ese detalle, Debevec propone utilizar un método estéreo que utiliza la geometría que ya se posee (la obtenida en la etapa de modelado fotogramétrico) para medir la diferencia de ésta con la de la escena real y para resolver el problema de la correspondencia. El problema que plantea Debevec es que dadas dos imágenes, una denominada *Clave* y la otra *Desplazada*, la diferencia que existe entre ambas imágenes es muy grande. Sin embargo, si se proyecta la imagen *Desplazada* sobre el modelo y éste es visualizado desde la posición de la cámara desde la que se obtuvo la imagen Clave, el resultado es muy similar a la imagen Clave (ver figura 22). A esta nueva imagen le llama *Desplazada transformada*. Así, en vez de utilizar la imagen *Desplazada* junto con la imagen *Clave* para resolver la correspondencia, utiliza la *Desplazadatransformada* y la *Clave*. Con estas dos imágenes establece la correspondencia para cada elemento, calcula la disparidad, encuentra la disparidad entre la imagen *Desplazada* y la *Clave* y, finalmente, la profundidad.

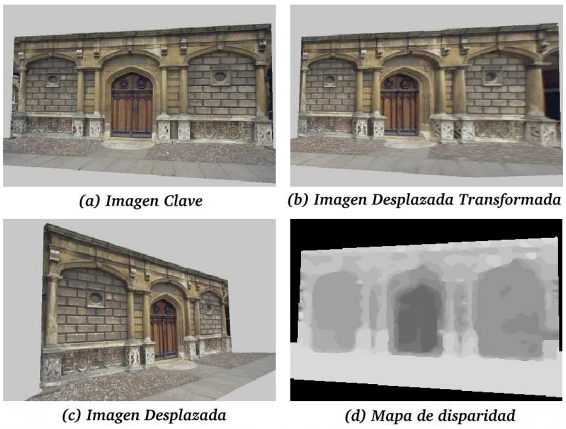

Figura 22: *Las imágenes (a) y (c) son dos fotografías del modelo real* [4]. *En (b) se observa la imagen desplazada vista desde la posición de la imagen clave (a). El mapa de disparidad se muestra en (d).*

Referencias

1. P. Besl, Active optical imaging sensors, *Advances in machine vision: architectures and applications*, Springer Verlag, 19

2. P. Debevec, *http://www.debevec.org/*

3. P. Debevec, The chevette project, *http://www.debevec.org/*

4. P. Debevec, C.J. Taylor, J. Malik, Modeling and Rendering Architecture from Photographs: A hybrid geometry- and image-based approach, *Proc. of Siggraph'96*, 11-20, August, 1996

5. P. Debevec, G. Borshukov, Y. Yu, Efficient View-Dependent Image-Based Rendering with Projective Texture-Mapping, *Proc. of 9th Eurographics Rendering Workshop*, Vienna, Austria, June 1998

6. X. Decoret, G. Schaufler, F. Sillion, J. Dorsey, Multi-layered impostors for accelerated rendering, *Proc. of Eurographics'99*, Vol 18(3), September, 1999

7. G. Dobrinov, New Algorithms for Modeling and Rendering Architecture from Photographs, *Phd Thesis*, Electrical Engineering and Computer Science Department, University of California at Berkeley, Spring, 1997

8. O. Faugeras, Q. Luong, S. Maybank, Camera self-calibration: theory and experiments, *European Conference on Computer Vision*, 321-34, 1992

9. O. Faugeras, Three-dimensional computer vision: A geometric viewpoint, *MIT Press*, Cambridge, MA, 1993

10. N. Greene, M. Kass, G. Miller, Hierarchical Z-Buffer Visibility, *Proc. of Siggraph'93*, 231-238, August, 1993

11. J. Lengyel, J. Snyder, Rendering with coherent layers, *Proc. of Siggraph'97*, 232-242, August, 1997

12. D. Liebowitz, A. Criminisi, A. Zisserman, Creating Architectural Models from Images, *Proc. of Eurographics'99*, September, Milano, 1999

13. A. López, F. Pla, J. Ribelles, 3D Modeling of Structured Scenes Through Binocular Stereo Vision, *Proc. of 12th Scandinavian Conference on Image Analysis*, 401-408, June, 2001

14. Persistence of Vision Raytracer, *http://www.povray.org/*

15. M. Regan, R. Pose, Priority rendering with a virtual reality address recalculation pipeline, *Proc. of Siggraph'94*, 155-162, July, 1994

16. J. Ribelles, Síntesis de Imagen, *Technical Report DI01-12/95*, Departamento de Informática, Universitat Jaume I, December, 1995

17. J. Ribelles, Mallas Ordenadas Multirresolución, *Phd Thesis*, Departamento de Informática, Universitat Jaume I, December, 2000

18. G. Schaufler, W. Stürzlinger, A three dimensional image cache for virtual reality, *Proc. of Eurographics'96*, 227-236, September, 1996

19. G. Schaufler, Per-object image warping with layered impostors, *Proc. of Eurographics Rendering Workshop*, 145-156, June, 1998

20. M. Schuresko, *http://www.cs.cmu.edu/∼ph/869/results/schuresko/asst2/*

21. S.M. Seitz, C.R. Dyer, View Morphing, *Proc. of Siggraph'96*, 21-30, August, 1996

22. J. Shade, D. Lischinski, D. Salesin, T. DeRose, J. Snyder, Hierarchical image caching for accelerated walkthroughs of complpex environments, *Proc. of Siggraph'96*, 75-82, August, 1996

23. R. Szeliski, Shape from rotation, *Proc. of IEEE conference on Computer Vision and Pattern Recognition*, 625-630, June, 1991

24. R. Szeliski, Image mosaicing for tele-reality applications, *IEEE Workshop on Applications of Computer Vision (WACV'94)*, 44-53, December, 1994

25. R. Szeliski, From images to models (and beyond): a personal retrospective, *Vision Interface '97*, 126-137, May, 1997

26. R. Szeliski and H.-Y. Shum, Creating full view panoramic image mosaics and texture-mapped models, *Proc. of Siggraph'97*, 251-258, August, 1997

27. J. Toborg, J. Kajiya, Talisman: Commodity real-time 3D graphics for the PC, *Proc. of Siggraph'96*, 353-364, August, 1996

28. C. Tomasi, T. Kanade, Shape and motion from image streams under orthography: a factorization method, *International Journal of Computer Vision*, 9(2):137-154, November, 1992

29. G. Wolberg, Digital Image Warping, *IEEE Computer Society Press*, Los Alamitos, CA, 1990

30. Yanghai Tsin, *http://www.cs.cmu.edu/∼ytsin/mosaicing/*

31. H. Zhang, Visibility Culling Using Hierarchical Occlusion Maps, *Course Notes 20, Siggraph '99*, 1999