



**CENTRO POLITÉCNICO SUPERIOR  
UNIVERSIDAD DE ZARAGOZA**



# **Síntesis de vistas a partir de imágenes e información 3D**

**PROYECTO FIN DE CARRERA  
INGENIERÍA INDUSTRIAL  
AUTOMATIZACIÓN INDUSTRIAL Y ROBÓTICA**

**CÉSAR SORIANO MARÍN**

**DIRECTOR: JAVIER CIVERA SANCHO  
CODIRECTOR: LUIS MONTESANO DEL CAMPO**

**DEPARTAMENTO DE INFORMÁTICA E INGENIERÍA DE SISTEMAS  
AREA DE INGENIERÍA DE SISTEMAS Y AUTOMÁTICA**

**Junio 2010  
Curso 2009/2010**

# **Síntesis de vistas a partir de imágenes e información 3D**

## RESUMEN

En los últimos años, la reconstrucción 3D a partir de fotografías está siendo cada vez más estudiada, ofreciendo modelos más cercanos a la realidad. Con las nuevas tecnologías, en la actualidad se pueden realizar con facilidad fotografías en cualquier lugar, y además gracias a internet, tener acceso a toda esa información en muy sencillo.

Con todos los datos que se tienen a disposición, se pueden obtener modelos 3D de muchísimos lugares del mundo, y a partir de ellos realizar síntesis de vistas, que consiste en obtener imágenes de lugares desde donde no se tomaron fotografías. Lo que se consigue con la síntesis de vistas, es poder observar esos lugares desde nuevas perspectivas, e incluso poder realizar videos con movimiento fotorrealista alrededor de la escena.

Para comenzar a sintetizar vistas, primero es necesario obtener un modelo 3D de la escena, obteniendo la localización en el espacio de las cámaras y de puntos característicos observados en las fotografías.

Los algoritmos desarrollados en el proyecto permiten utilizar la información de la reconstrucción en 3D, y ubicar cámaras virtuales en distintas posiciones en las que se puede proyectar la escena, para obtener nuevas imágenes.

Durante la memoria, se desarrollan diferentes métodos de síntesis de vistas, realizando una comparativa entre calidad de imagen y tiempo de procesado, obteniendo una solución de compromiso en la que se obtenga un buen resultado gráfico sin un tiempo excesivo en su computación.

Para finalizar, los algoritmos se evalúan en distintas escenas de interior y exterior, obteniendo resultados muy similares, por lo que se demuestra que los métodos son robustos.

La memoria está acompañada de un DVD en el que se pueden observar distintos videos en varias escenas, además de los algoritmos desarrollados.

<b>Índice</b>	
1. Introducción .....	1
2. Aplicaciones.....	4
2.1. Free Viewpoint Television .....	4
2.2. Google Street View .....	5
2.3. Photo Tourism .....	6
3. Reconstrucción 3D a partir de imágenes .....	7
3.1. Reconstrucción 3D no densa.....	7
3.2. Reconstrucción 3D densa.....	9
4. Síntesis de vistas mediante impostores planares .....	11
4.1.Paso 1: Reconstrucción no densa .....	12
4.2.Paso 2: Obtención de puntos, plano dominante y posiciones.....	13
4.3.Paso 3: Proyección de imágenes en el plano dominante.....	17
4.4.Paso 4: Proyección en la imagen sintética.....	20
4.5.Visualización de las imágenes sintéticas.....	21
4.6.Limitaciones de síntesis de vistas con plano dominante.....	23
5. Síntesis de vistas mediante triangulación de Delaunay .....	25
5.1.Implementación de la síntesis de vistas mediante triangulación de Delaunay .....	26
5.2.Limitaciones de la síntesis de vistas mediante triangulación de Delaunay .....	29
6. Síntesis de vistas mediante triangulación de Delaunay densa.....	31
7. Resultados experimentales.....	34
7.1.Habitación de lectura.....	34
7.2.Rana Gustavo .....	38
7.3.Cuarto Completo.....	42
7.4.Paseo Independencia.....	46
7.5.El Pilar .....	47
8. Conclusiones .....	49
9. Bibliografía .....	51
Anexos.....	53
A. Manual de instrucciones .....	54

## 1. Introducción

La síntesis de vistas consiste en crear imágenes sintéticas a partir de imágenes reales. En este proyecto se utiliza un método en el que por ejemplo, dos fotografías que tienen partes en común de una escena, se toman con una separación de dos metros (figura 1.1). Y se realizan los cálculos para determinar cómo se vería en el punto medio de las dos fotografías, a un metro de cada una. (figura 1.2)



Figura 1.1 Fotografías separadas 2 metros



Figura 1.2 Fotografía a calcular en el punto medio a un metro de separación

El objetivo del proyecto es, sintetizar vistas en lugares donde no se tomaron imágenes, a partir de un conjunto de dichas imágenes e información 3D también extraída de ellas. Con las suficientes vistas sintéticas, creadas entre dos imágenes reales, se conseguirá realizar videos que crearán una sensación de movimiento 3D fotorrealista. El efecto que se crea en el espectador es el de estar moviéndose realmente a través de una escena en lugar de visualizar fotos separadas sin relación entre ellas.

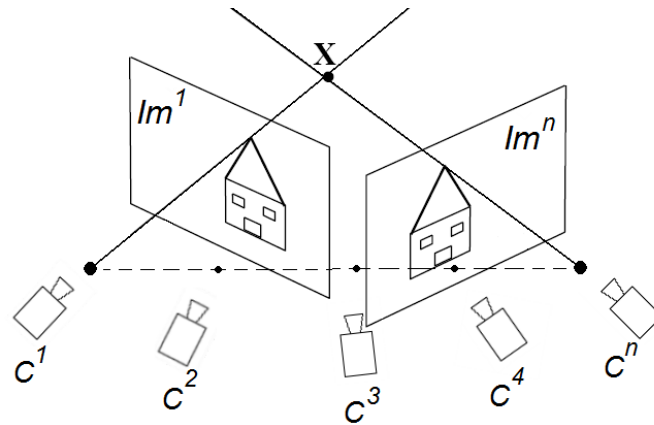


Figura 1.3 Escena 3D [13]

La idea general se muestra en la figura 1.3, inicialmente se tienen dos imágenes,  $Im^1$  e  $Im^n$ , de una misma escena, proyectadas sobre unas cámaras  $C^1$  y  $C^n$ , de las cuales se conoce su posición. Desde la posición de la cámara  $C^1$  a la  $C^n$ , se traza una línea recta en la que se situarán las cámaras virtuales  $C^2$ ,  $C^3$ , etc. En los lugares donde se sitúan estas cámaras no fueron tomadas fotografías, por lo que se deberán sintetizar las imágenes. Para crear estas imágenes, se necesitará la localización 3D de los puntos que hay en la escena, como por ejemplo, la del punto  $X$  que se muestra en la figura 1.1. Con esta información, se podrán crear imágenes en las cámaras virtuales que se coloquen a lo largo del recorrido.

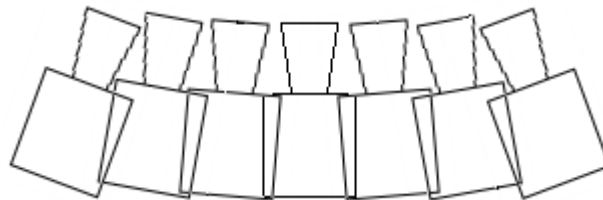


Figura 1.4 Movimiento de cámara

En la figura 1.4 se observa que si las cámaras virtuales se sitúan lo suficientemente cerca, se podrán realizar videos con movimiento fluido.

Para empezar a desarrollar el proyecto, el primer paso es obtener una reconstrucción 3D de una escena a partir de fotografías donde se localizarán las cámaras y los puntos característicos de la escena en el espacio (figura 1.5).

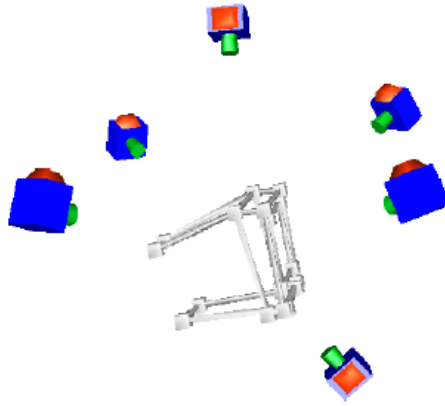


Figura 1.5 Reconstrucción 3D [13]

Se realizarán dos tipos de reconstrucción, una simple en la que solamente se reconstruyen algunos puntos, y otra en la que se añaden más puntos y se parece más a la realidad. La primera se utilizará en los dos primeros métodos estudiados, el método del plano mayoritario, y el método de la triangulación de Delaunay. La segunda se utilizará para el método de triangulación de Delaunay, pero a partir de la reconstrucción densa.

Finalmente, se realizará un estudio de calidades y tiempos para los distintos métodos argumentando cuál es el mejor.

En los anexos se incluye un manual para poder utilizar la aplicación creada.

Este proyecto está englobado en el grupo de robótica, percepción y tiempo real del Departamento de Informática e Ingeniería de Sistemas (DIIS) de la Universidad de Zaragoza. Una de sus líneas de investigación es la reconstrucción 3D, que es donde se sitúa el estudio realizado.

## 2. Aplicaciones

La tecnología aplicada a los campos de visión ha avanzado a grandes pasos últimamente, por lo que se tienen numerosas imágenes de casi cualquier parte del mundo debido al gran número de cámaras de fotos y teléfonos con cámara que hay en la actualidad.

Con toda esta información, cada vez se están generando mayores utilidades que ofrecer, por lo que cada vez surgen nuevas aplicaciones, y es en lo que se basa el siguiente proyecto. Algunas de ellas son las siguientes.

### 2.1. Free Viewpoint Television

Una de las más llamativas es el sistema llamado Free Viewpoint Television [3] (FTV, Televisión de punto de vista libre), que es un sistema de información en 3D que engloba adquisición, procesado y reproducción de imágenes. (figura 2.1)

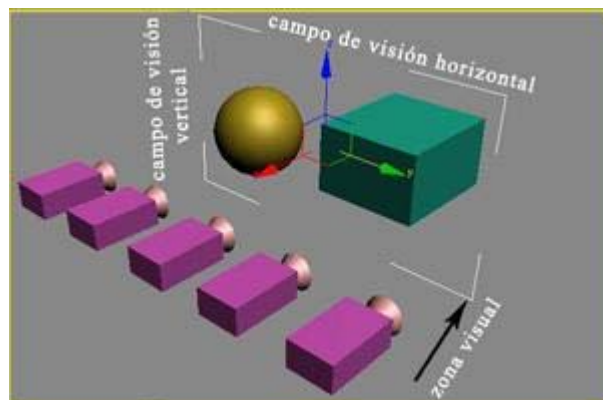


Figura 2.1 Free Viewpoint Television

FTV pretende crear una nueva época en la historia de la televisión porque permite ver libremente un mundo real, de 3D, únicamente cambiando el punto de vista tal y como se haría estando presente en la escena adquirida. Se trata de un sistema en el que el usuario elige qué es lo que quiere ver de una escena. Así, este método, se puede aplicar a amplias áreas tales como el entretenimiento, la observación de la naturaleza, el turismo, el arte, los museos, la educación, la medicina, la seguridad, etcétera.

## 2.2. Google Street View

Otra aplicación bastante conocida es Google Street View [4], el sistema de Google para viajar por las calles de numerosas conocidas ciudades a través de fotografías tomadas con un vehículo que pasea por dichas calles. (figura 2.2)



Figura 2.2 Adquisición de fotografías de Google Street View

Después de haber tomado todas las fotografías, se componen para poder tener desde cada punto en el que se sitúe el usuario una vista de 360°, y además, poder realizar una transición entre posiciones consecutivas (figura 2.3), aunque como se puede comprobar la transición es muy difuminada, (figura 2.4) y es una característica que se pretende mejorar en el proyecto.



Figura 2.3 Dos posiciones del paseo Independencia de Zaragoza en Google Street View



Figura 2.4 Imagen de transición de dos vistas consecutivas en Google Street View



### 2.3. Photo Tourism

Finalmente, otra aplicación es Photo Tourism de Microsoft [1,2]. Una aplicación que consiste en coger colecciones de fotos de internet, de lugares representativos de ciudades (figura 2.5 a), y reconstruirlos en 3D (figura 2.5 b) para poder visitar todos sus rincones (figura 2.5 c)

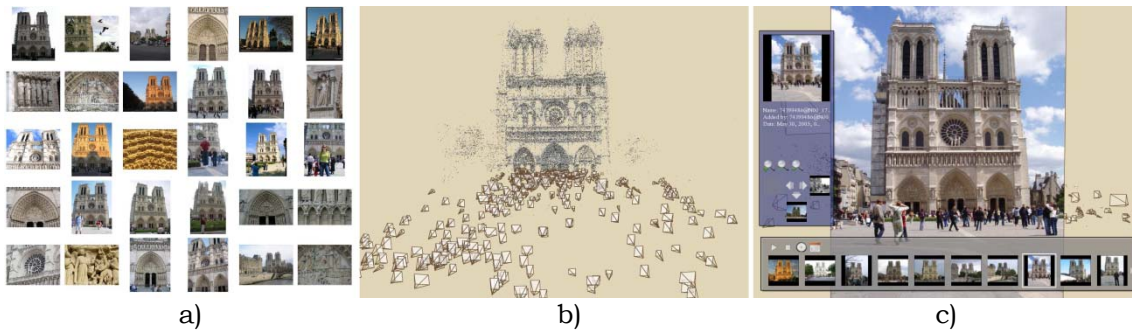


Figura 2.5 a) Colección de fotografías b) Situación de puntos y cámaras en el espacio  
c) Visualización de fotografías.

En este sistema el usuario se puede situar en la posición de una de las cámaras, y viajar hasta otra (figura 2.6) a través de una transición continua (figura 2.7) en la que la imagen inicial va desapareciendo y la final va apareciendo.

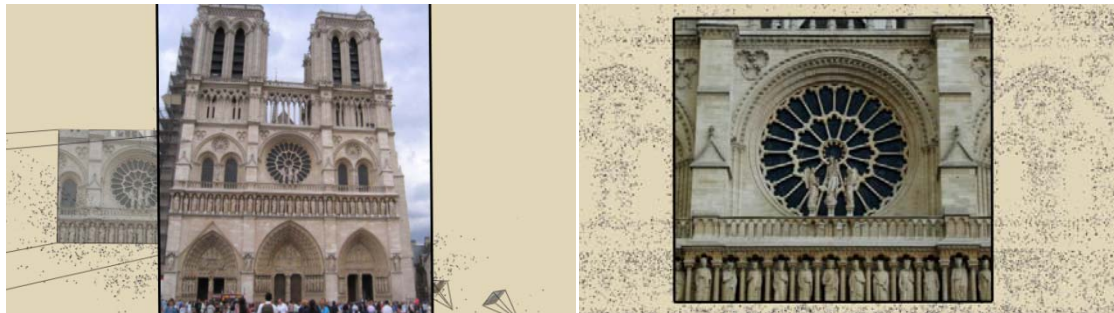


Figura 2.6 Posiciones inicial y final sobre la que hacer la transición en Photo Tourism



Figura 2.7 Imagen de transición de dos vistas consecutivas en Photo Tourism

### 3. Reconstrucción 3D a partir de imágenes

Durante la memoria de este proyecto, se trabajará con las mismas fotografías para poder comprobar las diferencias que se produzcan entre los distintos métodos estudiados. La escena se trata de una habitación con una estantería, un sofá y una mesita que consiste en 11 fotografías (figura 3.1).



Figura 3.1 Set de fotografías

Para la síntesis de vistas, en primer lugar es necesario realizar una reconstrucción en 3D, en la que se realizan dos pasos. Primero se realiza una reconstrucción no densa, y a partir de ella, se realiza una reconstrucción densa. Ambas se explican a continuación.

#### 3.1. Reconstrucción 3D no densa

El primer paso para realizar una reconstrucción 3D no densa es extraer y emparejar los puntos característicos en cada imagen. Para lo cual se utiliza un detector de puntos SIFT [8], los cuales se pueden ver en la figura 3.2.

La primera imagen contiene el conjunto de puntos  $x = \{x_1, \dots, x_m\}$ , y los puntos de la segunda imagen son  $x' = \{x'_1, \dots, x'_n\}$

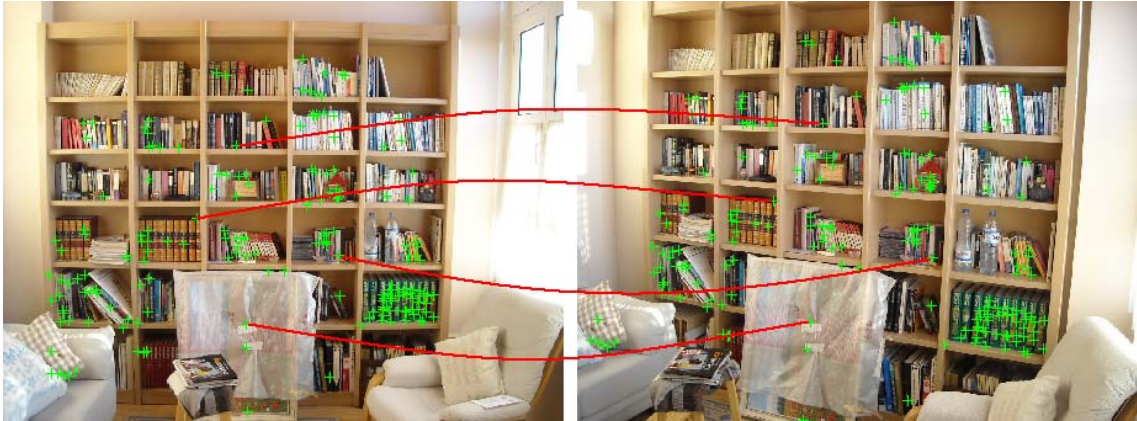


Figura 3.2 Puntos característicos SIFT con sus correspondencias

Para relacionar los píxeles de ambas cámaras se utiliza la matriz fundamental  $F$ , que contiene los parámetros de las cámaras, y la rotación y traslación entre ambas. Cumple la siguiente relación.

$$x' \cdot F \cdot x = 0$$

Esta matriz tiene siete grados de libertad, por lo que serán necesarios siete puntos que se puedan reconocer en cada fotografía para conseguir la matriz fundamental. También se necesitan al menos dos fotografías, y que cada punto que se desee situar en el espacio, esté al menos en dos fotografías. El paralaje óptimo entre fotografías para poder definir correctamente la profundidad de cada punto es  $90^\circ$ .

Con este sistema se consigue la localización de todos los puntos y todas las cámaras en la escena 3D.

Para realizar todo este proceso se utiliza el programa Bundler [10], que es un sistema de estructura a partir del movimiento para colecciones de imágenes desordenadas, como por ejemplo, imágenes de internet. Bundler toma una serie de imágenes, parámetros de la imagen y relaciones de imagen como entrada, y produce una reconstrucción 3D de la geometría de las cámaras y de la escena. Automáticamente sitúa en el espacio dónde están situadas las cámaras desde donde se realizaron las fotografías, y una serie de puntos característicos en el espacio.

Los puntos en el espacio creados por este método pueden observarse en la figura 3.3.

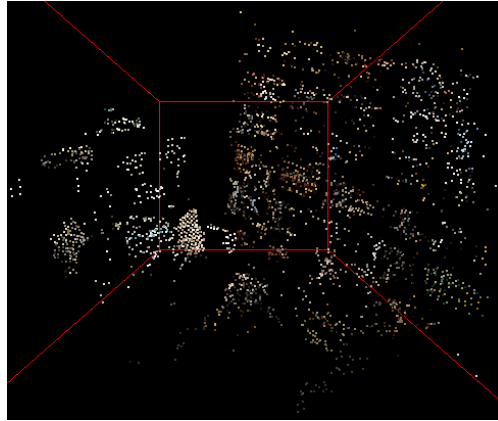


Figura 3.3 Reconstrucción 3D densa

### 3.2. Reconstrucción 3D densa

Como se puede comprobar, la nube de puntos de la figura 3.3 no es muy densa e incluso es difícil diferenciar la escena, para lo cual será necesario densificar dichos puntos.

Para densificar se utiliza un sistema basado en parches [9], los cuales son unos rectángulos centrados en cada punto de la reconstrucción, y con vector normal orientado hacia la cámara (figura 3.4 a)). Cada parche  $p$ , se asocia a la imagen desde la que se ve  $R(p)$ . También se añade a las imágenes desde las que deberían ser visibles, pero puede que no sean reconocibles  $S(p)$ , y se detecta en las imágenes desde las que realmente se ve  $T(p)$  (figura 3.4 b)).

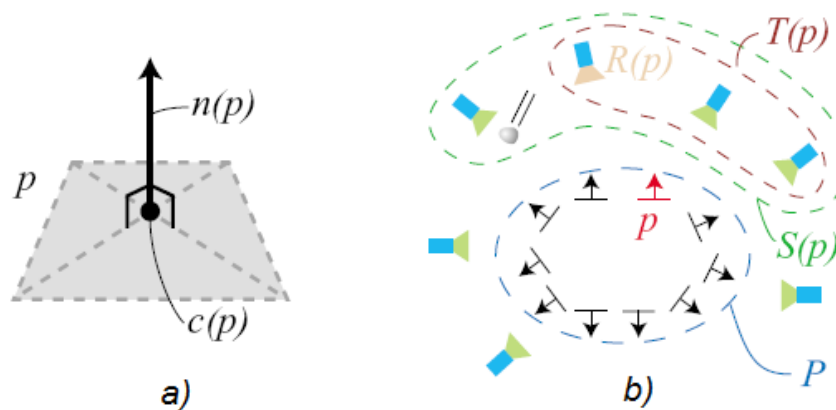


Figura 3.4 Modelo de parches [9]

Cuando se crea el modelo de parches, posteriormente se procede a su expansión. Se añaden vecinos a los parches que se han creado, hasta que se cubren las superficies visibles en la escena.

Finalmente se hace un filtrado en el que según las superficies estimadas, se detecta si los parches añadidos se encuentran dentro de ella o

no, aceptando todo aquél que se encuentre dentro de la superficie estimada, y eliminando los que caigan fuera de ella (figura 3.5).

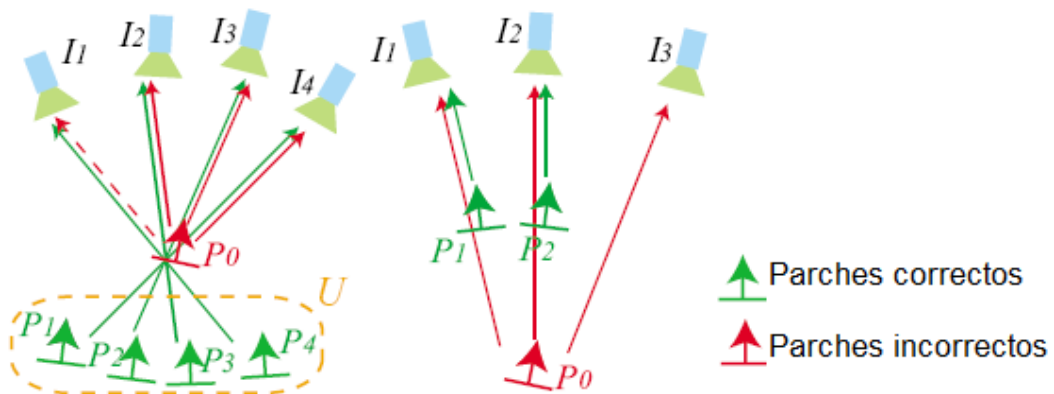


Figura 3.5 Modelo de expansión de parches [9]

Finalmente se obtiene una reconstrucción densa como se puede apreciar en la figura 3.6

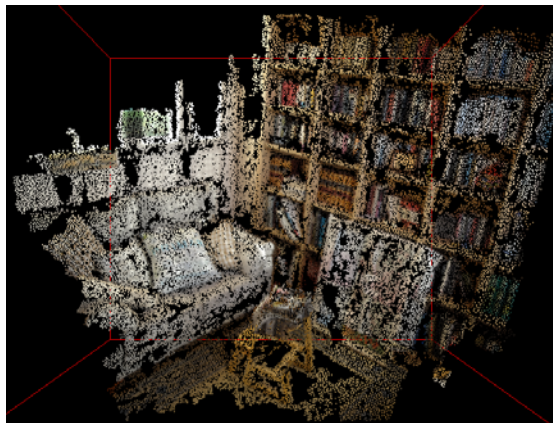


Figura 3.6 Reconstrucción 3D densa

Para realizar el modelo 3D denso, se utiliza el software PMVS2 [11], que crea el modelo de la figura 3.6, con el algoritmo explicado, a partir de los datos obtenidos del software Bundler anteriormente mencionado.

#### 4. Síntesis de vistas mediante impostores planares

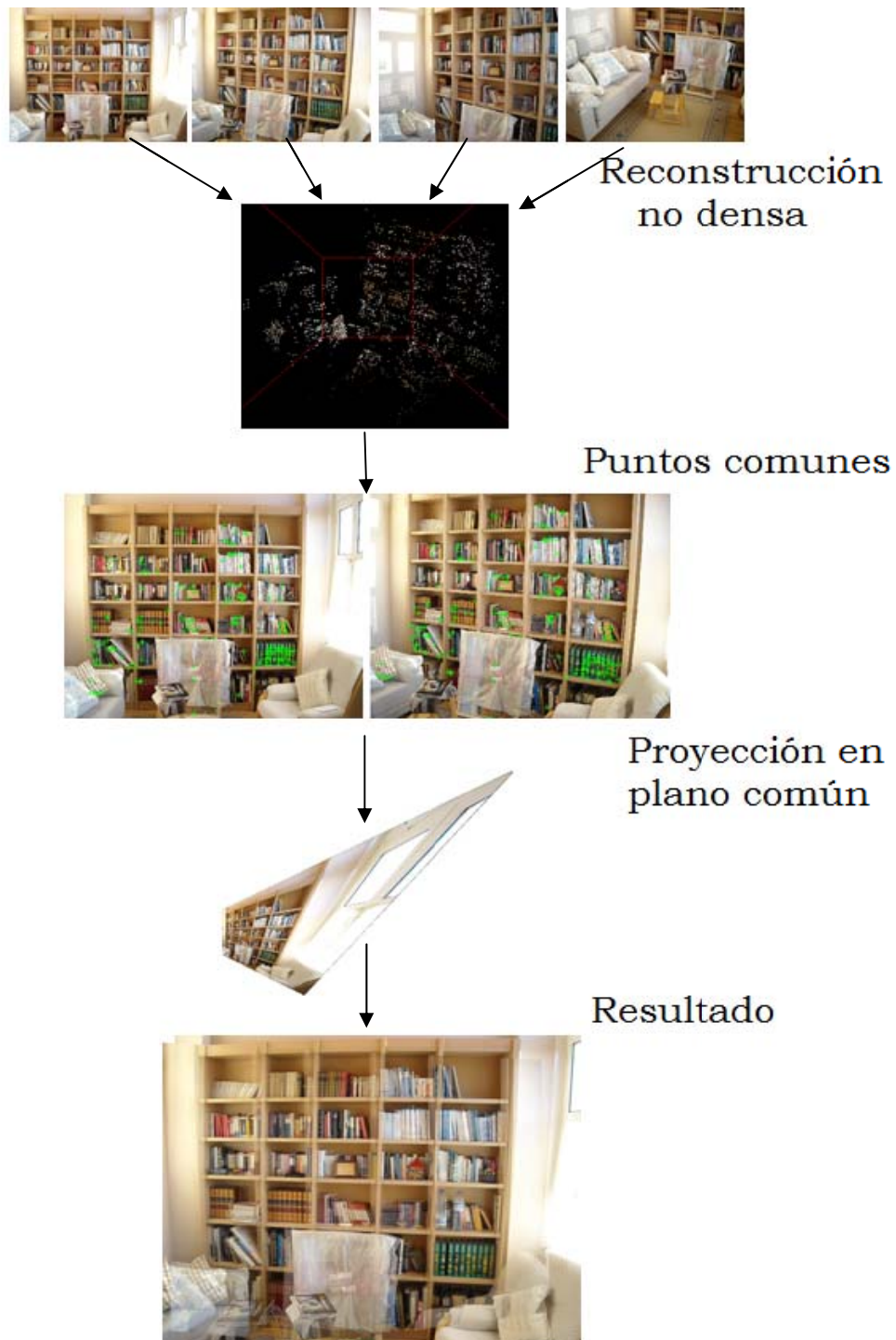


Figura 4.1 Diagrama de flujo

El método de síntesis de vistas mediante impostores planares, consiste en obtener un plano dominante común a las dos fotografías sobre las que se quiere hacer la transición. El proceso a seguir se observa en la figura 4.1 y se explica a continuación.

Inicialmente se tiene que hacer una reconstrucción 3D no densa para saber dónde se sitúan los puntos y las cámaras en el espacio.

Con esta información de la reconstrucción, se obtiene qué puntos son comunes a ambas cámaras. A partir de ahí se obtiene el plano que mejor encaje con la mayoría de ellos.

En este plano, se proyectan las imágenes que se ven desde las cámaras. Desde ese plano, se proyecta la vista en las cámaras ficticias situadas en el movimiento que se produce de una cámara a otra, mientras se realiza un desvanecimiento cruzado. La primera imagen se va desvaneciendo, mientras la segunda va apareciendo.

Las imágenes intermedias resultantes no son completamente fieles a la geometría real, ya que toda la imagen se proyecta sobre el plano dominante en la escena, pero los resultados son generalmente fácilmente asumibles.

A continuación se explica el algoritmo paso a paso.

#### **4.1. Paso 1: Reconstrucción no densa**

Como se ha explicado en el apartado 3.1 se realiza la reconstrucción en 3D no densa, donde se obtienen los puntos de la figura 4.2. A la derecha de la imagen se encuentra la estantería y a la izquierda el sofá. A pesar de ser una reconstrucción poco densa y que es difícilmente reconocible, los puntos son suficientes para realizar la síntesis de vistas.

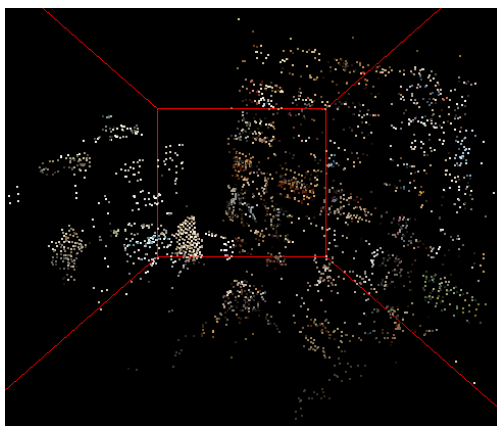


Figura 4.2 Nube de puntos

#### 4.2. Paso 2: Obtención de puntos, plano dominante y posiciones

- Obtención de puntos de cada cámara:

Con la información de la reconstrucción no densa que proporciona Bundler, se ordena para que esté todo distribuido por cámaras. Cuando se seleccione una cámara, allí se obtendrá toda su información con los puntos que se ven desde dicha cámara. Para ello, se crea un registro para que cada cámara tenga su información por separado, los datos característicos y de situación de la cámara y los puntos que se ven desde ella.

Posteriormente, con un control de errores, se comprueba que las cámaras sucesivas por las que se va a pasar, tengan al menos tres puntos en común, y además se comprueba que la cámara haya sido correctamente situada en el espacio, ya que si alguna imagen guarda poca relación con el resto, el reconstructor en 3D es incapaz de situarla.

Con los datos ordenados y sabiendo que no se va a tener ningún error en las transiciones de imagen, se comienza a realizar el algoritmo para moverse de una cámara a otra. Por lo tanto, sólo habrá dos cámaras en procesamiento a la vez. Se escogen dos cámaras, por ejemplo las de la figura 4.3, se comparan los puntos en 3D que se ven desde cada cámara, y se obtienen los puntos comunes a ambas cámaras, que se pueden ver en la figura 4.4.



Figura 4.3 Imágenes iniciales





Figura 4.4 Puntos característicos

- Obtención del plano dominante:

Con esos puntos se utiliza el método Ransac [5] para obtener el plano dominante que mejor encaja con ellos. Para ello es necesario que ambas cámaras tengan al menos tres puntos en común, lo cual se ha comprobado en el anterior control de errores.

Se decide utilizar el método Ransac ya que es un ajuste muy robusto cuando hay puntos alejados de la zona principal sobre la que hacer el ajuste. La explicación matemática siguiente se presenta sobre una recta, ya que de esta forma es más fácil su comprensión. Posteriormente se puede extrapolar a cualquier forma geométrica, en este caso se utiliza el plano.

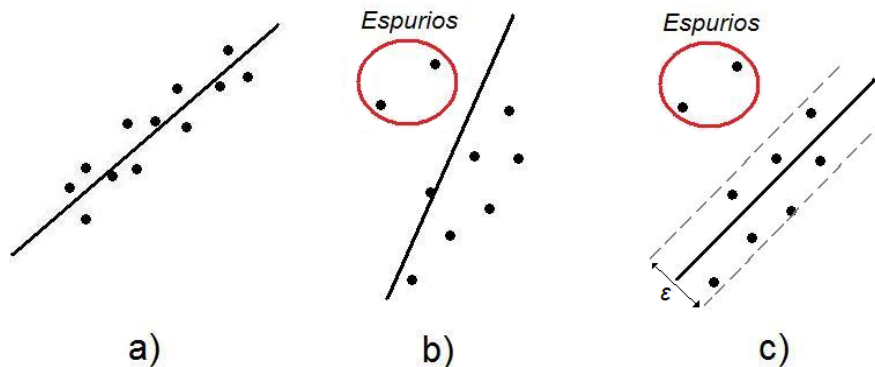


Figura 4.5 a) Ajuste mediante mínimos cuadrados. b) Ajuste mediante mínimos cuadrados con puntos espurios c) Ajuste mediante Ransac

Cuando se realiza un ajuste por mínimos cuadrados a partir de puntos que se conocen están cercanos a una recta (figura 4.5 a), el ajuste es muy bueno. Pero cuando se encuentra algún dato alejado del resto de datos principales (figura 4.5 b), el ajuste por mínimos cuadrados tiene en cuenta todos los puntos, y los resultados no son válidos al producirse un apalancamiento por estos datos espurios. Realizando un ajuste por Ransac (figura 4.5 c), el ajuste sólo tiene en cuenta los puntos correctos y solamente hace el ajuste a partir de ellos.

El algoritmo es el siguiente:

- Se selecciona el número mínimo de puntos  $m$ . Dos para una recta, y tres para un plano.
- Se determina cuantos puntos de todo el conjunto encajan en una tolerancia predefinida  $\varepsilon$ .
- Si la fracción del número de puntos correctos sobre el total de puntos excede el valor umbral  $\tau$ , se reestiman los parámetros del modelo utilizando todos los puntos correctos identificados y se termina.
- Si no, se repiten los pasos anteriores, un máximo de  $N$  veces.

El número de iteraciones  $N$ , se elige para asegurar que la probabilidad  $p$  (generalmente 0.99) de al menos uno de los conjuntos de muestras aleatorias no incluye un dato alejado de los principales. Siendo  $u$  la probabilidad de obtener un punto correcto y  $v = 1 - u$  la probabilidad de obtener un dato alejado, el número mínimo de iteraciones es:

$$1 - p = (1 - u^m)^N$$

Manipulando la ecuación se obtiene:

$$N = \frac{\log(1 - p)}{\log(1 - (1 - v)^m)}$$

Para el caso de la síntesis de vistas, se obtendrán tres puntos mediante Ransac, que son los que forman el plano. Cuando se han obtenido los que mejor ajustan el modelo, se obtiene el plano que pasa por esos tres puntos, que será donde se proyectarán las imágenes. En este caso, el plano dominante se sitúa en la estantería.

- Obtención de las posiciones de cámara:

El siguiente paso es obtener la interpolación de la posición y la rotación entre una cámara y otra. Teniendo la matriz de traslación  $t$ , y la matriz de rotación  $R$  de cada cámara a sintetizar, se puede obtener la posición de las cámaras de la forma:

$$\text{Posición} = -R' * t$$

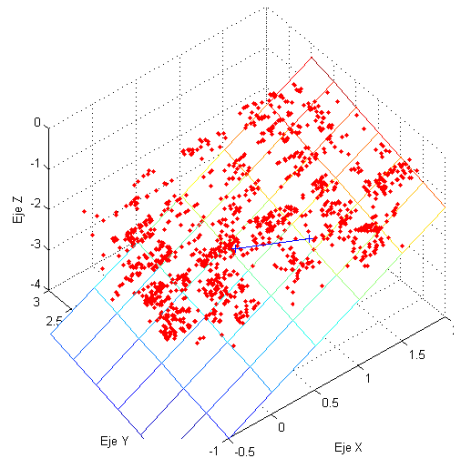


Figura 4.6 Puntos en plano

En la figura 4.6 se observan puntos rojos que son los situados en el plano dominante, y la línea azul son las posiciones de cámara sobre las que se calcularán las imágenes sintéticas entre las imágenes inicial y final.

Para obtener las posiciones entre las dos cámaras, se interpola. La matriz de traslación se interpola, ya que son simplemente coordenadas. Las matrices de rotación deben ser necesariamente ortonormales. La interpolación lineal entre una matriz de rotación inicial y otra final lleva a matrices intermedias no ortonormales y, por lo tanto, que no corresponden a matrices de rotación.

Se procede a transformar la matriz a ángulos de roll-pitch-yaw. La utilización de estos ángulos, además de ser una representación más compacta, no presenta ningún inconveniente al interpolarlos, ya que muestran los ángulos de la rotación respecto al mundo. Así, para pasar de una orientación inicial  $(\alpha_i, \beta_i, \gamma_i)$  a una final  $(\alpha_f, \beta_f, \gamma_f)$  se utilizan las siguientes funciones lineales:

$$\alpha(t) = (\alpha_f - \alpha_i) \frac{t - t_i}{t_f - t_i} + \alpha_i$$

$$\beta(t) = (\beta_f - \beta_i) \frac{t - t_i}{t_f - t_i} + \beta_i$$

$$\gamma(t) = (\gamma_f - \gamma_i) \frac{t - t_i}{t_f - t_i} + \gamma_i$$

Donde  $t_i$  y  $t_f$  son los instantes de tiempo en los que se sitúa en la orientación inicial y final respectivamente. El parámetro  $t$ , variará linealmente entre  $t_i$  y  $t_f$  según el número de pasos que haya entre imagen e imagen. Además la trayectoria del punto de vista es bastante intuitiva.

Para cada punto que se obtiene en la interpolación, se vuelve a transformar en la matriz de rotación, y con ello, obtener la posición y rotación de todas las cámaras ficticias que se encuentran en el camino entre la cámara inicial y la final. Cuantos más puntos se calculen entre las dos cámaras más fluido será el movimiento que se obtenga

#### 4.3. Paso 3: Proyección de imágenes en el plano dominante

Para este punto, hay que tener presente, qué significa una imagen en una cámara, para ello se utiliza el modelo de cámara pinhole [6]. Una imagen está situada en el espacio, separada por una distancia focal  $f$  del centro de la cámara (figura 4.7). Pero con una sola imagen, es imposible saber qué profundidad tiene cada punto de esa imagen. Lo que sí se puede saber es la recta que se proyecta desde el centro de la cámara  $O$ , y que pasa por cada punto  $m$  de la imagen. Como se va a proyectar dicha imagen en el plano dominante, no es necesario saber su profundidad. Bastará con obtener la intersección  $M$  de dicha recta con el plano. Comprendido esto, a continuación se explica su fundamento matemático.

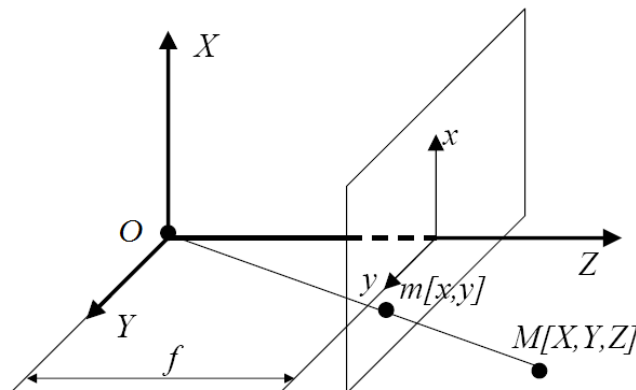


Figura 4.7 Esquema de cámara pinhole [13]

Los superíndices C, W e I, son los utilizados para indicar la cámara, el mundo y la imagen respectivamente.

Cuando se tiene un punto en el espacio  $X_W=(x,y,z)$ , para poder obtenerlo en coordenadas de la cámara, el proceso a realizar, sería el siguiente:

$$P_C = R_{CW} * X_W + t_{CW}$$

O también se puede realizar:

$$P_C = T_{CW} * X_W$$

Donde  $T_{CW}$  es un matriz 4x4:

$$\begin{bmatrix} R_{CW} & t_{CW} \\ 0 & 1 \end{bmatrix}$$

Se divide por la tercera componente para obtener el punto en el plano de la cámara en coordenadas homogéneas.

$$p_C = -P_C / Pz_C$$

Finalmente se multiplica por la focal y se elimina la distorsión radial, para pasar a coordenadas píxel de la imagen.

$$p'_I = f * rp * p_C$$

Sin embargo, lo que se pretende realizar, es pasar dichas coordenadas píxel a su situación en el mundo y buscar su proyección en el plano, por lo que el proceso es el inverso. Así que en principio se obtienen unas coordenadas píxel, que se llamarán w y h, longitud y altura respectivamente.

Se divide por la focal para pasar a las coordenadas de la cámara:

$$p_C = [w \ h]_I / f$$

Se convierte la coordenada en homogénea. Esa coordenada está situada en el plano de la cámara y no se conoce su profundidad, por lo que se le añade un 1, que sería el resultado anterior de dividir por la tercera coordenada:

$$p_C = [x' \ y' \ 1]$$

Para pasar de coordenadas de la cámara a coordenadas del mundo:

$$X_W = T_{WC} * p_C$$

Donde  $T_{WC} = T_{CW}^{-1}$

Como no se conoce la profundidad, ese no es el punto situado realmente en el mundo, pero se sabe que estará en la recta formada por el centro de la cámara y ese punto, que es el que se quiere obtener. Para lo cual, habrá que obtener dicha recta. Hay distintas formas de representar una recta en el espacio. Pero la elegida es la que está expresada mediante la intersección de dos planos.

Conocidos dos puntos de esa recta, el centro de la cámara, y el punto que hemos situado en el espacio.  $P = (p_1, p_2, p_3)$  y  $Q = (q_1, q_2, q_3)$ . La dirección de la recta está definida por  $v = Q - P$ . La ecuación vectorial de la recta será:

$$(x, y, z) = P + t * v$$

Donde la recta se forma al recorrer el parámetro  $t$ . Separando la ecuación en sus tres términos, se obtiene:

$$x = p1 + t * v1$$

$$y = p2 + t * v2$$

$$z = p3 + t * v3$$

Donde despejando  $t$ , se obtiene:

$$\frac{x - p1}{v1} = \frac{y - p2}{v2} = \frac{z - p3}{v3}$$

Donde sólo son necesarias dos ecuaciones para obtener la recta.

$$\frac{x}{v1} - \frac{y}{v2} = \frac{p1}{v1} - \frac{p2}{v2}$$

$$\frac{x}{v1} - \frac{z}{v3} = \frac{p1}{v1} - \frac{p3}{v3}$$

La tercera ecuación sería la del plano obtenido por Ransac:

$$A * x + B * y + C * z + D = 0$$

Por lo que se tiene un sistema lineal, que al resolverlo, se obtiene el punto que corta la recta con el plano.

Sólo será necesario proyectar las cuatro esquinas de la imagen en el plano, ya que la imagen es plana y se proyecta por igual en el mismo plano, por lo que toda la imagen se modificará según sus esquinas. De esta forma se podrá transformar la imagen de los ejes de la cámara a los ejes del mundo situados en el plano dominante. Se hará esto mismo para ambas imágenes, la inicial y la final sobre las que estamos calculando la transición. La imagen proyectada en el plano dominante se observa en la figura 4.8.



Figura 4.8 Imagen proyectada en el plano

#### 4.4. Paso 4: Proyección en la imagen sintética

Las dos imágenes proyectadas en el plano se mantienen estáticas durante el movimiento entre las dos cámaras. Lo que se mueve es el punto de vista. Para lo que es necesario hacer una retroproyección (figura 4.9) de las esquinas situadas en el plano, a las cámaras ficticias que se sitúan entre las cámaras originales. Esto se realizará con las ecuaciones anteriormente descritas para pasar puntos del mundo a la cámara que se esté calculando, sabiendo su posición y orientación.

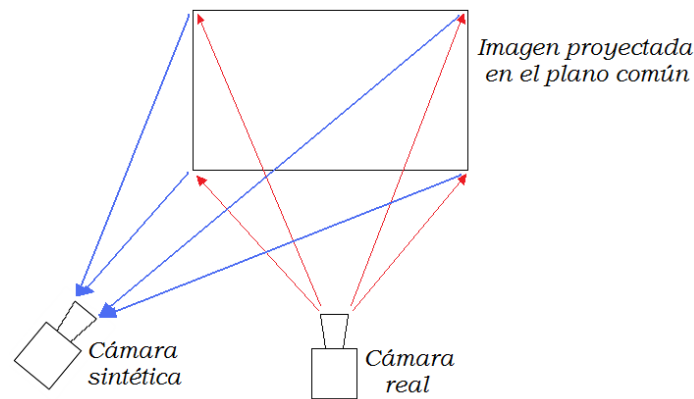


Figura 4.9 Retroproyección de imagen

Este proceso se repite para cada una de las cámaras que se desee calcular. Partiendo de la primera cámara, inicialmente se mostrará sólo la primera imagen, y la otra se mantendrá transparente. Posteriormente, se pasa al siguiente punto calculado entre la imagen inicial y la final, donde se transforman las imágenes situadas en el plano dominante, según cómo están proyectadas las esquinas en la posición actual. Mientras se traslada la cámara desde la posición inicial a la final, la imagen inicial se va desvaneciendo, y la imagen final va apareciendo. De esta forma, al principio se verá la imagen inicial, durante el proceso se muestra una mezcla de las dos imágenes superpuestas y finalmente la imagen final. En la zona común de ambas fotografías, donde se obtiene la mezcla de las imágenes superpuestas, se realiza de la siguiente forma:

$$\text{Imagen} = \text{Alfa1} * \text{Imagen1} + \text{Alfa2} * \text{Imagen2}$$

Donde Alfa1 y Alfa2 son complementarios para sumar uno. Inicialmente Alfa1 = 1, lo que quiere decir que es opaco, y Alfa2 = 0, lo que quiere decir que es transparente. En la traslación, el valor de Alfa1 va disminuyendo, haciéndose transparente la imagen inicial, y Alfa2 va aumentando, volviéndose opaca la imagen final.

En la figura 4.10 se puede ver una transición calculada entre dos fotografías.



Figura 4.10 Imagen de transición

Cada imagen que es calculada se introduce en un video donde se obtiene el resultado del movimiento.

Con una calidad de video de 12 fps (fotogramas por segundo), se obtiene un video suficientemente fluido. Como comparación, los videos mostrados en internet suelen variar entre 8 y 12 fps, y la gran mayoría de películas tienen 24 fps. Dejando 2 segundos por transición para comprobar con exactitud el movimiento, serán necesarias 22 imágenes intermedias, que con la imagen inicial y la final, se tienen 24 fotogramas para completar los 2 segundos de transición.

#### **4.5. Visualización de las imágenes sintéticas**

Con el algoritmo ya realizado, se procede a realizar diferentes cambios y para mejorar la visualización. Una modificación realizada, es que en las zonas no comunes, en vez de difuminarse, se mantenga la imagen original. De esta forma, hay más contraste entre la zona común, y la zona con imagen original, aunque en movimientos cortos, la imagen se mantiene fiel, e incluso, al verse con más relleno la imagen da sensación de reconstrucción más densa (figura 4.11).





Figura 4.11 Imagen densa comparada con la imagen no densa

Otra modificación es realizar una línea alrededor del cambio de imagen, aunque como se comprueba, se remarca demasiado el cambio en vez de difuminarlo, y la reconstrucción es peor (figura 4.12).



Figura 4.12 Línea remarcada

En vez de remarcar una línea, por la misma zona de separación entre imágenes se realiza un filtro mediana por la misma zona, con el cual, las imperfecciones de cambio de colores, se difuminan ligeramente teniendo un acabado mejor. También se eliminan ligeras líneas blancas de la transformación de la imagen (figura 4.13).

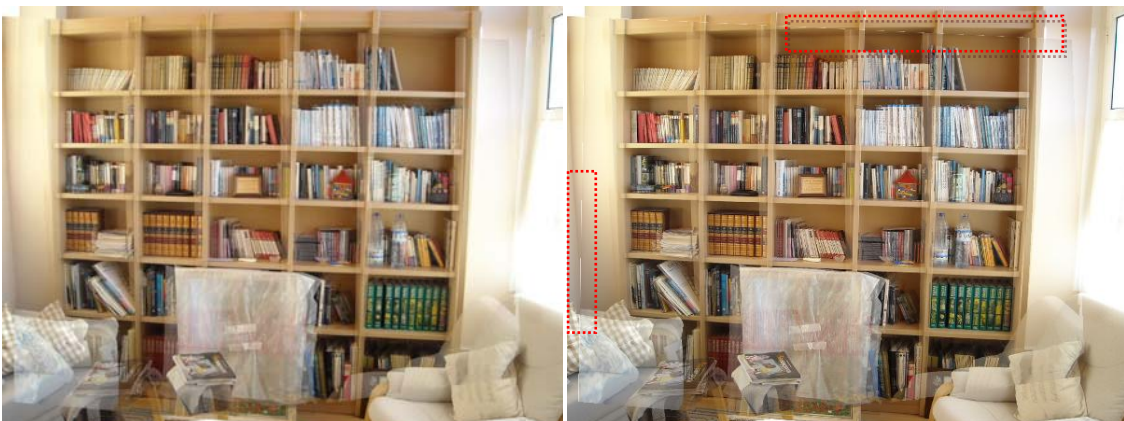


Figura 4.13 Imagen densa con filtro comparada con la imagen sin filtro

También se puede dar el caso de tener imágenes horizontales y verticales, para lo cual hay que cambiar los ejes. El alto de la imagen se verá del mismo tamaño, por lo que se ve una imagen reducida (figura 4.14).



Figura 4.14 Imágenes de distinto tamaño

Finalmente se puede dar el caso de tener imágenes de distintas cámaras de fotos, por lo que pueden tener distintas distancias focales, y distintos tamaños de fotografía. Para la focal se arregla interpolando también este parámetro al interpolar la posición. Para los distintos tamaños de fotografía, se reduce la imagen asegurando el alto, igual que se ha asegurado para el caso de imágenes verticales.

#### 4.6. Limitaciones de síntesis de vistas mediante plano dominante

El método presenta una solución bastante buena, aunque como se ha explicado, todo lo que se aleja del plano dominante contiene distorsiones visuales aunque son fácilmente asumibles.

Otro problema que surge es el caso de que no haya un plano dominante claro, como se puede comprobar en la figura 4.15.



Figura 4.15 Imagen sin plano dominante claro

En esta escena al haber puntos en diferentes planos, no está claro cuál será el dominante, y todos los demás puntos que se encuentren fuera de ese plano, presentarán una distorsión excesiva, por lo que lo conveniente es encontrar otro método en el que se reproduzca mejor la geometría espacial.

## 5. Síntesis de vistas mediante triangulación de Delaunay

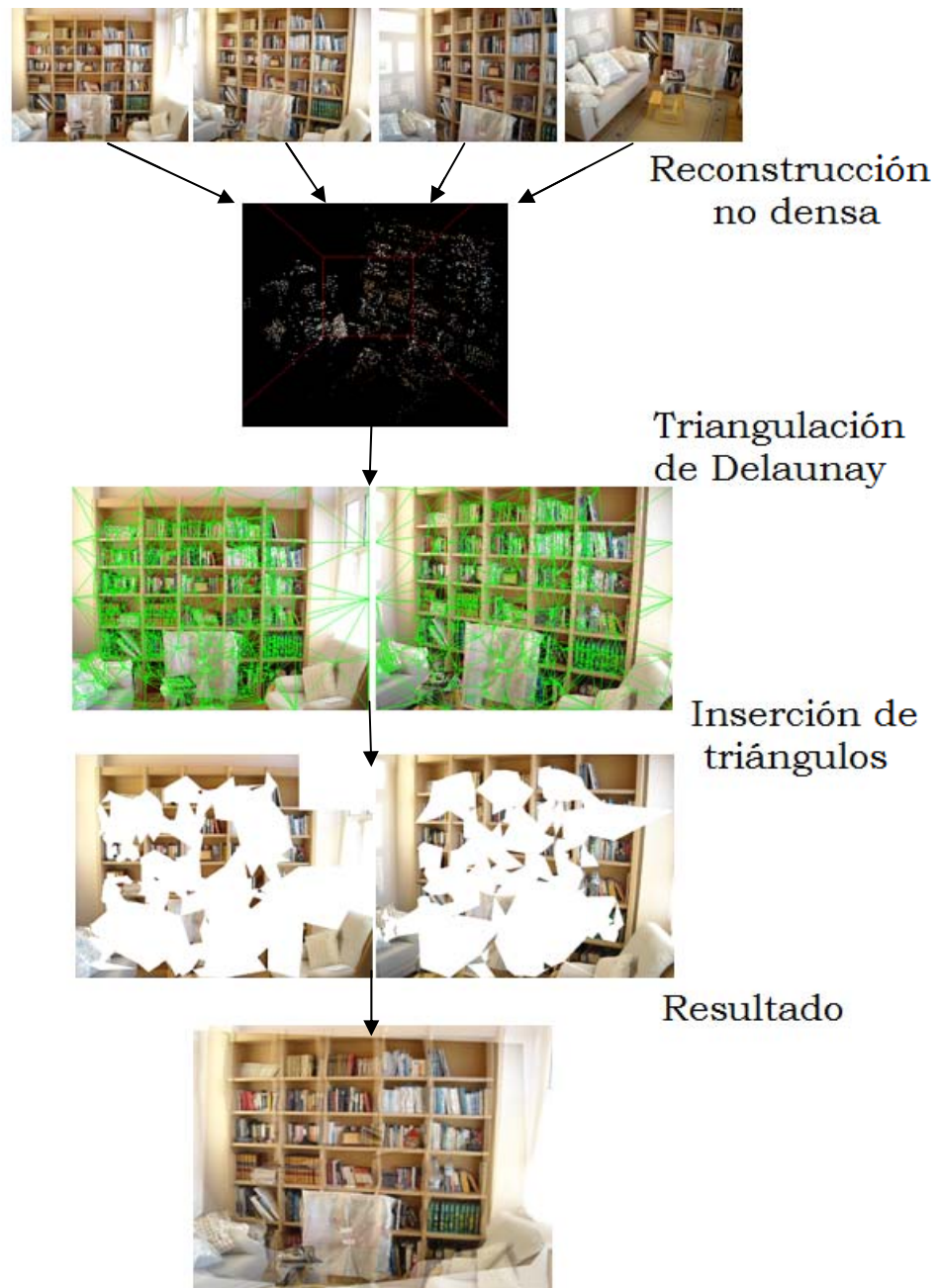


Figura 5.1 Diagrama de flujo

Para solucionar las carencias de la síntesis de vistas anterior, se decide realizar otro método para obtener el movimiento entre imágenes. El método utilizado es el de síntesis de vistas mediante triangulación de la nube de puntos (triangular morphs), para obtener una superficie más próxima a la real sobre la que proyectar, en vez de simplemente un plano dominante.

Partiendo de dos cámaras, se realizan los cálculos de triangulaciones 2D para cada imagen de cada cámara usando las proyecciones de los puntos del mundo en la imagen.

Las triangulaciones impuestas resultantes puede que no cubran la imagen entera, así que se superpone una cuadrícula en cada imagen y se añade cada punto no contenido dentro de la triangulación original. Cada punto añadido es asociado a un punto 3D en el plano dominante obtenido mediante Ransac a partir de los puntos de cada imagen. Con los nuevos puntos introducidos, se crea la triangulación en 2D para la imagen completa.

Para hacer la transición entre las dos cámaras, se mueve la cámara virtual desde la inicial hasta la final mientras se hace una decoloración cruzada entre las dos triangulaciones. La imagen inicial se desvanece mientras que la imagen final aparece gradualmente.

Esta técnica no utiliza geometría completamente acertada, pero los mallados por la triangulación son suficientes para dar una sensación de 3D a la geometría de la escena. Aunque a veces puede causar elementos con grandes deformaciones.

### 5.1. Implementación de la síntesis de vistas mediante triangulación de Delaunay

El método en gran parte es similar al anterior, por lo que se van a describir solamente las modificaciones realizadas.

Se comienza leyendo los datos de la cámara y realizando el control de errores igual que en el método anterior. Se obtienen los puntos en el 3D que ve cada cámara, y su proyección en la imagen 2D en coordenadas pixel (figura 5.2).

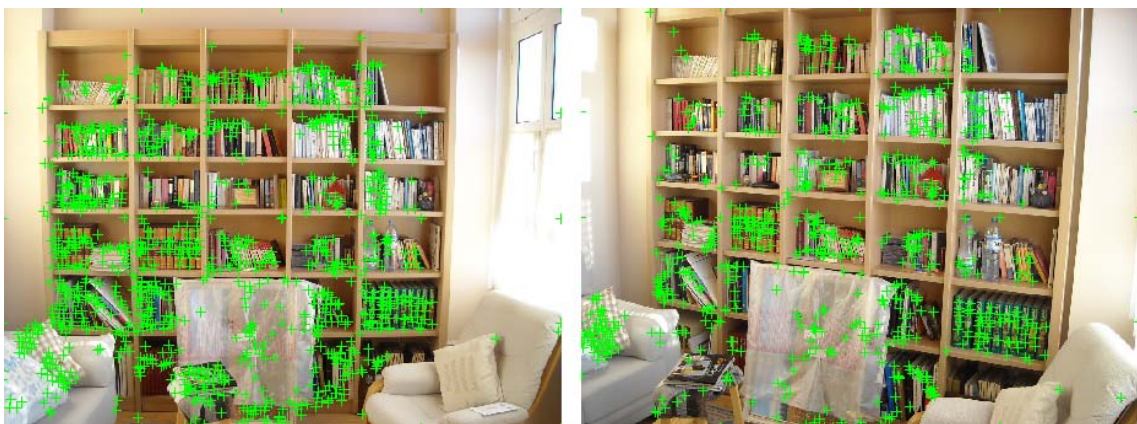


Figura 5.2 Puntos en la imagen

Posteriormente se obtiene el plano dominante de cada cámara por separado mediante Ransac, en vez de obtener el plano de los puntos comunes

de las dos cámaras como se hacía antes. En esos planos de las cámaras, se introducen los puntos que quedan fuera de la nube de puntos, hasta las esquinas. También se obtiene su proyección en coordenadas pixel de la imagen. Con esas proyecciones en el 2D, se realiza el mallado a partir de una triangulación, utilizando el método de Delaunay [7] para las dos imágenes (figura 5.3).

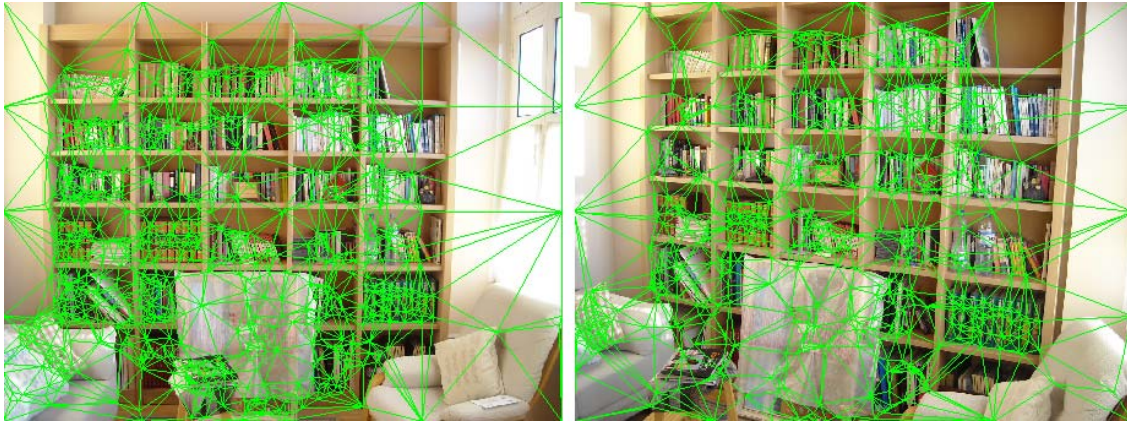


Figura 5.3 Triangulaciones de Delaunay

Una triangulación de Delaunay, es una red de triángulos que cumple la condición de Delaunay. La condición es que la circunferencia circunscrita de cada triángulo de la red no debe contener ningún vértice de otro triángulo, con lo que se consigue maximizar los ángulos de los triángulos.

Para realizar la triangulación, se utiliza el método incremental, donde se parte de un triángulo, y se van añadiendo puntos uno a uno, cumpliendo siempre la condición de Delaunay. Primero se localiza el triángulo que contiene el nuevo punto (figura 5.4 a). Se unen nuevas aristas a los vértices (figura 5.4 b) y se comprueba la condición de Delaunay para todas ellas. Si se satisface la condición, la arista se mantiene (figura 5.4 c). Si no se cumple la condición de Delaunay (figura 5.4 d), la arista es eliminada y se reemplaza por otra línea de los vértices que le rodean (figura 5.4 e). El proceso continúa hasta que todos los triángulo cumplen con la condición de Delaunay, obteniendo la triangulación resultante (figura 5.4 f).

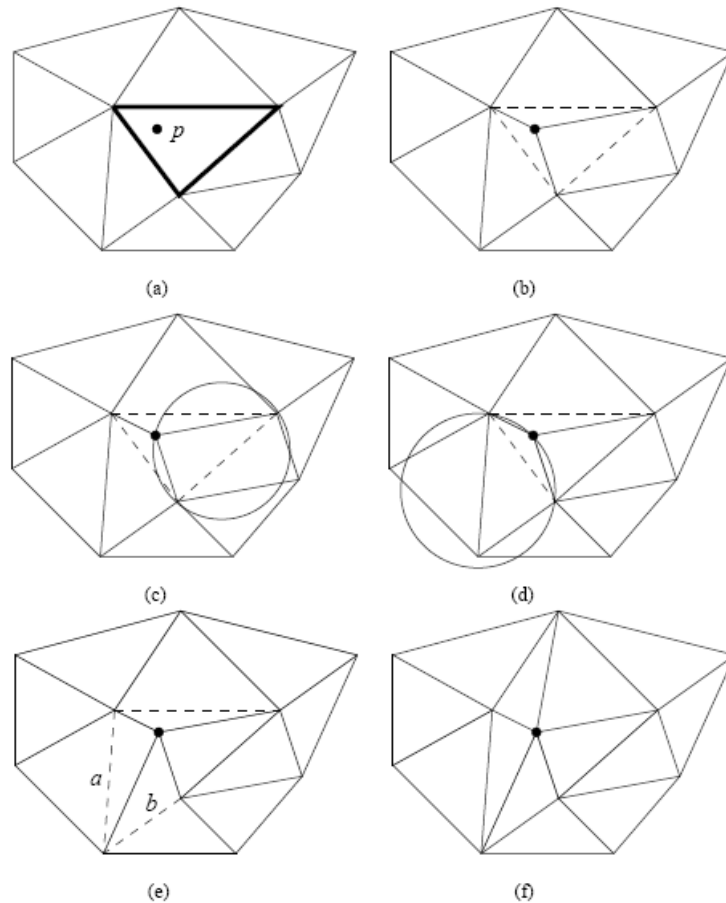


Figura 5.4 Formación de la triangulación de Delaunay

Después de haber obtenido la triangulación de Delaunay, el siguiente paso es mover la cámara, haciendo los cálculos de las cámaras ficticias que hay en medio del camino, para lo cual se vuelve a interpolar la posición y rotación de esas cámaras ficticias igual que antes.

Para cada posición se proyectan los puntos que hay en el 3D vistos desde las dos cámaras originales, en la cámara ficticia.

La gran diferencia se produce ahora, donde en vez de transformar la imagen entera, se transforma cada triángulo de la triangulación de Delaunay, anteriormente calculada, por separado. Se conocen las coordenadas de cada punto en la imagen, y se calculan las coordenadas que ocuparán en la imagen sintética para cada posición de cámara calculada. De esta forma, se recorren todos los triángulos realizando dicha transformación, para obtener la imagen transformada. Se realizará para las dos imágenes, la inicial y la final, añadiendo la transparencia necesaria según la posición en que se encuentre.

En la figura 5.5 se observa cómo se van introduciendo las transformaciones de los triángulos.

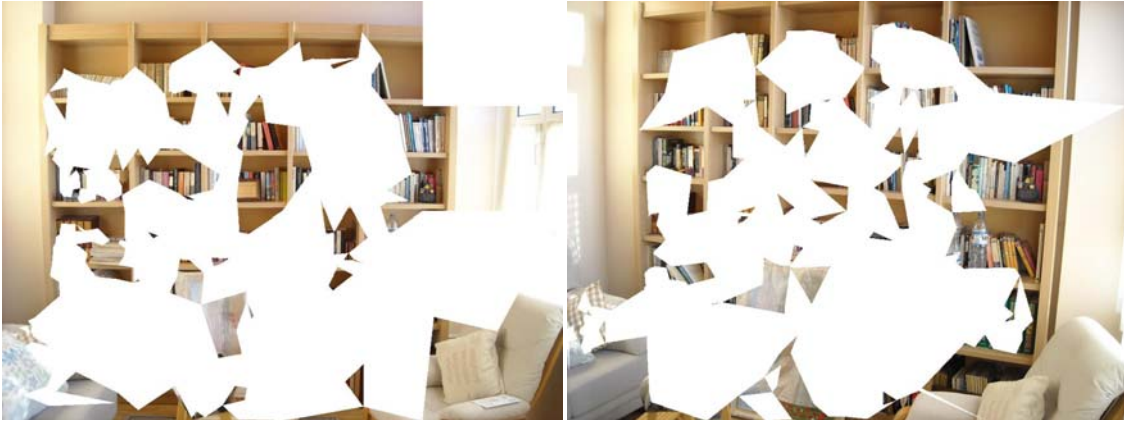


Figura 5.5 Introducción de triángulos

Mientras se realiza el movimiento, la imagen inicial se va desvaneciendo, y la imagen final va apareciendo, igual que sucedía en el método anterior. Una imagen intermedia se aprecia en la figura 5.6.



Figura 5.6 Síntesis mediante triangulación y con síntesis de plano dominante

### 5.2. Limitaciones de la síntesis de vistas mediante triangulación de Delaunay

En este método se puede comprobar que los objetos que se encuentran alejados del plano mayoritario, ahora se reproducen más fielmente a la realidad, aunque se observan dos problemas importantes. Uno de los problemas es, que al reconstruir parte a parte la imagen, no se tiene una clara continuidad en los objetos de la escena y hay partes que se aprecian ligeramente deformadas, lo que provoca una distorsión visual que con el anterior método no ocurría (figura 5.6). El otro problema se produce si la reconstrucción que se obtiene no es del todo precisa, y algún punto se ha detectado alejado de su posición real, la triangulación la detecta alejada de sus vecinos y se produce una gran deformación.



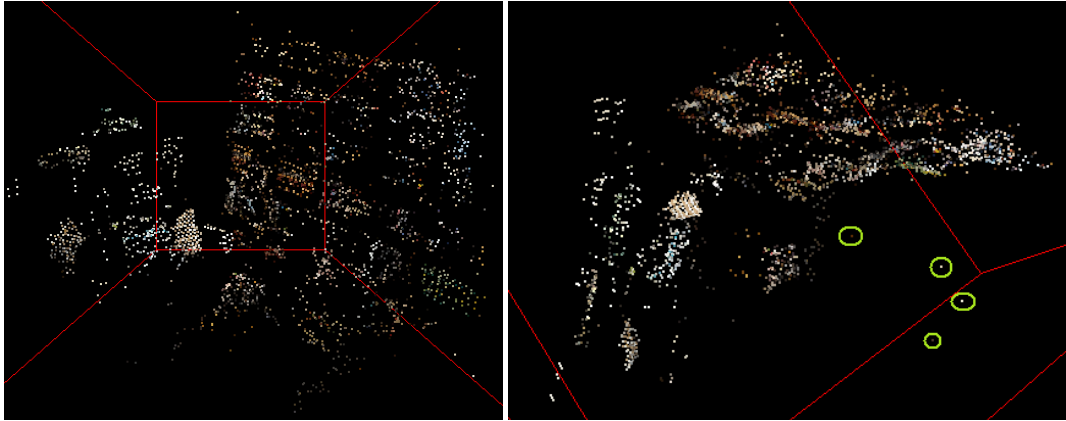


Figura 5.7 Reconstrucción no densa

En la figura 5.7 se han utilizado los puntos de la reconstrucción no densa para reconstruir la escena. En la imagen izquierda se puede comprobar que es prácticamente irreconocible la escena. Y en la imagen de la derecha, se aprecia que hay píxeles aislados, marcados con un círculo verde, culpables de las grandes deformaciones que se observan en la reconstrucción de transición de una imagen a otra. La nube de puntos no es suficientemente densa, y tampoco es totalmente fiel a la realidad, por lo que no es del todo válida.

## 6. Síntesis de vistas mediante triangulación de Delaunay densa

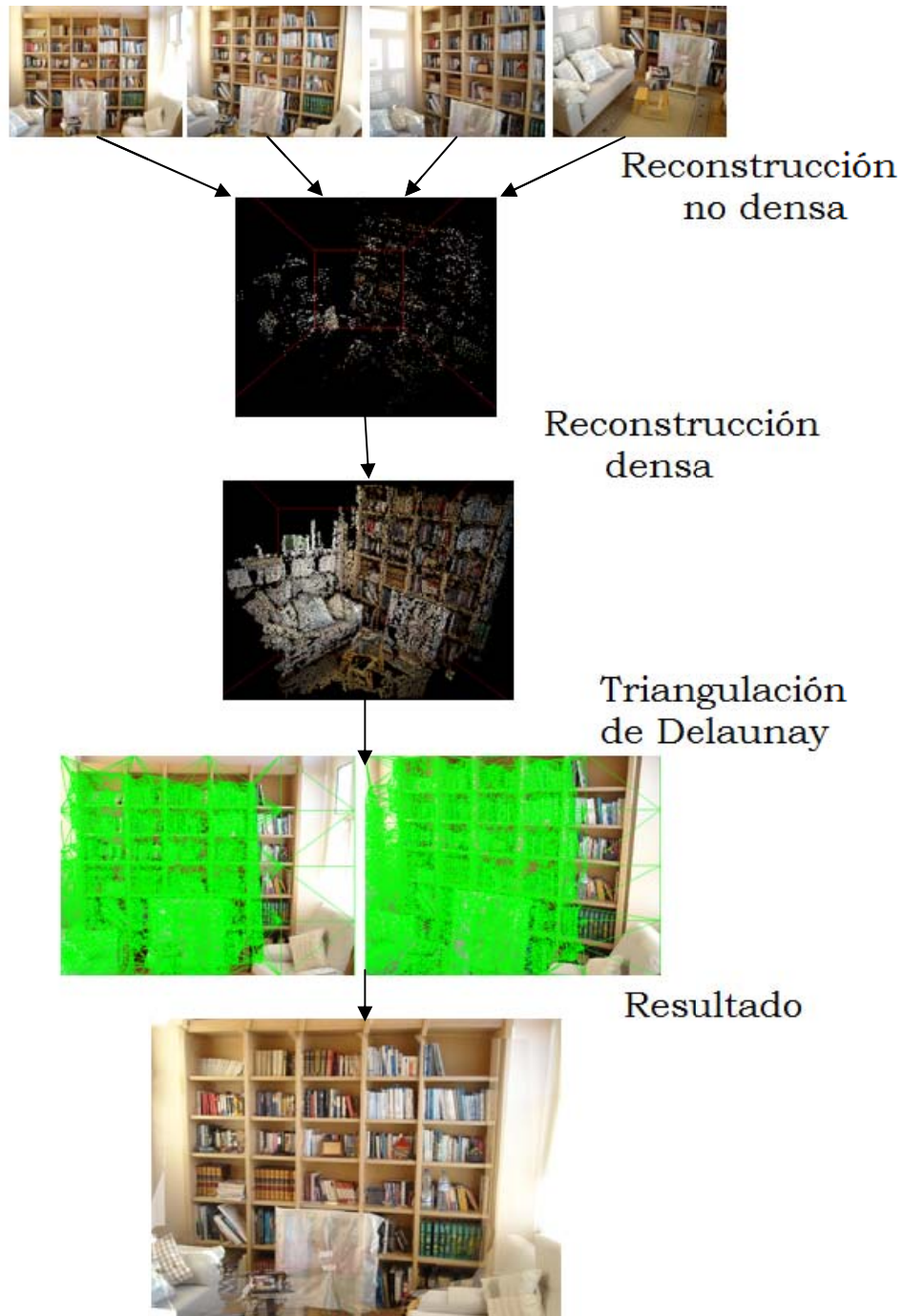


Figura 6.1 Diagrama de flujo

Como mejoras para los métodos anteriormente estudiados, se propone uno alternativo, utilizando una nube de puntos más densa. Para densificarla, a partir de los puntos del 3D, se utiliza el método explicado en el apartado 3.2, donde se obtiene una nube de puntos densa. A partir de aquí se utilizarán estos puntos, y se utilizará el método de síntesis de vista a partir de la triangulación de Delaunay.

Se puede observar en la figura 6.2 que los espurios se han eliminado y la reconstrucción es mucho más parecida a la realidad, a pesar de alguna zona que no es capaz de completar.

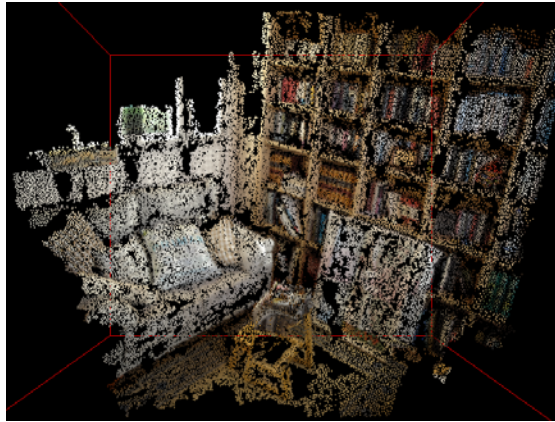


Figura 6.2 Reconstrucción 3D

Al realizar la triangulación de Delaunay, se observa que ahora los triángulos son más pequeños y envuelven la superficie de los objetos mucho mejor que antes (figura 6.3).

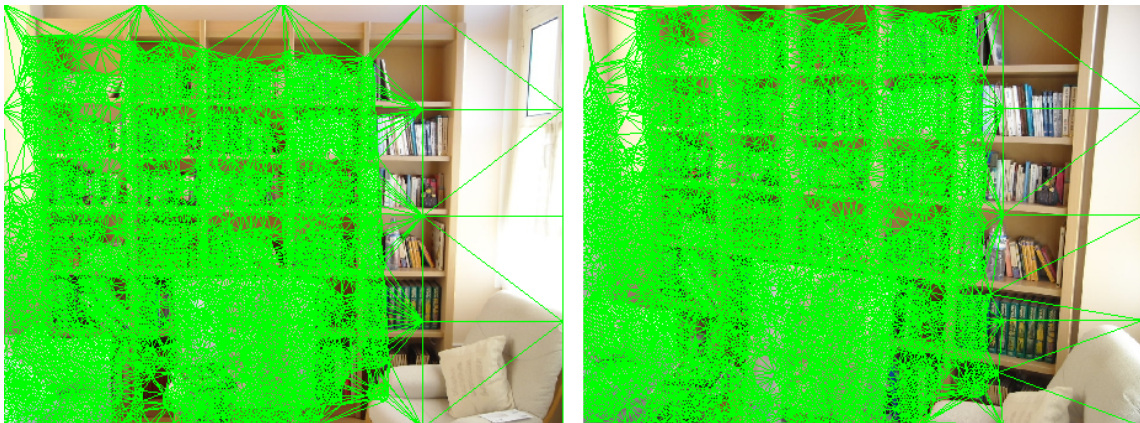


Figura 6.3 Triangulación densa

Volviendo a utilizar la síntesis de vistas mediante triangulación de Delaunay, pero teniendo en cuenta los puntos obtenidos después de densificar la escena, se observa que ahora la transición entre fotografías es muchísimo mejor debido a la gran densidad de puntos que hay, y que las superficies son casi perfectas. En los pequeños huecos en los que no se consigue tener puntos reconstruidos en 3D, es donde se aprecian ciertas distorsiones, aunque por lo general, esas zonas son lo suficientemente pequeñas como para que no afecten a la vista.



Figura 6.4 Triangulación densa comparada con la triangulación no densa

En la figura 6.4, en la parte de la izquierda, se ve como ahora todos los objetos se reconstruyen según la realidad de una forma más acertada, comparado con el del método de la triangulación no densa de la imagen 6.4 en la parte de la derecha, en el que presenta más distorsiones visuales. También se han introducido las mejoras de mostrar la imagen completa fuera de la zona común, para dar aún más sensación de densificación. Además, los puntos que están fuera de la nube de puntos, se proyectan sobre el plano dominante a las dos cámaras, en vez de proyectar un plano para cada imagen, ya que producía más distorsión.

## 7. Resultados experimentales

### 7.1. Habitación de lectura

Comparando los tres métodos estudiados, se pueden comprobar las grandes diferencias que hay entre ellos.

Para ello se parte de dos imágenes iniciales (figura 7.1).



Figura 7.1 Imágenes iniciales

Se tratará de conseguir la imagen representada en la figura 7.2. Esta es una imagen intermedia entre el movimiento de las dos imágenes representadas en la figura 7.1, pero está tomada con cámara de fotos. Al conocer su posición y rotación, se calculará la imagen en ese mismo lugar para los distintos métodos estudiados.



Figura 7.2 Imagen intermedia realizada con cámara de fotos

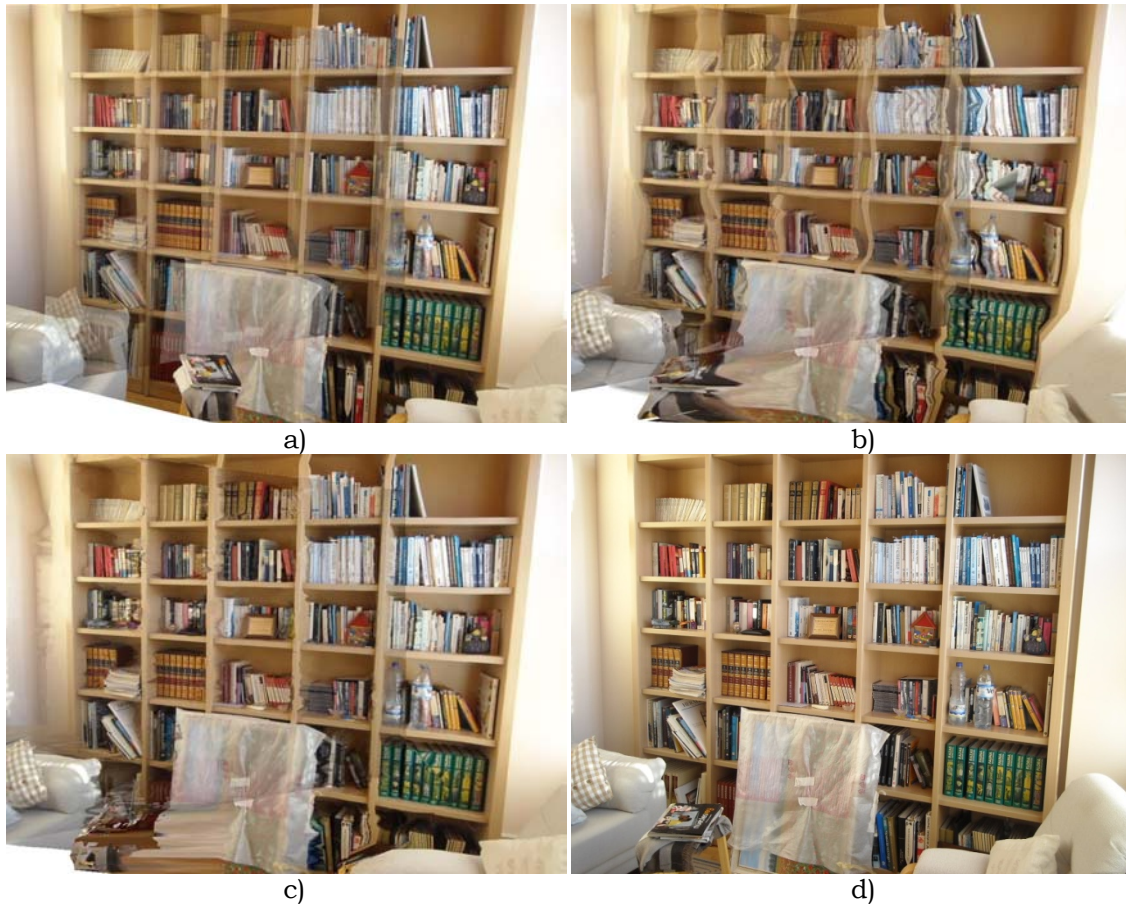


Figura 7.3 a) Método de impostores planares b) Método de triangulación de Delaunay en reconstrucción no densa c) Método de triangulación de Delaunay en reconstrucción densa d) Imagen original

En la figura 7.3 se comparan los tres métodos estudiados. Si bien es cierto que con el método de impostores planares, no se obtienen distorsiones visuales de los objetos contenidos en la fotografía, todo se basa en el plano dominante común de ambas fotografías, por lo que todo lo que resulta contenerse fuera de él sufre grandes cambios de posición. Va desapareciendo de un lugar y apareciendo en otro.

Por otra parte, con el método de la triangulación de Delaunay, no se obtienen esos cambios de situación de los objetos, cada objeto se mantiene en su sitio cambiando de orientación dependiendo del punto de vista. Sin embargo, sí que se pueden observar distorsiones en los lugares donde no hay muchos puntos reconstruidos en el 3D. Con la reconstrucción no densa, las deformaciones son muy grandes, y con la reconstrucción densa, esas deformaciones se vuelven prácticamente imperceptibles en la mayoría de la imagen.

En la figura 7.3 a), se muestra el método del plano dominante, donde se puede ver que el cuadro apoyado en la estantería y el sillón de la izquierda están difuminados debido a que desaparecen en un sitio y aparecen en otro. Aunque se puede ver que no sufren ninguna deformación. En la figura 7.3 c), es una transición del método de triangulación de Delaunay densa. El mismo cuadro y el sofá, se pueden apreciar cómo están reconstruidos en su lugar sin sufrir cambios de posición. Sin embargo, se aprecia que la mesita, al no haberse reconstruido completamente bien en la nube de puntos, sufre alguna distorsión. La estantería también se aprecia mucho mejor en el método de la triangulación de Delaunay densa, ya que aunque está muy próxima al plano dominante, tiene zonas, como las baldas, que sobresalen, y por pequeña que sea la distancia, en el método de impostores planares llega a sufrir deformaciones. En este método, cada punto se muestra según su situación real en el espacio, y la reconstrucción es muy buena.

La mejora es notable, pero el tiempo de cálculo también es mucho mayor. Para lo cual, se decide eliminar ciertos puntos, ya que al realizar la densificación se han obtenido puntos en exceso que ralentizan demasiado el proceso de obtención de imágenes.

Se realiza un muestreo en el que cada cierto número de píxeles se eliminan todos los que se encuentren entre los seleccionados. Haciendo experimentos, se prueba eliminar todos los píxeles que haya en un intervalo cada 8 píxeles, y otro experimento, eliminando los píxeles que hay cada 12 de ellos. Los resultados se muestran en la tabla 7.4.

Método	Tiempo	Tiempo relativo	Error relativo	Puntos utilizados
Planos	11.02 seg	0.021	1.689	4
Delaunay	37.75 seg	0.072	1.258	1005
Delaunay cada 12	80.09 seg	0.153	1.097	3068
Delaunay cada 8	142.88 seg	0.273	1.036	6263
Delaunay denso	523. 58 seg	1	1	23454

Tabla 7.4 Comparación de tiempos y errores

Para calcular el error, se resta la imagen sintética que se ha obtenido con cada método (figura 7.3), con la imagen original (figura 7.2), ya que en el fondo, las imágenes no son más que matrices donde en cada celda, que representa a cada pixel, está el valor del color de 0 a 255 para cada capa R (rojo), G (verde) y B (azul). Se obtiene el valor absoluto de todos los valores, y se suman. El valor de error relativo, se obtiene dividiendo el error de cada método, entre el método que tiene el error absoluto menor, que es el de la triangulación de Delaunay denso.

El tiempo se calcula en comparación con el método que ofrece mejor resultado, el de la triangulación de Delaunay denso, pudiéndose observar el orden de magnitud de diferencia de tiempo comparado con este método.

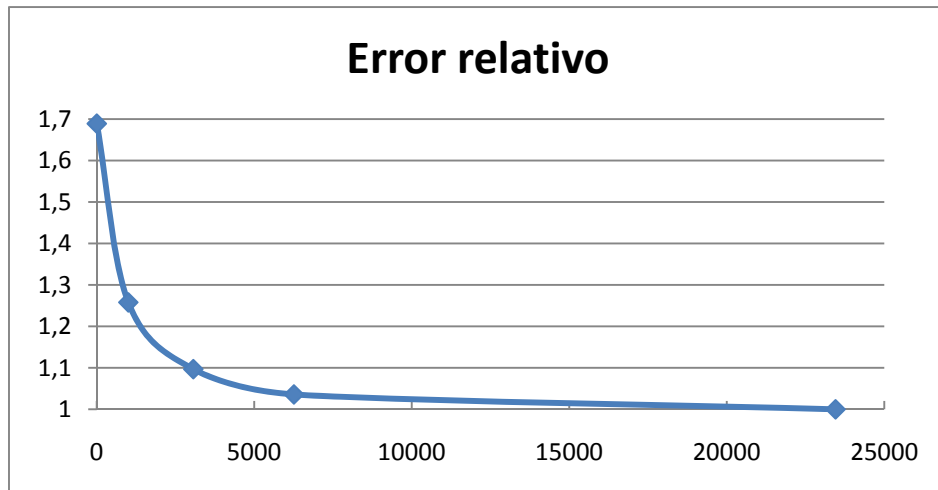


Figura 7.5 Error relativo frente a número de puntos

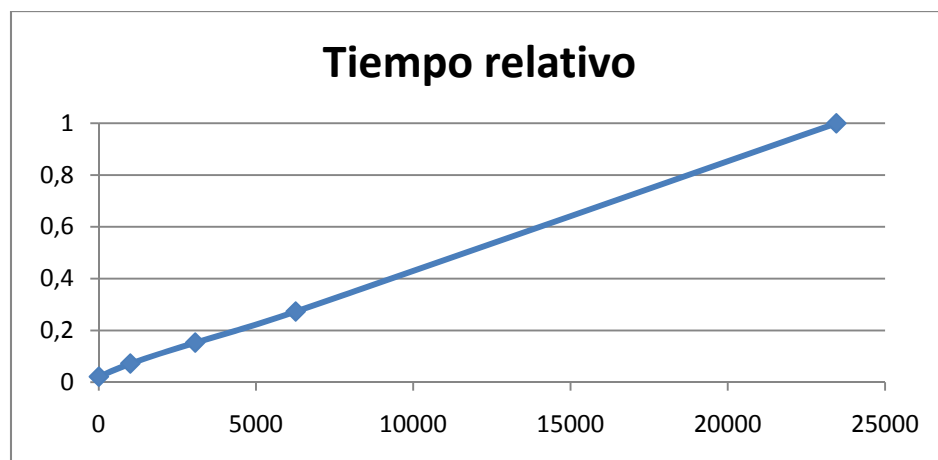


Figura 7.6 Tiempo relativo frente a número de puntos



Como se aprecia en la gráfica de la figura 7.5. Cuantos más puntos se utilizan en la reconstrucción, el error va disminuyendo, hasta alcanzar un valor en el que a partir de él ya casi no se disminuye. Por lo que no es necesario utilizar tantos puntos para obtener un resultado muy similar. La imagen a partir de la que ya casi no disminuye el error, es la de triangulación de Delaunay densa eliminando puntos cada 8 píxeles. Si se eliminan más puntos, el error aumenta considerablemente

Observando la gráfica de la figura 7.6, se aprecia que el tiempo aumenta casi linealmente al aumentar el número de puntos.

Eligiendo el método de eliminación de puntos cada 8 píxeles, frente al método de puntos denso, se consigue una reducción de más de 3 veces en tiempo, obteniendo solamente un 3.6% más de error. En la figura 7.7 se puede comprobar que la diferencia es mínima entre el método de puntos denso, y el de eliminación de puntos cada 8 píxeles.



Figura 7.7 a) Método de triangulación de Delaunay denso b) Método de triangulación de Delaunay denso eliminando puntos cada 8 píxeles

## 7.2. Rana Gustavo

Se realiza el mismo experimento con otro escenario, en el que no existe un plano dominante claro, debido a que la escena tiene objetos irregulares.

Inicialmente se realiza la nube de puntos. (figura 7.8)



Figura 7.8 Nube de puntos

Se parte de dos imágenes iniciales (figura 7.9), y se intenta lograr a partir de ellas, la imagen intermedia de la figura 7.10.

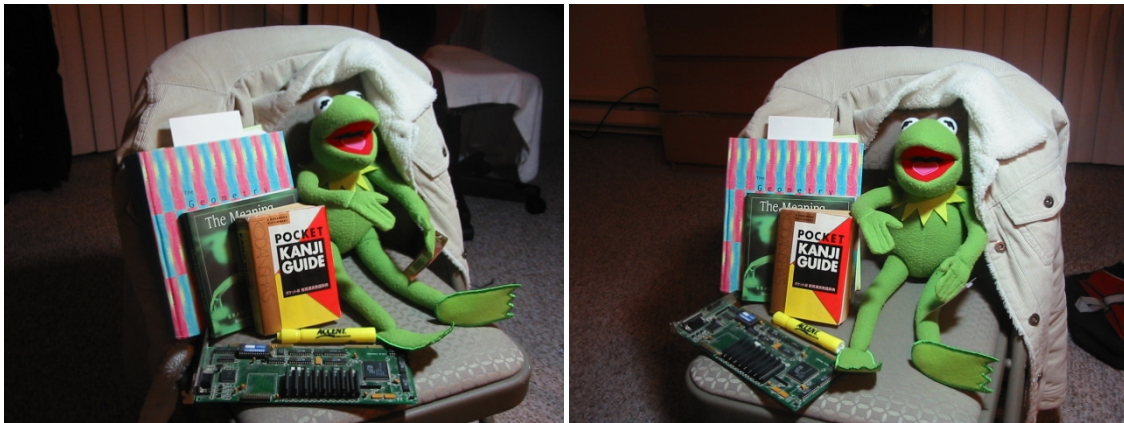


Figura 7.9 Imágenes iniciales



Figura 7.10 Imagen intermedia realizada con cámara de fotos

En la figura 7.11 se obtienen las imágenes de los tres métodos estudiados. Se observa que en la primera imagen al no encontrar un plano dominante claro, ya que hay bastantes objetos, todo está bastante borroso. En los métodos de triangulación de Delaunay, los resultados son mucho mejores, y especialmente en el método de la nube de puntos densa, los resultados son especialmente buenos.



Figura 7.11 a) Método de impostores planares b) Método de triangulación de Delaunay en reconstrucción no densa c) Método de triangulación de Delaunay en reconstrucción densa d) Imagen original

Se vuelve a experimentar a eliminar puntos, para saber si la relación se mantiene al utilizar otro escenario, o era un comportamiento exclusivo de aquel escenario, y se obtienen unos resultados muy similares, que se pueden ver en la tabla 7.12 y las gráficas de la figura 7.13 y 7.14.

Método	Tiempo	Tiempo relativo	Error relativo	Puntos utilizados
Planos	4.26 seg	0.020	1.601	4
Delaunay	12.35 seg	0.058	1.384	251
Delaunay cada 12	51.43 seg	0.243	1.063	1159
Delaunay cada 8	61.43 seg	0.291	1.025	2470
Delaunay denso	211.23 seg	1	1	9698

Tabla 7.12 Comparación de tiempos y errores

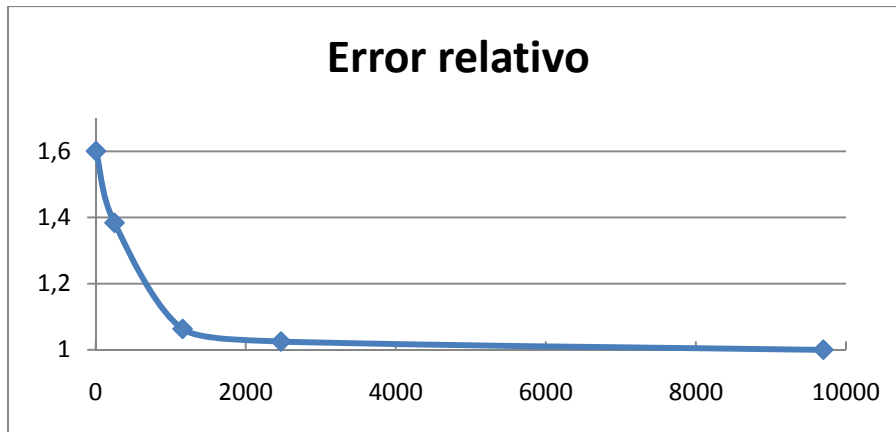


Figura 7.13 Error relativo frente a número de puntos

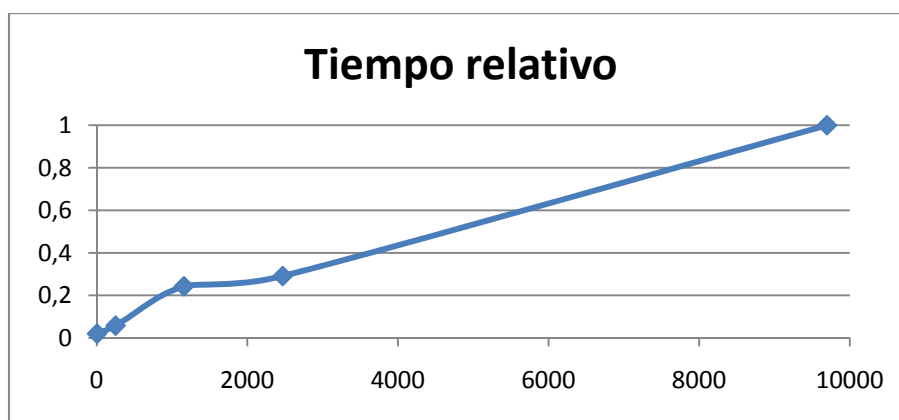


Figura 7.14 Tiempo relativo frente a número de puntos

De estas gráficas se deduce que utilizando el método de Delaunay denso eliminando puntos cada 8 píxeles se reduce el tiempo entre más de 3 veces, obteniendo sólo un error del 2,5%, una conclusión muy similar al escenario anterior.

En la figura 7.15 se observa la diferencia entre el método de Delaunay denso, y el de eliminar puntos cada 8 píxeles, que es casi imperceptible.



Figura 7.15 a) Método de triangulación de Delaunay denso b) Método de triangulación de Delaunay denso eliminando puntos cada 8 píxeles

A continuación se muestran diferentes imágenes sintéticas:



Figura 7.16 Imágenes sintéticas

### 7.3. Cuarto completo

Reconstrucción de una habitación completa.

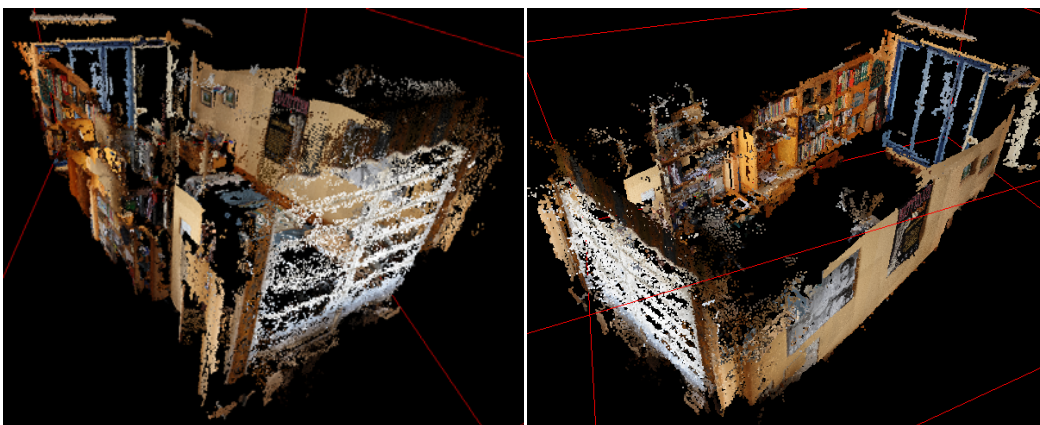


Figura 7.17 Nube de puntos

Se vuelve a repetir el experimento para otra escena diferente, partiendo de las imágenes de la figura 7.18, se pretende conseguir la imagen de la figura 7.19.



Figura 7.18 Imágenes iniciales



Figura 7.19 Imagen intermedia realizada con cámara de fotos

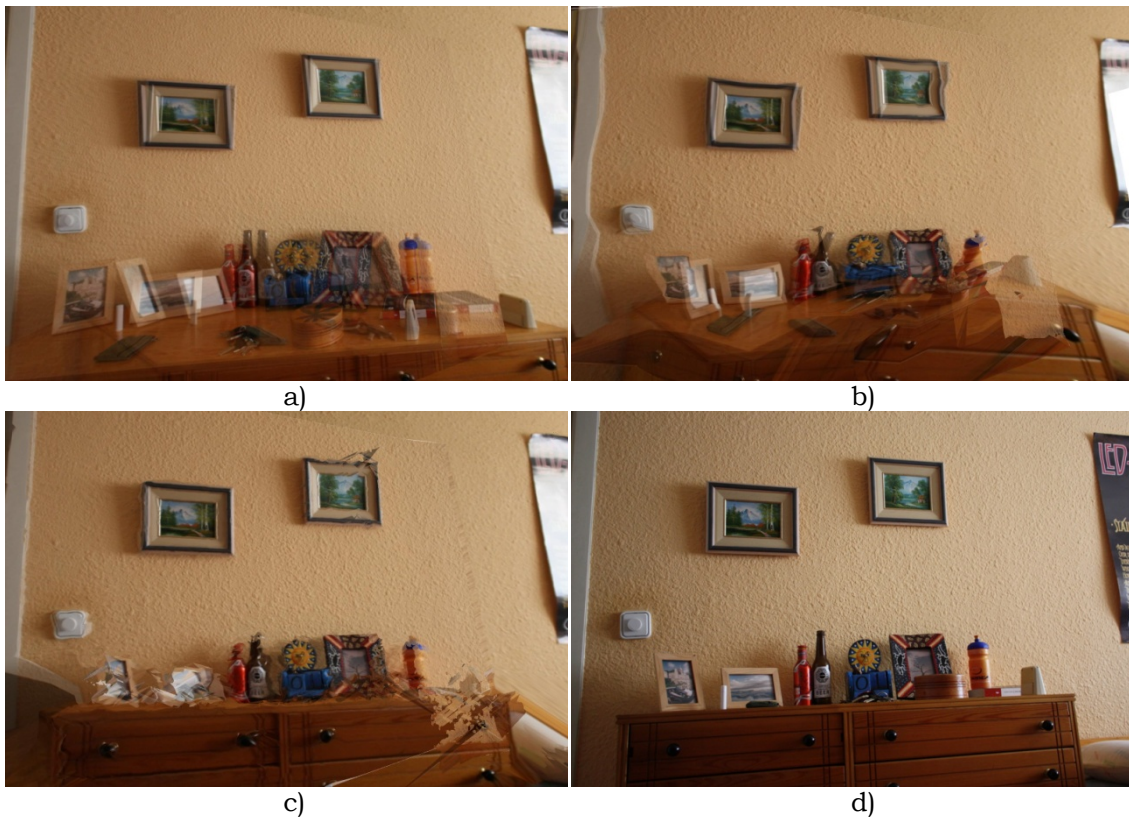


Figura 7.20 a) Método de impostores planares b) Método de triangulación de Delaunay en reconstrucción no densa c) Método de triangulación de Delaunay en reconstrucción densa d) Imagen original

Se observan los resultados para los tres métodos obtenidos en la figura 7.20.

En la tabla 7.21 se obtienen los resultados de la imagen tomada a partir de distintos métodos, incluyendo el experimento de eliminar puntos.

Método	Tiempo	Tiempo relativo	Error relativo	Puntos utilizados
Planos	16.99 seg	0.017	1.66	4
Delaunay	82.76 seg	0.083	1.34	2408
Delaunay cada 12	199.38 seg	0.205	1.123	5247
Delaunay cada 8	280.14 seg	0.281	1.042	10211
Delaunay denso	996.94 seg	1	1	37028

Tabla 7.21 Comparación de tiempos y errores

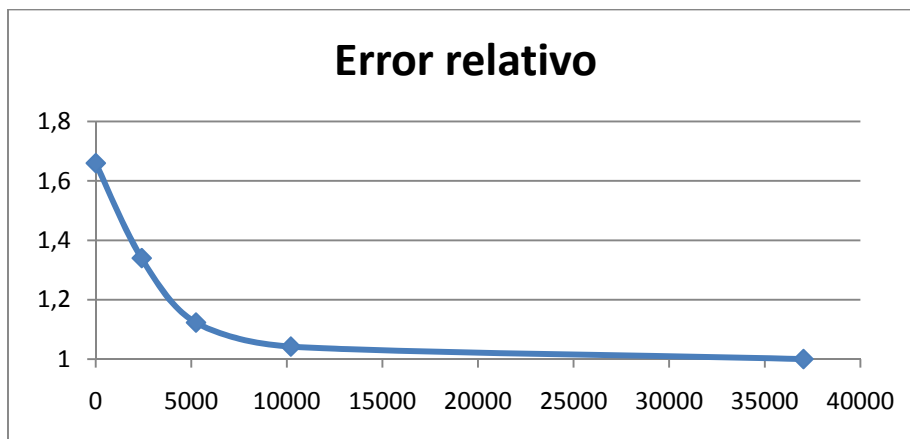


Figura 7.22 Error relativo frente a número de puntos

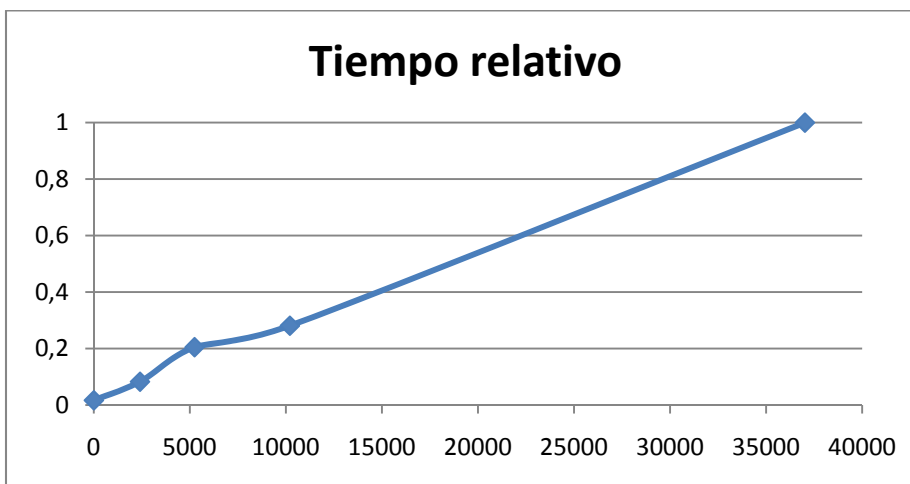


Figura 7.23 Tiempo relativo frente a número de puntos

A partir de los datos de la tabla 7.21 y de las gráficas de las figuras 7.22 y 7.23, se llega a la misma conclusión que en las dos escenas anteriores. Con el método de reconstrucción densa con eliminación de puntos cada 8 píxeles se reduce en más de tres veces el tiempo con sólo un error de 4.2%. Por lo que se puede afirmar que este método es robusto y representa un buen compromiso entre tiempo de cálculo y precisión obtenida en las imágenes, para diferentes escenas.



Figura 7.24 a) Método de triangulación de Delaunay denso b) Método de triangulación de Delaunay denso eliminando puntos cada 8 píxeles

La diferencia entre las imágenes del método de reconstrucción densa y el de eliminación de puntos cada 8 píxeles se puede comprobar en la figura 7.24, donde como ocurría en los escenarios anteriores, es mínima.

A continuación se muestran diferentes imágenes sintéticas de la reconstrucción del cuarto completo:







Figura 7.25 Imágenes sintéticas

#### 7.4. Paseo Independencia

Habiendo demostrado que el método es robusto, se realiza una reconstrucción de una parte del paseo de la Independencia de Zaragoza:

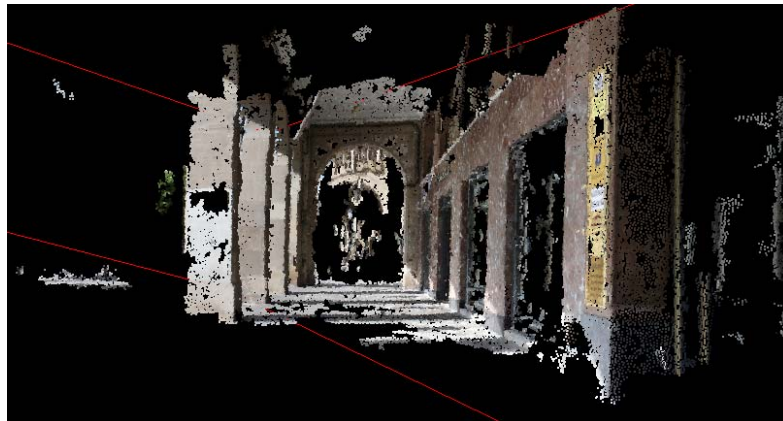


Figura 7.26 Nube de puntos

Se puede comprobar que fuera de los arcos, no hay puntos reconstruidos en el 3D, por lo que en esas zonas se producirán deformaciones. Incluso las columnas no están perfectamente reconstruidas, por lo que en esas zonas se producirán distorsiones.

Con esta escena, se demuestra que estos algoritmos se puede utilizar para mejorar el sistema Google Street View para recorrer calles, ya que las transiciones las hace muy difuminadas, y sin embargo con este método se consigue mayor fluidez en el movimiento.

A continuación se observan diferentes imágenes sintéticas:



Figura 7.27 Imágenes sintéticas

### 7.5. El Pilar

La última escena realizada es la reconstrucción de la fachada principal de la basílica del Pilar en Zaragoza

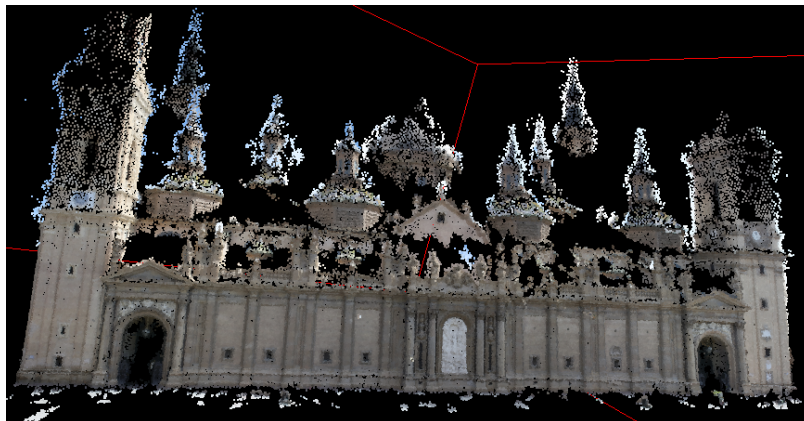


Figura 7.28 Nube de puntos

En esta reconstrucción la fachada principal está perfectamente reconstruida, pero no las torres, en las que faltan bastantes puntos, por lo que esas serán las zonas en las que se observarán deformaciones.

A continuación se observan diferentes imágenes sintéticas:



Figura 7.29 Imágenes sintéticas

## 8. Conclusiones

La reconstrucción 3D es un campo en el que se está trabajando en los últimos años para conseguir obtener modelos densos y fotorrealistas.

En este proyecto se ha conseguido obtener modelos en los que a partir de solamente unas cuantas fotografías se consiguen crear videos con movimiento real, muy similares a los que se podrían obtener con una cámara de video.

Los métodos estudiados, todavía están siendo desarrollados. En este proyecto se ha conseguido darles solución, ofrecer un método alternativo y realizar un estudio comparativo de todos ellos.

Según un estudio del proyecto Photo Tourism [2], el mejor método a utilizar visualmente, es el de síntesis de vistas mediante impostores planares, aunque como se ha podido comprobar, el método alternativo que se ha desarrollado en este proyecto tiene un resultado mucho mejor.

A partir de la reconstrucción en 3D ya conocida de forma no densa y de otra reconstrucción con mayor densidad de puntos, en este proyecto se han implementado desde cero dos métodos de síntesis de vistas mencionados en el estudio del proyecto Photo Tourism de Microsoft [2] con los resultados de la reconstrucción no densa. A partir de ellos se ha trabajado, desarrollando un nuevo método que combina las características positivas de ambos métodos, incluyendo además puntos de una reconstrucción 3D densa. Y con la combinación de todos ellos se ha conseguido realizar síntesis de vistas con mejor calidad visual y de sensación de movimiento fotorrealista a lo largo de una escena.

Como los tiempos de cálculo son muy altos, se ha realizado un estudio de tiempos de cálculo y calidades para los distintos métodos desarrollados, en el cual se consiguen eliminar algunos puntos que no son necesarios, con lo que se consigue un resultado de compromiso en el que se consigue obtener un resultado muy bueno, con tiempos de cálculo menores.

Finalmente se ha comprobado en distintos escenarios obteniéndose resultados muy similares, por lo que se puede asegurar que el método que se ha propuesto en la memoria es robusto, obteniendo buen acabado gráfico sin excesivos tiempos de cálculo.

Las posibles líneas de trabajo futuras para mejorar este algoritmo podrían ser:

- Hacer interpolaciones de posición con curvas en vez de lineales para hacer un movimiento aún más continuo.
- Mejorar la eliminación de puntos del 3D para eliminar más puntos en superficies planas, y eliminar menos en superficies irregulares, para optimizar tiempos de cálculo y mejorar el resultado.
- Mejorar superficies en las que haya huecos importantes en la nube de puntos para eliminar distorsiones grandes en las transiciones.

---

## 9. Bibliografía

[1] Noah Snavely, Steven M. Seitz, Richard Szeliski. *Photo Tourism: Exploring image collections in 3D*. ACM Transactions on Graphics (Proceedings of SIGGRAPH 2006), 2006.

[2] Noah Snavely, Steven M. Seitz, Richard Szeliski. *Modeling the World from Internet Photo Collections*. International Journal of Computer Vision, 2007.

[3] Masayuki Tanimoto. *FTV (Free viewpoint TV) and Creation of Ray-Based Image Engineering*. ECTI Transactions on Electrical Eng., Electronics, and Communications VOL.7, NO.2 August 2009

[4] Taehee Lee. *Robust 3D Street-View Reconstruction using Sky Motion Estimation*. Computer Science Department, University of California

[5] Konstantinos G. Derpanis. *Overview of the RANSAC Algorithm*. Department of Computer Science and Engineering. York University. May 13, 2010

[6] Yannick Morvan, *Pinhole camera model*  
<http://www.epixea.com/research/multi-view-coding-thesisse8.html>

[7] John Harer. *Voronoi Diagrams and Delaunay Triangulations*. Center for Computational Science, Engineering and Medicine. Duke University. July 2006

[8] David G. Lowe. *Distinctive Image Features from Scale-Invariant Keypoints*. Computer Science Department. University of British Columbia. Vancouver, B.C., Canada. January 2004

[9] Yasutaka Furukawa, Jean Ponce. *Accurate, Dense, and Robust Multi-View Stereopsis*. *IEEE Conference on In Computer Vision and Pattern Recognition*, 2007

[10] Programa informático: *Bundler*. Structure from Motion for Unordered Image Collections. Software de Noah Snavely. Última versión: 0.3. del 4 de mayo de 2009

[11] Programa informático *PMVS2*: Patch-based Multi-view Stereo Software

[12] Peter Kovesi, *MATLAB and Octave Functions for Computer Vision and Image Processing*. School of Computer Science & Software Engineering. The University of Western Australia. <http://www.csse.uwa.edu.au/>

[13] José María Martínez Montiel. *Apuntes de la asignatura: "Sistemas de percepción"*, Mayo 2007