Trabajo Fin de Grado

Reconstrucción densa de escenas 3D a partir de imágenes

Autor

Berta Bescós Torcal

Director

Javier Civera Sancho

Escuela de Ingeniería y Arquitectura (EINA)
Curso 2014/2015

TRABAJOS DE FIN DE GRADO / FIN DE MÁSTER

Escuela de Ingeniería y Arquitectura Universidad Zaragoza

DECLARACIÓN DE AUTORÍA Y ORIGINALIDAD

(Este documento debe acompañar al Trabajo Fin de Grado (TFG)/Trabajo Fin de Máster (TFM) cuando sea depositado para su evaluación).

D./Dª. Berta Bescós Torcal,

con nº de DNI 25202254 G en aplicación de lo dispuesto en el art. 14 (Derechos de autor) del Acuerdo de 11 de septiembre de 2014, del Consejo de Gobierno, por el que se aprueba el Reglamento de los TFG y TFM de la Universidad de Zaragoza,

Declaro que el presente Trabajo de Fin de (Grado/Máster) Grado, (Título del Trabajo) Reconstrucción densa de escenas 3D a partir de imágenes,

es de mi autoría y es original, no habiéndose utilizado fuente sin ser citada debidamente.

Zaragoza, a 06/09/15

Fdo: Berta Bescós Torcal

Resumen

En los años 70 del siglo pasado, paralelamente a los avances de la ciencia en el entendimiento de la visión humana, empezó a desarrollarse la visión por computador. El objetivo de este campo de investigación, desde su nacimiento hasta ahora, ha sido conseguir que un ordenador sea capaz de ver, para lo cual es necesario el entendimiento total de las escenas 3D.

Varios algoritmos han sido capaces de obtener las profundidades de puntos característicos de escenas 3D a partir de imágenes, y así tener una ligera idea de cómo son éstas. Este trabajo, sin embargo, busca obtener una reconstrucción densa de una escena para el total entendimiento de ésta. De todas formas, para llevar a cabo una reconstrucción densa a partir de imágenes tomadas por una cámara RGB, es necesario obtener primero los puntos característicos para más tarde aplicar la reconstrucción densa.

Existe un algoritmo básico para llevar a cabo estas reconstrucciones densas. El objetivo de este trabajo ha sido buscar formas de mejora de éste para obtener reconstrucciones más precisas.

Índice

Introducción

Las personas percibimos el mundo tridimensional que nos rodea sin ningún tipo de dificultad aparente. Entre otras cosas, con la vista y a partir de imágenes 2D podemos conocer las profundidades a las que están los objetos.

Los científicos han pasado décadas intentando entender cómo funciona nuestro sistema de visión, consiguiendo dejar claros muchos principios (ilustración 1) pero quedando muchos otros aspectos sin resolver.

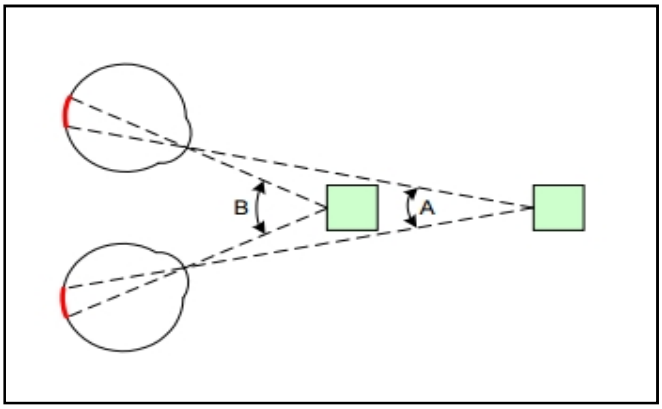

Ilustración 1: Esquema del funcionamiento de la visión humana (Blanes, Jiménez, Puerto, Ñeco, & Reinoso, 2005)

En la ilustración 1 se muestra un esquema del funcionamiento de la visión humana: al mirar un objeto, las rectas que unen dicho objeto con nuestra pupila forman un ángulo determinado. Dependiendo de este ángulo, éstas incidirán en una zona u otra de la córnea. El cerebro interpreta el lugar de incidencia de las rectas en la córnea como la distancia a la que está el objeto observado. Cuanto más separadas incidan las rectas más cerca estará el objeto.

Gracias a muchos de los progresos en el entendimiento de la visión humana, la visión por computador empezó a desarrollarse a principios de los años 70, partiendo de un campo ya desarrollado: el procesamiento de imágenes digitales, (Rosenfeld & Pfaltz, Sequential operations in digital picture processing, 1966) (Rosenfeld & Kak, Digital Picture Processing, 1976). El objetivo a largo plazo de la visión por computador en los años 70 era la reconstrucción de la estructura tridimensional del mundo que nos rodea a partir de imágenes para así poder llegar al entendimiento total de las escenas.

Los primeros pasos para esta reconstrucción fueron la detección de bordes y puntos característicos de las imágenes, así como la reconstrucción de cuerpos simples como cilindros o prismas (Davis, 1975). Más adelante se empezó a trabajar con la reconstrucción de estructuras 3D y escenas en movimiento (Ullman, 1979) (Longuet-Higgins, 1981), campo que aún sigue siendo explorado hoy en día.

En los años 80 se produjo una gran mejora en la detección de bordes y contornos (Canny, 1986), así como el desarrollo de otras técnicas de optimización como el entendimiento de las sombras y texturas (Pentland, 1984).

Todos estos avances, junto con el desarrollo de las tarjetas gráficas e interacción entre ambos, han supuesto un notable desarrollo de la visión por computador en la década de los 90, principalmente para la representación y modelado realista en 3D basado en imágenes (Seitz & Szeliski, 1999).

En la última década se ha seguido viendo este avance conjunto de la visión por computador y los gráficos, y uno de los temas más abordados ha sido el reconocimiento de objetos (Ponce, Hebert, Schmid, & Zisserman). En el año 2003 se desarrolló un algoritmo denominado SIFT que revolucionó el mundo del procesamiento de imágenes: su idea principal es la transformación de la imagen a una representación compuesta de "puntos de interés", que resumen la información de la imagen. (Lowe, 2003)
Otro avance importante de la última década ha sido la reconstrucción densa de imágenes (Richard A. Newcombe), en la que no sólo se reconstruyen puntos característicos de imágenes como hasta ahora, si no que se reconstruye toda la escena en conjunto.

El objetivo global es "que un computador vea", tarea mucho más general que la reconstrucción en 3D de escenas. Sin embargo, la reconstrucción 3D es todavía una de las tareas más relevantes.

A continuación se ilustran algunas de las aplicaciones de la visión por computador, con muchas de las cuales estamos cada vez más acostumbrados a convivir.

I. Hay algoritmos que son capaces de reconstruir un modelo en nube de puntos de una escena a partir de de cientos de fotos de una escena. (Ilustración 2)

Ilustración 2: Reconstrucción en nube de puntos (Structure From Motion) (Snavely, Seitz, & Szeliski, 2006)

II. Mediante algoritmos se pueden unir imágenes superpuestas de un edificio para conseguir una reconstrucción detallada en 3D de éste. Estas imágenes pueden ser tanto propias como encontradas gracias a motores de búsqueda. (Ilustración 3)

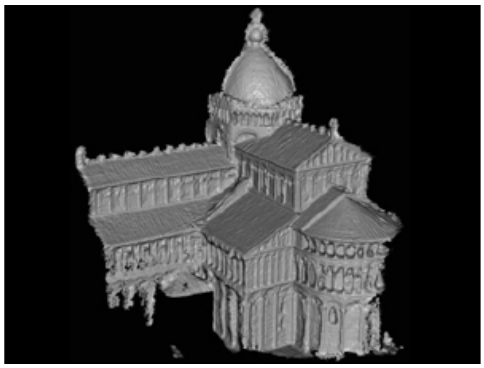

Ilustración 3: Reconstrucción densa en 3D de un monumento (Goesele, Snavely, Curless, Hoppe, & Seitz, 2007)

III. Existen algoritmos capaces de detectar objetos, animales o personas, y hacer un seguimiento del movimiento de éstos en una escena. (Ilustración 4)

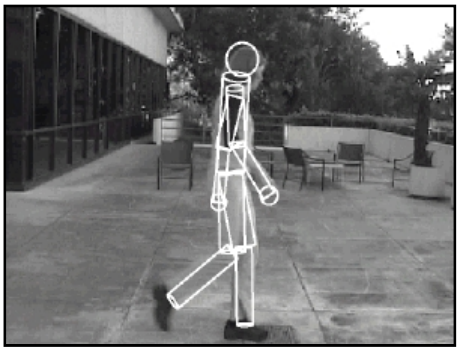

Ilustración 4: Tracking de una persona (Sidenbladh, Black, & Fleet, 2000)

IV. En las redes sociales también hay aplicaciones de la visión por computador. Por ejemplo, Facebook, entre otras, es capaz de localizar y reconocer a las personas que hay en una imagen mediante algoritmos que llevan a cabo detección de rostros, de pelo, de ropa... (Ilustración 5)

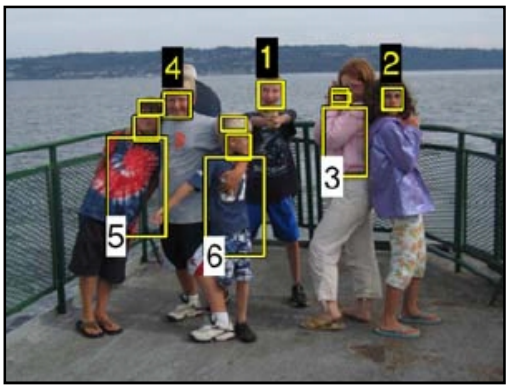

Ilustración 5: Ejemplo de detección de rostros (Sivic, Zitnick, & Szeliski, 2006)

Además de estas aplicaciones de uso doméstico, cabe destacar algunas de uso industrial:

I. Reconocimiento óptico de caracteres ASCII.

II. Seguridad en el sector de la automoción: reacción automática en la conducción de un coche al detectar una persona en el camino de éste.

III. Vigilancia y monitorización del tráfico.

IV. Aplicaciones médicas: realizar estudios de cómo se modifica la morfología del cerebro de una persona conforme su edad va aumentando, registro en imágenes de órganos antes, durante y después de una operación...

Objetivos

El objetivo de este trabajo de fin de grado es la reconstrucción densa de escenas 3D a partir de imágenes, así como el análisis de algunos de los parámetros de los algoritmos estándar. Esta reconstrucción se hace buscando las correspondencias de todos los píxeles de una imagen de referencia en otras imágenes. Dichas correspondencias se llevan a cabo usando la geometría epipolar de las cámaras, y siendo el color de cada píxel su descriptor.

Los principales puntos analizados son los siguientes:

- Utilización de parches de píxeles.
  Hasta ahora para emparejar un píxel con otro sólo se utilizaba la geometría epipolar y el color de los mismos. Con la introducción de parches se compara no sólo el color del píxel en cuestión, si no que también los de su alrededor.
- Introducción de umbrales.
  Se introduce un umbral máximo que sólo permite reconstruir aquellos píxeles cuya diferencia de color con su emparejado sea menor que dicho umbral.
- Utilización de varias imágenes.
  Una imagen de referencia y una imagen de apoyo no son suficientes para realizar una correcta reconstrucción. Buscar el número óptimo de imágenes de apoyo será otro de los objetivos del trabajo.
- Análisis de un método de suavizado.
  Este método se basa en la idea de que si una parte de la imagen tiene el mismo color, su profundidad debería ser continua, modificando así las profundidades obtenidas inicialmente.

Otro objetivo del trabajo es conseguir, a partir de cualquier conjunto de imágenes, la posición (rotación y traslación) en la que han sido tomadas éstas, así como la calibración y distorsión de la cámara utilizada, para entonces poder aplicar la reconstrucción densa.

Modelo Geométrico

La reconstrucción tridimensional de puntos a partir de imágenes 2D está basada en la idea de triangulación (ilustración 6): conocidas las posiciones de los centros de las cámaras (c_0 y c_1) respecto de un sistema de coordenadas W , así como sus matrices de rotación (R_0 y R_1), se pueden reconstruir todos los puntos comunes a las dos fotografías. Sabiendo que el píxel x_0 en la imagen 0 corresponde al píxel x_1 en la imagen 1, uniendo el centro geométrico de las cámaras con sus correspondientes píxeles se crean dos rectas v_0 y v_1 cuya intersección es el punto real p al que corresponden ambos píxeles.

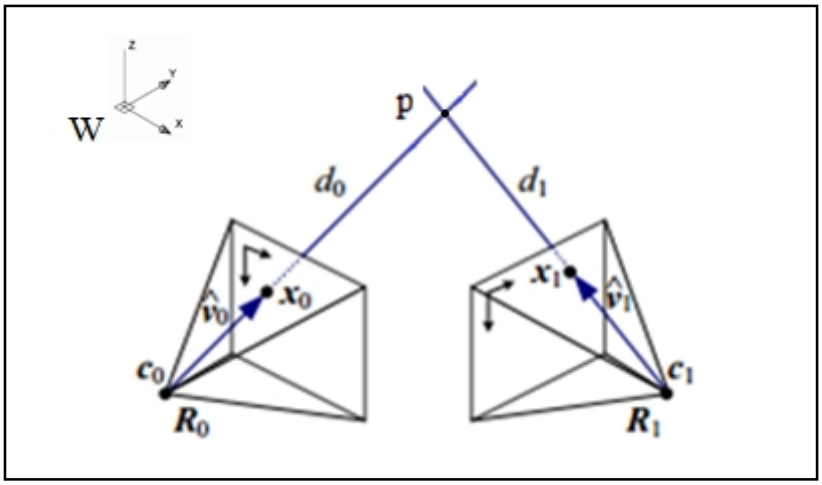

Ilustración 6: Reconstrucción a partir de triangulación (Szeliski, 2010, p. 346)

Esta idea imita a la visión humana (ilustración 1). Si sólo viéramos con un ojo no seríamos capaces de calcular las distancias y la información geométrica de los objetos observados. Es por eso que para reconstruir escenas a partir de imágenes necesitamos como mínimo dos cámaras o varias imágenes tomadas por una misma cámara.

En el caso a estudiar no se sabe qué píxel corresponde con cuál, entonces el procedimiento a seguir es unir un píxel de la imagen 0 x_0 con su correspondiente centro de cámara. Ahora sobre esta recta v_0 se sitúa el supuesto punto real p a una distancia cualquiera d_0 y se une con el centro de la otra cámara, dando como intersección el píxel x_1 en la imagen 1.

Matemáticamente, esta proyección de un píxel de la imagen de referencia sobre otra imagen se hace mediante el procedimiento siguiente:

$x_0 = (u \quad v \quad 1)^T$ es el píxel de la imagen 0 que se quiere proyectar en la imagen 1. Si se toma como origen de coordenadas el centro de esta cámara c_0, el vector posición del píxel quedaría definido como $x_0^{C0} = K^{-1} \cdot x_0$, donde K es la matriz de calibración de la cámara.

$$K = \begin{pmatrix} f & 0 & c_x \\ 0 & f & c_y \\ 0 & 0 & 1 \end{pmatrix}$$

f = distancia focal de la cámara medida en píxeles.

c_x, c_y = punto principal de la imagen medido en píxeles.

Convirtiendo este vector a coordenadas homogéneas con una distancia inicial cualquiera d_0 se obtiene $x_0^{C0\_h} = \begin{pmatrix} x_0^{C0} \\ \frac{1}{d_0} \end{pmatrix}$, que transformado al sistema de referencia global W resulta el punto $p = x_0^W = T^{C0} \cdot x_0^{C0\_h}$, donde la matriz T contiene información sobre la posición de la cámara c_0 respecto a la referencia W.

$$T^{C0} = \begin{pmatrix} R_{wc} & t_w \\ 0^T & 1 \end{pmatrix}$$

R_{wc} = matriz de rotación de la cámara respecto del sistema de coordenadas W [3, 3]

t_w = matriz de traslación de la cámara respecto del sistema de coordenadas W [3, 1]

Hasta ahora se ha proyectado el píxel x_0 en el sistema de referencia W, y ahora hay que proyectar este punto que se encuentra en el sistema de referencia W en el píxel equivalente x_1 en la cámara c_1.

1. $x_1^{C1\_h} = (T^{C1})^{-1} \cdot x_0^W$
2. $x_1^{C1} = K \cdot \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \cdot x_1^{C1\_h}$
3. Siendo $x_1^{C1} = (u' \quad v' \quad w')^T$, $x_1 = (\frac{u'}{w'} \quad \frac{v'}{w'})^T$.

Variando la distancia d_0 a la que está situado el punto p se obtiene una sucesión de píxeles de la imagen 1 que podrían corresponder al píxel elegido en la imagen 0. Estos píxeles forman una recta llamada línea epipolar. Este procedimiento se ve en la ilustración 7 en forma de esquema, y en la ilustración 8 se ve este método aplicado a dos imágenes reales: en la imagen de la izquierda se ha elegido un píxel cualquiera, dibujado en rojo, el cual tiene su línea epipolar dibujado en azul en la imagen de la derecha.

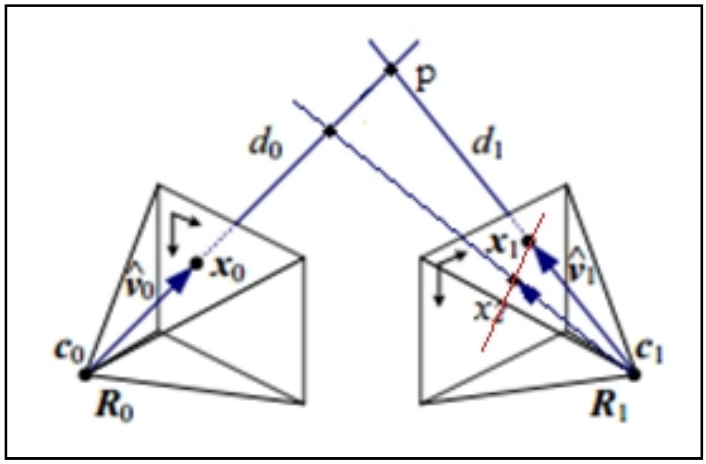

Ilustración 7: Obtención de la línea epipolar (Szeliski, 2010, p. 346)

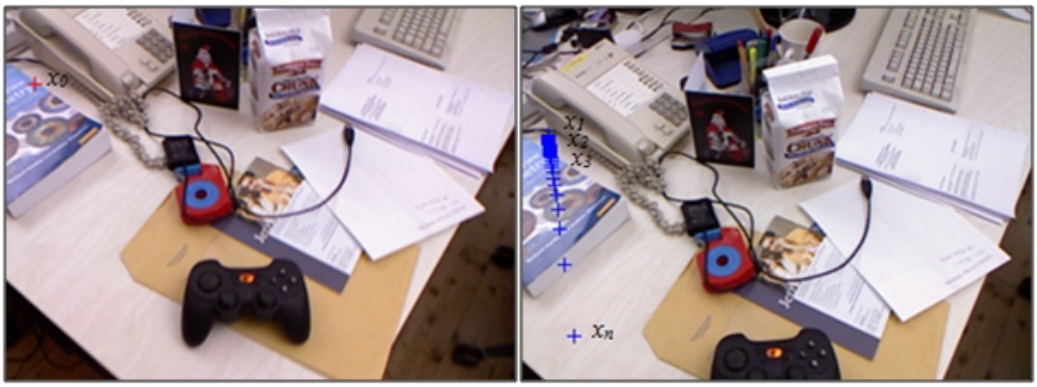

Ilustración 8: Línea epipolar (3b) del píxel de la figura 3a

Una vez obtenida la línea epipolar correspondiente al píxel x_0 hay que elegir uno de entre los n píxeles x_1, x_2, ..., x_n que la forman. Para esto es necesario comparar el color del píxel x_0 con los colores de los píxeles contenidos en la epipolar. En el caso de ser en escala de grises el color estará dado por un número entero de 8 bits (de 0 a 255), y en el caso de ser en color estará dado por tres números enteros de 8 bits correspondientes a las cantidades respectivas de rojo, verde y azul (escala RGB) que hay en el píxel. El píxel de la epipolar que tenga un color más parecido al del píxel x_0 será el que tenga un error fotométrico menor, y será el equivalente en la imagen 1, obteniéndose así la profundidad de cada píxel común a las imágenes 0 y 1.

En el caso de utilizar más de dos imágenes para la reconstrucción de una escena, el procedimiento es el siguiente (ilustración 9): en la imagen a reconstruir, imagen 0, se traza la recta que une el píxel a estudiar x_0 con el centro de la cámara c_0, y se elige una profundidad arbitraria d_0. Este punto se une con los centros de todas las cámaras (c_1, c_2 ...) que se tienen para comparar, intersectando así en los píxeles correspondientes a esa distancia d_0 (x_1, x_2 ...). Entre estos píxeles, el que tenga un menor error fotométrico con el píxel a estudiar x_0 será

con el que se reconstruirá la profundidad. Esto se hace para diferentes profundidades, y finalmente la profundidad para la cual el error fotométrico sea mínimo será la profundidad del píxel en cuestión.

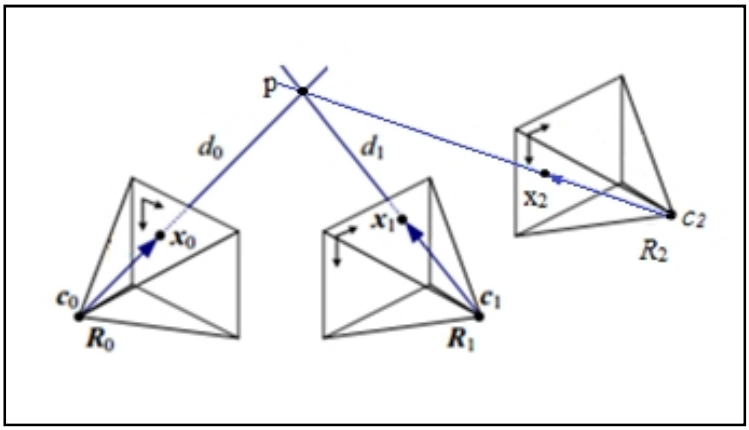

Ilustración 9: Reconstrucción de la imagen 0 a partir de dos imágenes (Szeliski, 2010)

A partir de los conceptos básicos anteriores se han desarrollado dos modelos más completos para la reconstrucción de espacios 3D: *Sparse Mapping* y *Dense Mapping*.

Sparse Mapping

Para comenzar cualquier reconstrucción es necesario conocer la posición en la que se han tomado todas las imágenes a utilizar. Obtener la posición de las cámaras implica precisar los parámetros de rotación y traslación de éstas respecto de un origen de coordenadas.

El método por el cual se obtienen estos parámetros es mediante el emparejamiento de puntos por geometría, problema que se reduce a emparejar los puntos salientes.

En primer lugar se obtienen los puntos salientes de cada imagen mediante el algoritmo SIFT (Lowe, 2003). Estos puntos salientes son zonas con alto gradiente de color, como vértices o bordes de objetos.

A continuación se realiza el emparejamiento entre los puntos salientes obtenidos de todas las imágenes utilizadas. Esto se hace comparando los gradientes de color en todas las direcciones de cada punto saliente. El vector que contiene estos gradientes de color se llama descriptor.

Cuando un píxel es emparejado correctamente cumple la siguiente ecuación: $x_1^T \cdot F \cdot x_2 = 0$, donde x_1 es un píxel en coordenadas homogéneas en la imagen de referencia, x_2 es la correspondencia al píxel x_1 en otra imagen diferente, y F es la llamada matriz fundamental, que es una matriz de dimensiones 3 x 3 que contiene información sobre la calibración de las dos cámaras, y la rotación y traslación entre ellas. Cuando las calibraciones de las cámaras son conocidas se utiliza de forma dual la matriz esencial E, que también cumple la ecuación $x_1^T \cdot E \cdot x_2 = 0$.

A partir de estas matrices se puede obtener la recta epipolar en las imagen de apoyo correspondiente al píxel x_1 en la imagen de referencia con la fórmula $l = F \cdot x_1$ o $l = E \cdot x_1$ en cada caso.

En el caso en el que las calibraciones de las cámaras son conocidas, como es nuestro caso, para hacer el emparejamiento entre dos imágenes se tienen como incógnitas tres orientaciones y tres traslaciones. Si con el método SIFT se consigue emparejar un punto saliente se tendrán las anteriores seis incógnitas más la profundidad de este punto, y como ecuaciones las dos que añade este punto referentes al eje *X* y al eje *Y*. Como no se tienen suficientes ecuaciones para obtener las siete incógnitas, se van emparejando más puntos salientes, hasta que con cinco emparejamientos se tienen 10 ecuaciones y 11 incógnitas. Como nunca se pueden obtener las distancias reales, si no que se obtienen las distancias multiplicadas por un factor de escala, siempre va a haber una incógnita en la reconstrucción, por lo que el mínimo número de emparejamientos para cada par de imágenes tiene que ser cinco.

Además, se suele utilizar un algoritmo llamado RANSAC, que utiliza más de cinco emparejamientos por imagen, optimizando este proceso para que no haya ningún emparejamiento erróneo.

A partir del procedimiento anterior se realiza, mediante un proceso iterativo, la optimización de la rotación y traslación de cada imagen, así como de la profundidad de los puntos emparejados. Este problema se denomina *Bundle Adjustment*, y es un problema iterativo de mínimos cuadrados no lineales. (Snavely, Seitz, & Szeliski, 2007)

La ecuación que formula la anterior optimización es la que se muestra a continuación:

$$\left\{ \hat{R}_{ci} \quad \hat{t}_{ci} \quad \hat{x}_j \right\} = \arg\min\left(\sum_{i,j} (z_j^{ci} - h(x_j, R_{ci}, t_{ci}))\right),$$

en la cual se obtienen las rotaciones, traslaciones y profundidades óptimas $\hat{R}_{ci}$, $\hat{t}_{ci}$, y $\hat{x}_j$ de cada cámara i y cada punto j respectivamente, haciendo mínima la diferencia entre la profundidad del punto obtenida mediante el emparejamiento SIFT y la profundidad obtenida mediante la proyección del píxel en el sistema de referencia mundo.

Con este método se obtienen reconstrucciones en forma de nube de puntos, como la que se ve en la ilustración 10.

Ilustración 10: Reconstrucción en mapa de puntos

Con este tipo de reconstrucciones se calcula con alta precisión la posición de las cámaras y de los puntos característicos de las imágenes.

Para obtener la reconstrucción de todos los puntos hay que recurrir a otras técnicas como Dense Tracking and Mapping.

Dense Tracking and Mapping (DTAM)

La técnica DTAM también conlleva el seguimiento de las cámaras o *tracking*. El *tracking* consiste en la estimación de la posición de las cámaras a partir del modelo 3D y de la posición de las cámaras previas.

Este proyecto se centra en la técnica de *Dense Mapping* pero no en el *Tracking*.

Una vez ya obtenidas las posiciones de las cámaras este modelo persigue minimizar una función de coste volumétrico $C_r(u,d)$ donde u es un vector con las coordenadas del píxel en cuestión $u=(x_i,y_i)^T$, y d es la proyección inversa de todas las posibles profundidades de este píxel.

Dado un conjunto de imágenes $\ell(r)$, esta función de coste $C_r(u,d)$ se define como la media del error fotométrico. Se obtiene proyectando un píxel en el volumen fotografiado para cada imagen tomada, y sumando la norma L_1 de cada error fotométrico, como se aprecia en la fórmula $C_r(u,d)=\frac{1}{|\ell(r)|}\cdot\sum_{m\in\ell(r)}\|\rho_r(I_m,u,d)\|_1$. (Richard A. Newcombe)

El error fotométrico de cada imagen se calcula mediante la siguiente fórmula: $\rho_r(I_m,u,d)=I_r(u)-I_m(\pi(KT_{mr}\pi^{-1}(u,d)))$, donde I_r es la imagen de referencia, e I_m es cada una de las imágenes utilizadas para la reconstrucción de I_r. Las constantes K y T_{mr} son respectivamente las siguientes matrices, que ya han sido explicadas anteriormente:

$K=\begin{pmatrix} f & 0 & c_x \\ 0 & f & c_y \\ 0 & 0 & 1 \end{pmatrix}$ y $T_{mr}=\begin{pmatrix} R_{wc} & t_w \\ 0^T & 1 \end{pmatrix}$, donde

f = distancia focal de la cámara en píxeles

c_x, c_y = posición del centro de la cámara en píxeles

R_{wc} = matriz de rotación de la cámara respecto de un sistema de coordenadas W [3 x 3]

t_w = matriz de traslación de la cámara respecto de un sistema de coordenadas W [3 x 1]

La función π^{-1} se define como $\pi^{-1}(u,d)=\frac{1}{d}K^{-1}\dot{u}$, donde $\dot{u}$ es el vector homogéneo de u $\dot{u}=(x_i,y_i,1)^T$, y la función π transforma a coordenadas euclídeas un vector (si $x_c=(x,y,z)^T$, $\pi(x_c)=(x/z,y/z)^T$).

La función ρ será mínima cuando el píxel que se tome en la imagen I_m sea el equivalente al de la imagen I_r de referencia.

Calculando el mínimo de la función de coste C_r para diferentes profundidades, y sacando el mínimo de ésta, se obtendrá la profundidad a la que está el píxel u. (Richard A. Newcombe)

Los resultados de este modelo son correctos cuando se trata de detectar bordes o zonas con altas texturas, pero a la hora de reconstruir píxeles de zonas de baja textura los resultados pueden dar lugar a error. Se asume que estas zonas deberían cambiar su profundidad de forma muy suave, por lo que va a introducirse un nuevo término regularizador para que la reconstrucción de estas zonas sea igualmente suave.

Hasta ahora la función de la energía del problema era igual a la del coste. De ahora en adelante ambas difieren al introducir este último término regularizador, resultando ésta en la siguiente fórmula: $E_\xi = \int_\Omega \{g(u) \cdot \|\nabla \xi(u)\|_\in + \lambda \cdot C(u, \xi(u))\} du$ donde $g(u) = e^{-\alpha \cdot \|\nabla I_r(u)\|_2^\beta}$, $\xi(u)$ es el mapa de profundidades inversas del píxel u, y el parámetro λ es el peso relativo del coste frente al término regularizador.

La norma de la función de energía es definida como norma de Huber, donde ϵ toma valores en torno a e^{-4}. La norma de Huber es una función compuesta por otras dos funciones convexas:

$$\|x\|_\epsilon = \begin{cases} \dfrac{\|x\|_2^2}{2\epsilon} & if\ \|x\|_2 \le \epsilon \\ \|x\|_1 - \dfrac{\epsilon}{2} & otherwise \end{cases}$$

Como ambos términos de la función de energía dependen de $\xi(u)$, para facilitar que se encuentre el mínimo de esta función se añade un término con una variable auxiliar α.

$$E_{\xi,\alpha} = \int_\Omega \{g(u) \cdot \|\nabla \xi(u)\|_\in + \frac{1}{2\theta}(\xi(u) - \alpha(u))^2 + \lambda \cdot C(u, \alpha(u))\} du$$

Con esta expresión, si $\xi \to \alpha$ se tendría la misma función de energía que antes, pero con los términos de coste y regularizador desacoplados. La variable θ es el peso relativo de este término. Esta expresión puede ahora ser optimizada de manera sencilla mediante métodos iterativos. (Anexo I) (Richard A. Newcombe)

Resultados Experimentales

Las imágenes utilizadas para llevar a cabo en le primera parte del experimento, cuyo tamaño es 480 x 640 píxeles, han sido tomadas de un dataset público (Sturm, 2009).

Para medir la validez de este método se ha calculado el error de cada píxel, comparando la profundidad hallada por el programa con las profundidades reales obtenidas previamente con una cámara Kinect, disponibles también en el dataset (Sturm, 2009).

Las cámaras Kinect cuentan con una cámara RGB, así como con un emisor y receptor de patrones infrarrojos para calcular las profundidades (ilustración 11). El emisor de infrarrojos proyecta sobre la imagen a reconstruir un patrón, y siguiendo la misma idea de reconstrucción de puntos por triangulación se obtienen las distancias de cada punto.

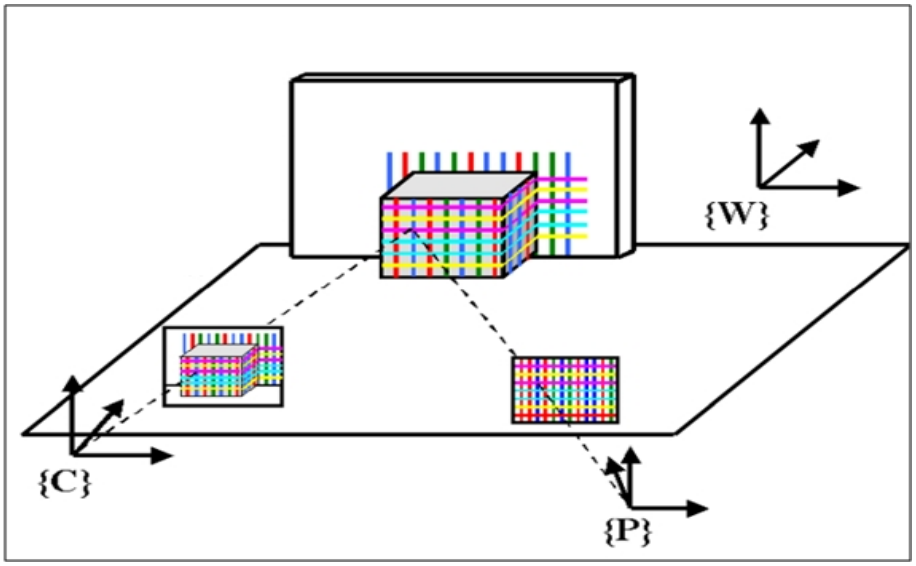

Ilustración 11: Funcionamiento de una cámara Kinect (Sturm, 2009)

Su rango de trabajo suele estar entre los 50 cm, distancia para la cual se obtiene un error de décimas de mm, y los 5 metros, con un error de unos pocos cm. (Guerig, 2012)

A pesar de que en todos los documentos y bibliografía utilizada para este trabajo el error se expresa siempre en unidades de profundidad, se ha decidido estimar el error como porcentaje, calculado como $error = \frac{\left| z_{real} - z_{experimental} \right|}{z_{real}} \times 100$, siendo z la profundidad, dando en algunos casos concretos el error en unidades de distancia.

Esta decisión se debe a que el mismo error en unidades de distancia puede significar un error despreciable en algunos casos (20 cm de error en reconstrucciones de zonas que están a 20 m), o un error significativo en otros (20 cm de error en reconstrucciones de zonas que están a 50 cm). Sin embargo, si se utiliza el error en tanto por ciento no se da lugar a este tipo de ambigüedades.

Además, dentro de una misma reconstrucción, si se dice que el error medio de ésta es de 20 cm se podría entender que todos los píxeles a reconstruir, tanto si están cerca como lejos, tienen un error de aproximadamente 20 cm, lo cual no es cierto porque varía mucho de unos píxeles a otros dependiendo de la distancia a la que estén. Sin embargo, si el error está dado en porcentaje, es más probable que éste sea un valor fijo o más parecido para todos los píxeles de la imagen a reconstruir.

De esta forma, al utilizar el error en porcentaje éste se vuelve lineal con la profundidad, mientras que si se utiliza el error en unidades de distancia variaría de forma cuadrática con la profundidad. (Anexo II).

Se ha llevado a cabo este experimento siguiendo la técnica de *Dense Mapping*, pero sin introducir el término regularizador, ejecutando el código realizado en MATLAB (Anexo III) sobre un gran número de pares de imágenes.

Se han utilizado tres librerías de imágenes diferentes para contrastar los resultados. En cada librería se han hecho las pruebas sobre 7 pares de imágenes diferentes, y los resultados finales son un promedio de los errores obtenidos de éstos. Algunas de las imágenes utilizadas para la reconstrucción y que cumplen las anteriores observaciones se muestran en las ilustraciones 12, 13 y 14.

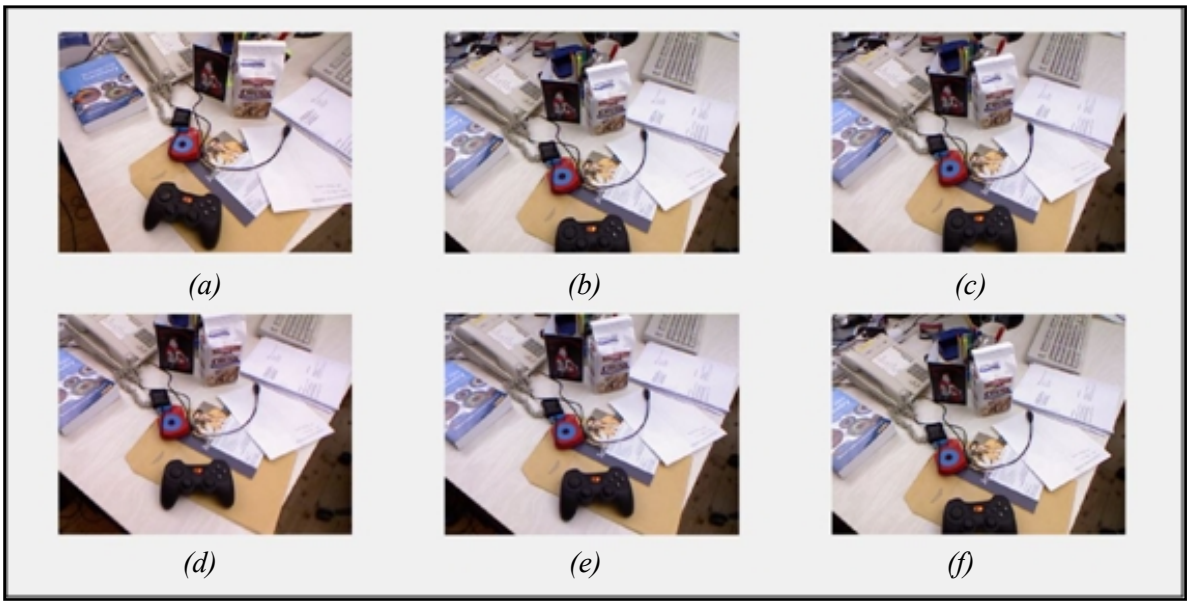

Ilustración 12: Algunas de las imágenes de la librería 1

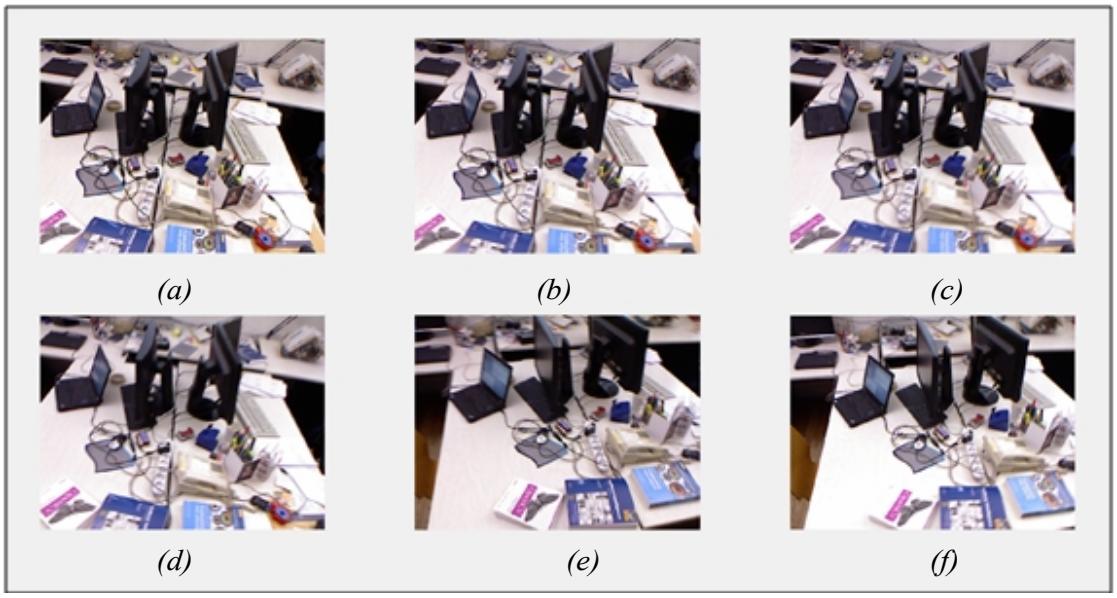

Ilustración 13: Algunas de las imágenes de la librería 2

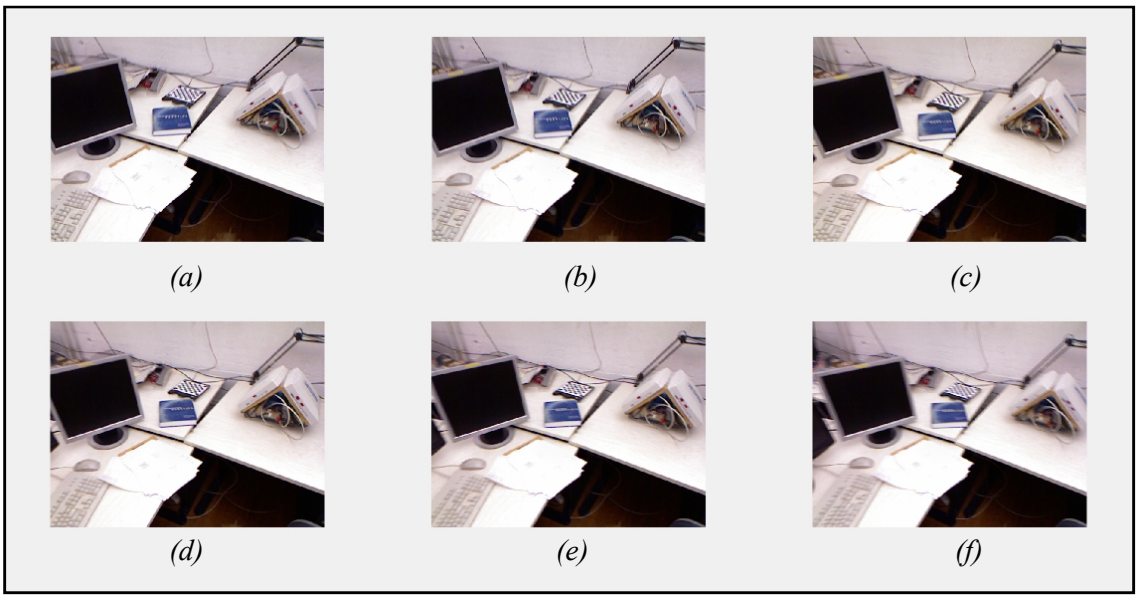

Ilustración 14: Algunas de las imágenes de la librería 3

La distribución del error que siguen todas las reconstrucciones es muy similar (ilustración 15). Hay un gran número de píxeles con error casi nulo, pero prácticamente ninguno con error cero. Esto último resulta lógico ya que al trabajar con píxeles se está discretizando la imagen, y por tanto, cometiendo un error al emparejar. Al aumentar el error, el número de píxeles va disminuyendo exponencialmente hasta llegar a cantidades muy pequeñas de píxeles con errores muy grandes (1 píxel con un 200 % de error).

Estos errores grandes que aparecen en todas las reconstrucciones se suelen deber a zonas en las que el color es muy homogéneo y entonces es muy fácil confundir píxeles, y también a zonas con altos gradientes de color que se repiten en el espacio (bordes de una mesa, lamas de una ventana, teclas de un teclado...).

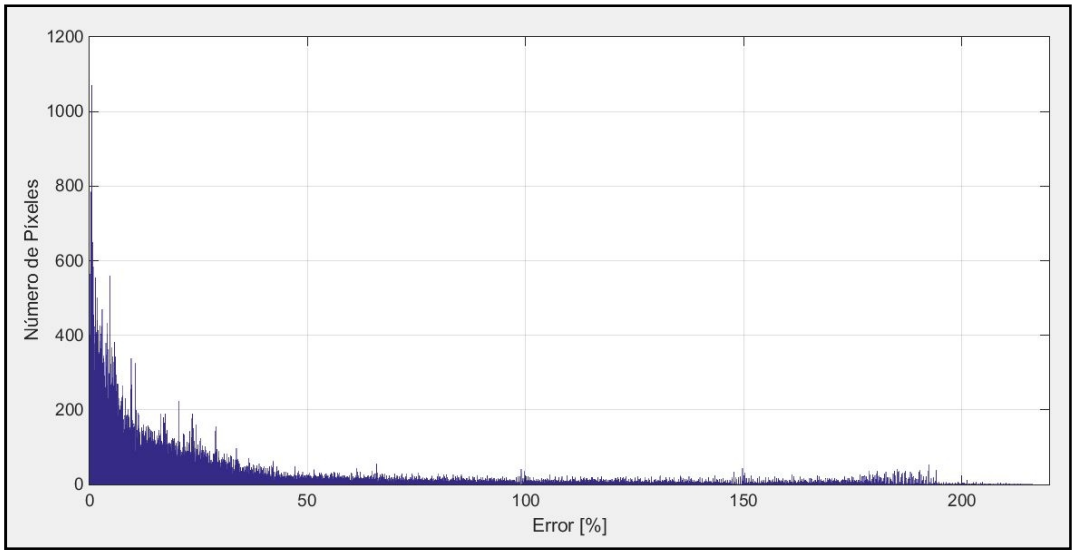

Ilustración 15: Histograma del error de una reconstrucción de la librería 1

Todas las reconstrucciones llevadas a cabo con las imágenes de la librería 1 tienen un error medio en torno al 23 %, y un error mediano del 12 %. Existe una pequeña diferencia de error entre unas parejas de fotos y otras, que se debe principalmente a la traslación entre las cámaras al tomar las imágenes utilizadas.

En el caso de la librería 2 se obtiene un error algo mayor: el error medio de todas las reconstrucciones llevadas a cabo se encuentra en torno al 33 %, mientras que el error mediano está en torno al 19 %. El aumento del error de la reconstrucción de fotos de esta librería se debe a que son fotos con más baja textura y menor definición.

En la librería 3 el error medio de todas las reconstrucciones es aproximadamente del 30 %, y el mediano del 24 %. Estos resultados son más parecidos a los de la segunda librería, ya que las fotos tienen baja textura como éstas. (Tabla 1)

	Error Medio		Error Mediano	
	(%)	(cm)	(%)	(cm)
Librería 1	23	14	12	7
Librería 2	33	44	19	26
Librería 3	30	40	24	30

Tabla 1: Resumen resultados promedio de las reconstrucciones

Los resultados de cada par de imágenes se recogen en el Anexo IV.

La principal causa de que el error de todas las reconstrucciones no sea el mismo es la distancia entre cámaras o *baseline* b. Estudiando este fenómeno se ha observado que a medida que aumenta la distancia entre las dos imágenes a utilizar, disminuyen el error medio y mediano, hasta alcanzar un mínimo a partir del cual empiezan a aumentar. (Ilustración 16)

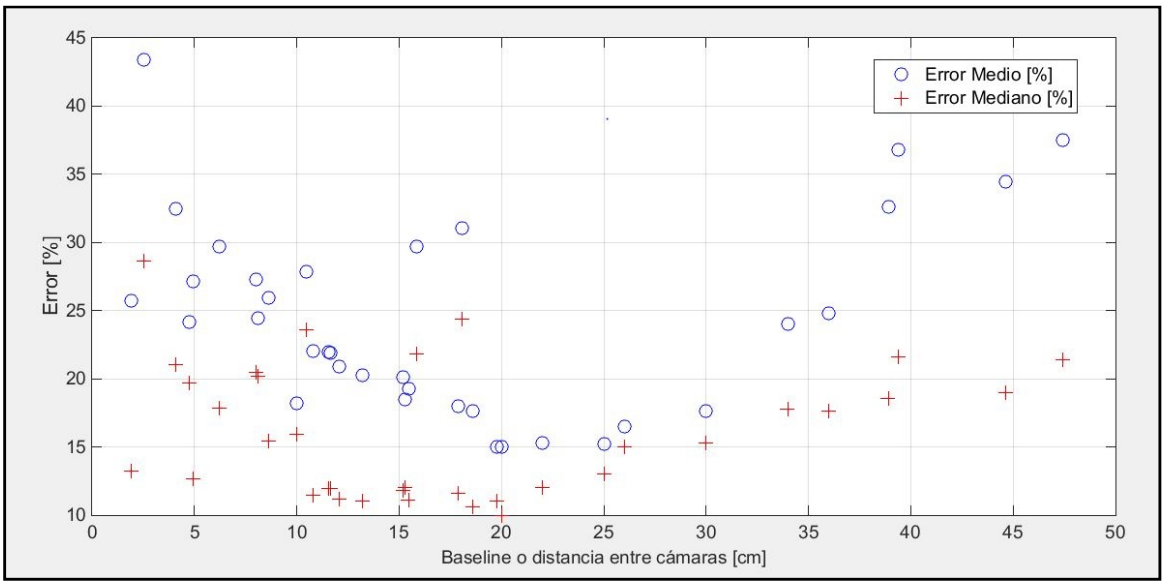

Ilustración 16: Variación del error medio y mediano al aumentar la distancia entre las imágenes utilizadas

Esta disminución del error se explica gráficamente en las ilustraciones 17 y 18.

Al utilizar píxeles se discretiza la imagen, concurriendo en un error intrínseco. Al unir el centro de la cámara con el píxel, éste se suele tomar como un punto, pero realmente tiene dimensiones. La pirámide formada por las aristas que pasan por el centro de la cámara y por los cuatro vértices del píxel es el margen de error debido a la discretización de la imagen, por tanto cuanta más resolución en píxeles tenga una imagen menor error tendrá el método. Al interferir las dos pirámides de dos píxeles de dos cámaras se obtendrá un volumen (mostrado en gris y en 2D en las ilustraciones 17 y 18), dentro del cual estará el punto a reconstruir. Este volumen es más alargado en el eje de la profundidad que en los ejes contenidos en el plano paralelo a la cámara, por lo que será en el eje de la profundidad en el que más error se cometa.

Un parámetro utilizado para cuantificar este error es el ángulo de paralaje α, ángulo entre la recta que une el centro de una cámara con el punto a reconstruir y la recta que une el centro de otra cámara con este mismo punto. Cuanto menor sea este ángulo, mayor será el error. El ángulo de paralaje α tiene valores bajos cuando las cámaras están muy próximas entre sí o cuando el punto a observar es muy lejano.

Es decir, se obtendrá un error más bajo al comparar imágenes con alta traslación y alta rotación entre éstas (especialmente la primera), y al reconstruir escenas cercanas.

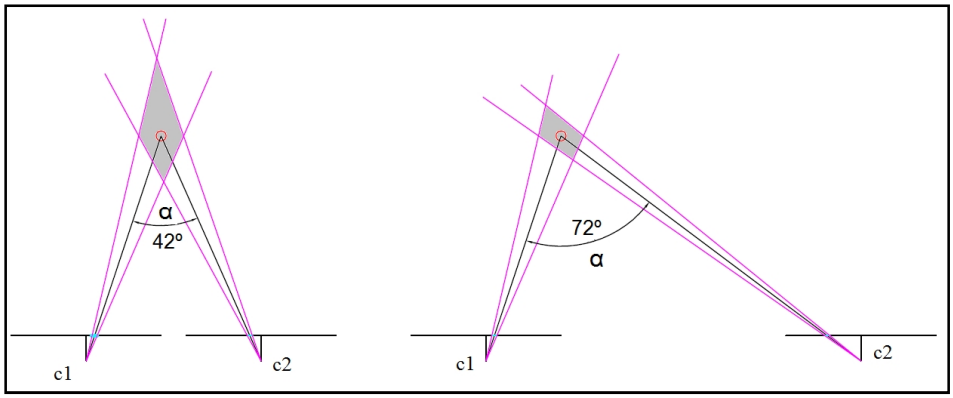

Ilustración 17: Vista en 2D de la disminución del error al aumentar la traslación entre las dos cámaras

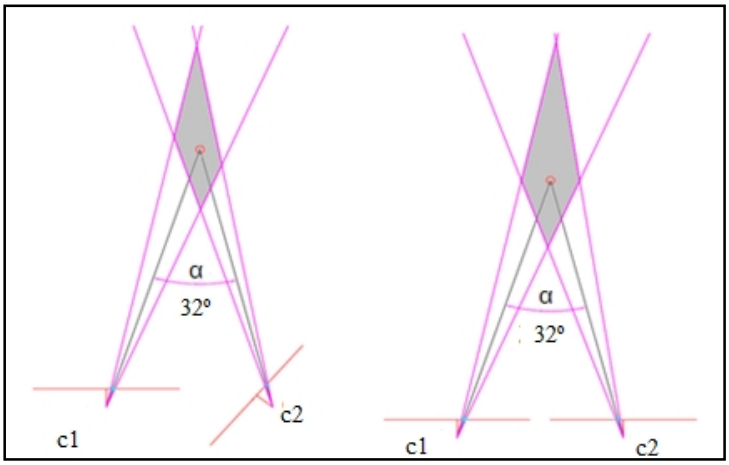

Ilustración 18: Vista en 2D de la disminución del error al aumentar la rotación entre las dos cámaras

Por otro lado, cuanto más separadas están las cámaras entre sí menor campo de visión tendrán en común, y más diferentes serán las escenas fotografiadas, por lo que hay que utilizar posiciones de las cámaras que cumplan cierto compromiso entre ambas observaciones. Este punto coincide con el mínimo de las funciones representadas en la ilustración 16.

Como se ha dicho antes, el ángulo de paralaje α depende de la distancia de los puntos a reconstruir y de la traslación entre cámaras. Si todas las imágenes utilizadas para las reconstrucciones están a una distancia similar de la cámara, la traslación óptima será la misma o muy parecida para cada par de imágenes, que en el caso de las imágenes utilizadas para el experimento sería entre 15 y 30 cm, como muestra la ilustración 16.

Pero si se utilizan imágenes cuya distancia a la cámara es muy diferente ya no será la misma la traslación óptima entre las cámaras. Estudiando la variación del error medio y mediano en función del ángulo de paralaje α (ilustraciones 19 y 20), para pares de imágenes con distancias a la cámara muy diferentes se observa un comportamiento claro. A mayor ángulo de paralaje menor error en la reconstrucción, hasta llegar a un mínimo a partir del cual el error empieza a aumentar. La curva obtenida en este experimento es más clara que en el experimento anterior ya que ahora no depende de la distancia del espacio fotografiado, si no que esta curva se mantiene siempre sean cuales sean las distancias.

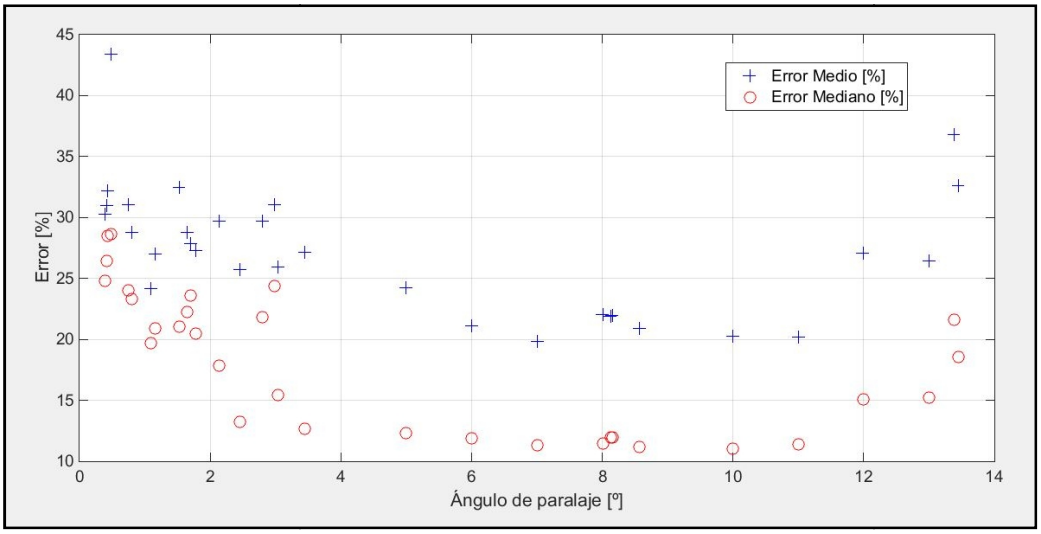

Ilustración 19: Variación del error al modificar el ángulo de paralaje entre cámaras

Hasta llegar al mínimo la mejora se debe a las razones geométricas explicadas anteriormente en las ilustraciones 17 y 18. El aumento progresivo del error a partir de un ángulo de paralaje de 10 ° se debe a que las escenas fotografiadas difieren cada vez más, y por tanto los emparejamientos son menos exactos. En la zona plana entre 5 y 10 ° se compensan estas dos tendencias opuestas creando una tendencia del error estable.

Se observa que el ángulo de paralaje óptimo a utilizar está entre 4 y 12°. Este ángulo se puede convertir en distancia entre las cámaras con la fórmula $\alpha = \cos^{-1}(\frac{(x-c_1)^T \times (x-c_2)}{norm(x-c_{1)}) \cdot norm(x-c_2)})$, siendo x la posición de un punto fotografiado respecto del origen de coordenadas mundo W, y c_1 y c_2 las posiciones de los centros de las cámaras 1 y 2 respectivamente, con el mismo origen de coordenadas W.

En la ilustración 20 se muestra el error hallado en metros frente al ángulo de paralaje para una distancia entre cámaras b fija igual a 1 m, así como la curva teórica (Anexo V).

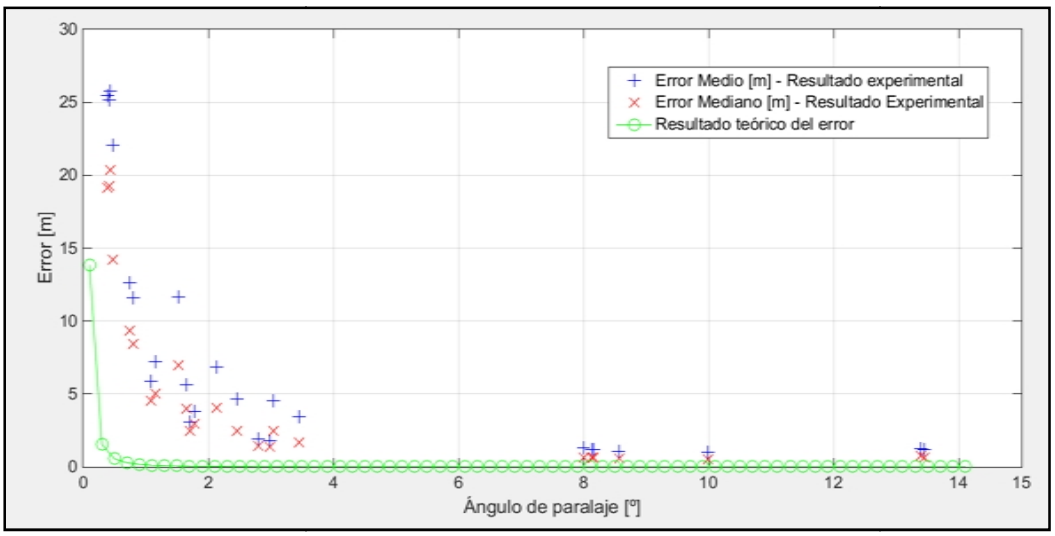

Ilustración 20: Variación del error [m] de las reconstrucciones al variar el ángulo de paralaje para una baseline fija

La curva teórica es calculada como $\Delta Z = \frac{\sigma_0 \cdot \sqrt{2}}{\alpha^2} \cdot b$, donde σ_0 es el ángulo de alcance de un píxel. Para hallar el ángulo σ_0 se ha hecho la siguiente aproximación: el ángulo β de la ilustración 21, que se define como el semi-ángulo de alcance en el eje X se puede calcular como $\beta = \arctan \frac{c_x}{f}$. Entonces, si todos los píxeles de la cámara en el eje X ($2 \cdot c_x$) abarcan el ángulo de alcance 2β, se puede aproximar que un píxel abarcará un ángulo σ_0.

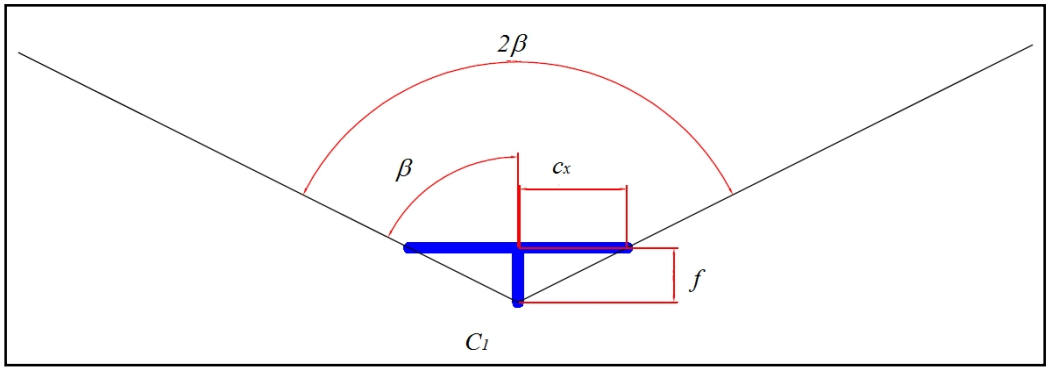

Ilustración 21: Parámetros geométricos de una cámara

La línea experimental se aproxima claramente a la línea teórica, dando siempre valores algo más grandes. Esto se debe a que la línea teórica está basada únicamente en aspectos geométricos, pero los resultados experimentales también se ven afectados por algunos emparejamientos erróneos.

En cuanto al aspecto que tiene una de las reconstrucciones llevadas a cabo, se muestra un ejemplo en la ilustración 22. La zona que aparece en negro es parte de la zona que no tienen en común las dos imágenes y que por tanto no ha podido ser reconstruida. Excepto por esta zona, las partes más oscuras de la imagen son las que están más cerca de la cámara, y las más blancas son las más lejanas. En este caso en particular se ha obtenido un error medio del 27.17 % o de 16.93 cm y un error mediano del 12.65 % o de 8.28 cm.

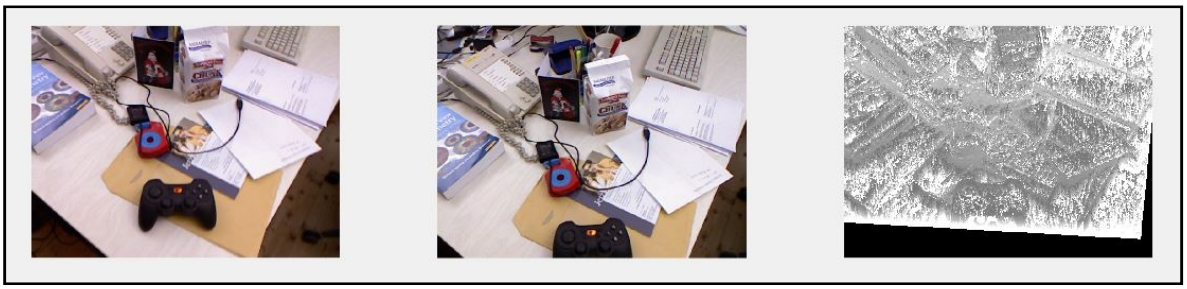

Ilustración 22: Imagen de profundidades (dcha) de una de las reconstrucciones (izda)

Se observa que hay mucho ruido en la reconstrucción, principalmente en las zonas con baja textura (zonas lisas como por ejemplo la mesa), por lo que se incorpora como objetivo a este trabajo el optimizar esta reconstrucción de escenas a partir de imágenes, lo cual se va a enfocar introduciendo cuatro variantes:

1. Utilización de parches
2. Introducción de un umbral
3. Comparación con más de una imagen
4. Introducción de un método de suavizado

Optimización mediante el uso de parches

Hasta ahora se habían comparado las dos imágenes a estudiar píxel a píxel. Ahora se van a comparar parches de 3 x 3 píxeles, 5 x 5, 7 x 7 ... de las dos imágenes a utilizar. De esta forma el emparejamiento de cada píxel será llevado a cabo no solamente a partir de su color sino también a partir de los colores de los píxeles circundantes.

En este caso es necesario prestar especial atención a que los pares de fotos utilizados tengan el ángulo de paralaje óptimo, para que no varíen de forma notoria los alrededores de los píxeles. Las imágenes utilizadas para este experimento son las mismas que se utilizaban en el anterior apartado.

En las tablas 2, 3 y 4 se observa como los valores de los errores medio y mediano van disminuyendo al aumentar el tamaño de los parches utilizados. Estos resultados son el promedio de múltiples pruebas realizadas sobre diferentes pares de imágenes de cada librería (Anexo VI).

Librería 1 Tamaño del parche	Error medio	Error mediano
1 x 1	22.85 %	11.93 %
3 x 3	20.45 %	8.86 %
5 x 5	18.35 %	7.02 %
7 x 7	17.20%	6.02 %
9 x 9	16.13 %	5.23%
11 x 11	15.24 %	4.73 %
13 x 13	14.58 %	4.39 %
15 x 15	13.99 %	4.19 %
17 x 17	13.57 %	4.08 %
19 x 19	13.26 %	4.01 %

Tabla 2: Resultados numéricos del error medio y mediano al utilizar parches de píxeles (librería 1)

Librería 2 Tamaño del parche	Error medio	Error mediano
1 x 1	32.79 %	19.28 %
3 x 3	29.61 %	16.02 %
5 x 5	27.61 %	14.17 %
7 x 7	25.43 %	12.18 %
9 x 9	24.05 %	10.95 %
11 x 11	22.97 %	10.08 %
13 x 13	22.16 %	9.46 %
15 x 15	21.54 %	8.95 %
17 x 17	20.95 %	8.52 %
19 x 19	20.45 %	8.19 %

Tabla 3: Resultados numéricos del error medio y mediano al utilizar parches de píxeles (librería 2)

Librería 3 Tamaño del parche	Error medio	Error mediano
1 x 1	29.88 %	24.32 %
3 x 3	29.71 %	23.50 %
5 x 5	29.15 %	22.85 %
7 x 7	28.51 %	22.19 %
9 x 9	27.94 %	21.59 %
11 x 11	27.49 %	21.13 %
13 x 13	27.11 %	20.74 %
15 x 15	26.74 %	20.28 %
17 x 17	26.48 %	20.02 %
19 x 19	26.38 %	19.84 %

Tabla 4: Resultados numéricos del error medio y mediano al utilizar parches de píxeles (librería 3)

En todos los ejemplos anteriores las reconstrucciones mejoran al introducir parches, pero la mejora no es igual en todos los casos. En la tercera librería la mejora es mucho menos apreciable debido a la baja textura que tienen estas imágenes. El ángulo de paralaje con el que han sido tomadas las imágenes también influye en que la mejora sea mayor o menor.

Se ha elegido el mismo ejemplo de la ilustración 22 para mostrar de forma más exhaustiva las mejoras que sufre la reconstrucción al aplicar la optimización mediante el uso de parches. En la ilustración 23 se ve el mapa de profundidades de la misma imagen al ir incorporando parches de diferentes tamaños. A medida que el tamaño de parche aumenta, la reconstrucción se vuelve más suave y el ruido va disminuyendo. En las primeras imágenes se observa una gran mejoría, pero a partir de parches de 13 x 13 píxeles apenas cambia la reconstrucción.

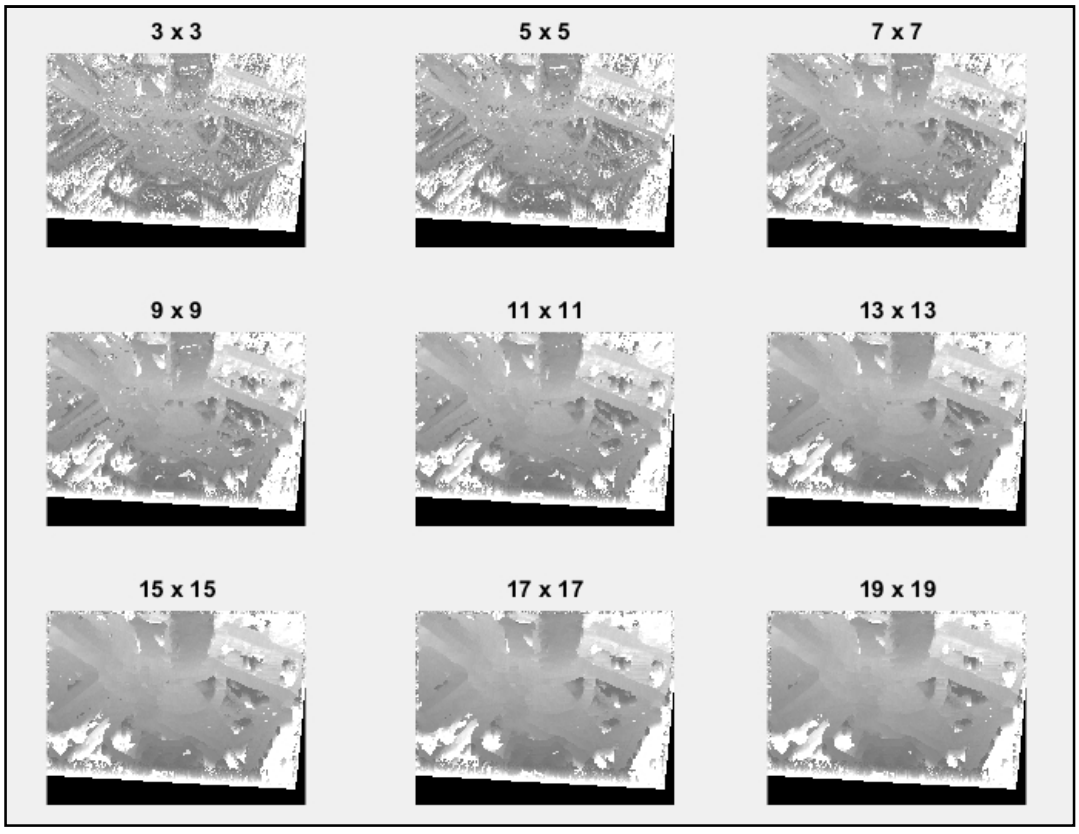

Ilustración 23: Mapa de profundidades de la misma reconstrucción para diferentes tamaños de parches

En la ilustración 24 se ve esta mejora de forma cuantitativa mediante un diagrama de "cajas y bigotes".

- A medida que aumenta el tamaño del parche hay más píxeles con errores casi nulos.
- El 75 % de los píxeles tienen un error por debajo del 30 % en el caso de no utilizar parches, y sin embargo, cuando se utiliza el parche de 19 x 19 píxeles el 75 % de los píxeles tienen un error por debajo del 10 %.
- Algo similar sucede con el 91 % de los píxeles: si no se utilizan parches el 91 % de los píxeles tienen errores menores del 85 %, pero si se utilizan parches de 19 x 19 píxeles el 91 % de éstos tienen un error por debajo del 58 %.

Estos cambios hacen que mejore también la media y mediana del error, que disminuyen de manera notable al ir aumentando el tamaño del parche. Es decir, al utilizar parches es mucho más fácil emparejar píxeles.

Los mayores cambios en los errores medio y mediano se producen con los parches más pequeños. Las mejoras que suponen los parches de a partir de 15 x 15 píxeles son mucho menos apreciables, por lo que un tamaño de parche razonable en este caso sería uno de 15 x 15 píxeles, ya que uno más grande no supondría una gran mejora y sin embargo los tiempos de ejecución aumentarían considerablemente.

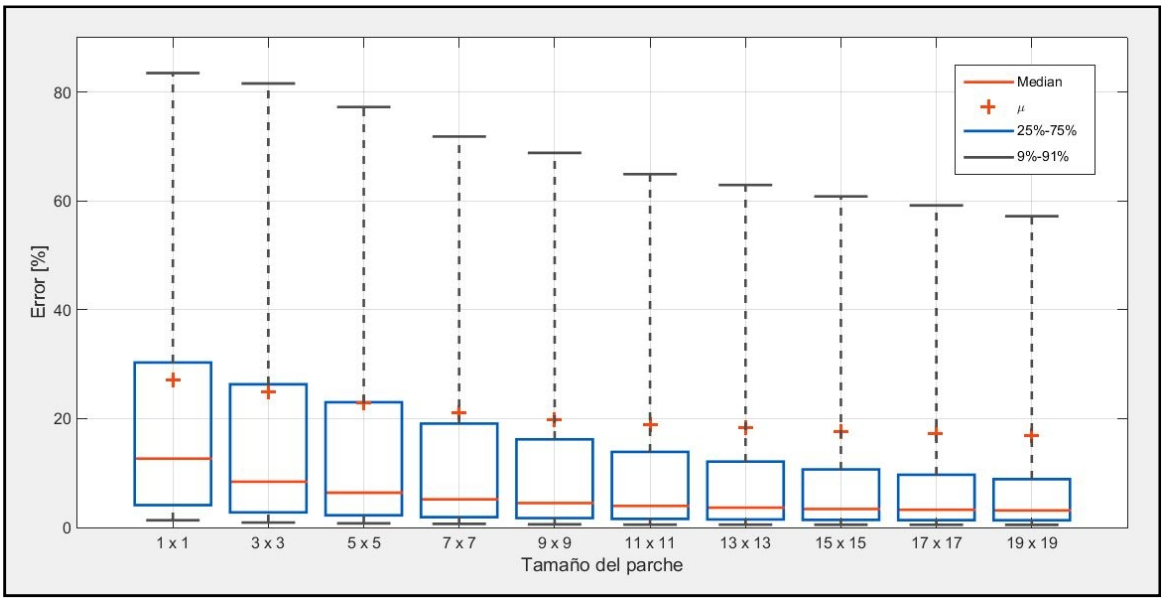

Ilustración 24: Variación del error al aumentar el tamaño de parche utilizado

El histograma del error varía de la forma en que se ve en la ilustración 25. Con el parche de 19 x 19 píxeles la distribución se vuelve mucho más plana, y se obtienen muchos más puntos con errores muy próximos a 0. Los píxeles con errores muy grandes disminuyen pero no llegan a desaparecer.

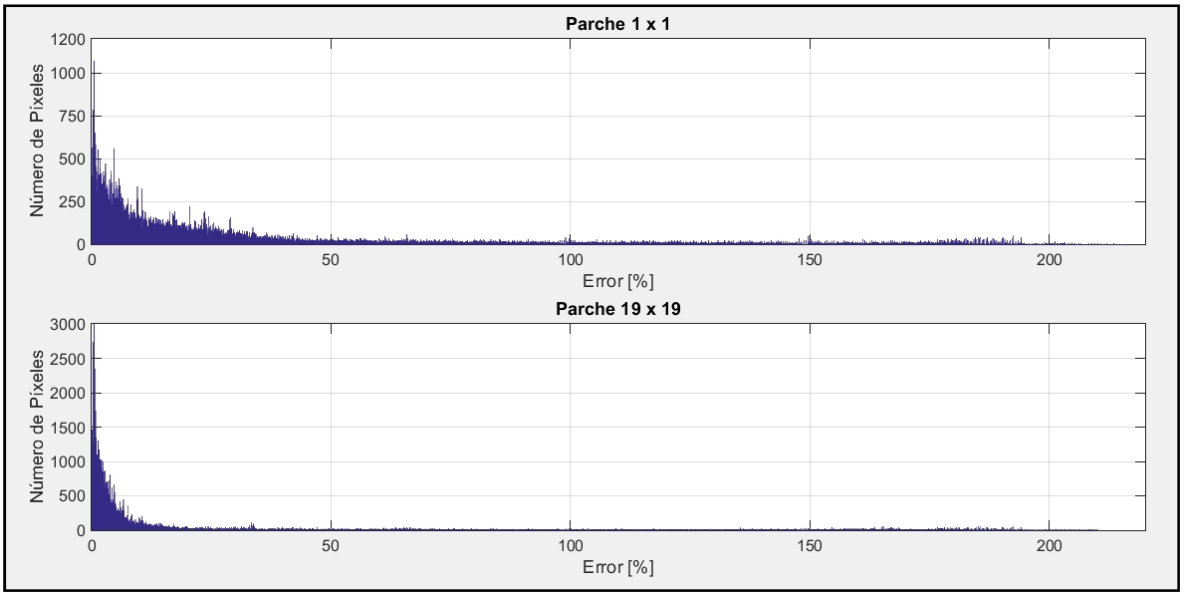

Ilustración 25: Comparación del histograma del error con diferentes tamaños de parches

El problema de la utilización de este parámetro surge cuando se aplica a imágenes que han sido tomadas con un ángulo de paralaje no óptimo entre ellas, ya que entonces no se verá lo mismo en una imagen que en otra, y un píxel no tendrá los mismos píxeles a su alrededor en una imagen que en otra. A medida que el ángulo de paralaje se aleja del óptimo habría que disminuir el tamaño del parche para que la utilización de este método de mejora fuera eficaz.

Optimización mediante el uso de umbrales

Como se ha explicado en el apartado de fundamentos físicos, una vez que se tiene hallada la recta epipolar, hay que elegir uno de los píxeles que la forman. Esto se hace cogiendo aquel que tenga menor error fotométrico, es decir, cuyo color sea más parecido al píxel en cuestión de la imagen a estudiar. El problema es que muchas veces se coge el píxel con menor error, a pesar de tener un color completamente diferente. Por esta razón se introduce el uso de umbrales, que marca la máxima diferencia de color entre dos píxeles para que la reconstrucción pueda ser correcta.

Para verificar el funcionamiento de este método se ha modificado este umbral desde 5 a 255 en varias reconstrucciones y se ha calculado el error medio y mediano en cada caso, así como el número de puntos reconstruidos. Un ejemplo de este método está representado en la ilustración 26.

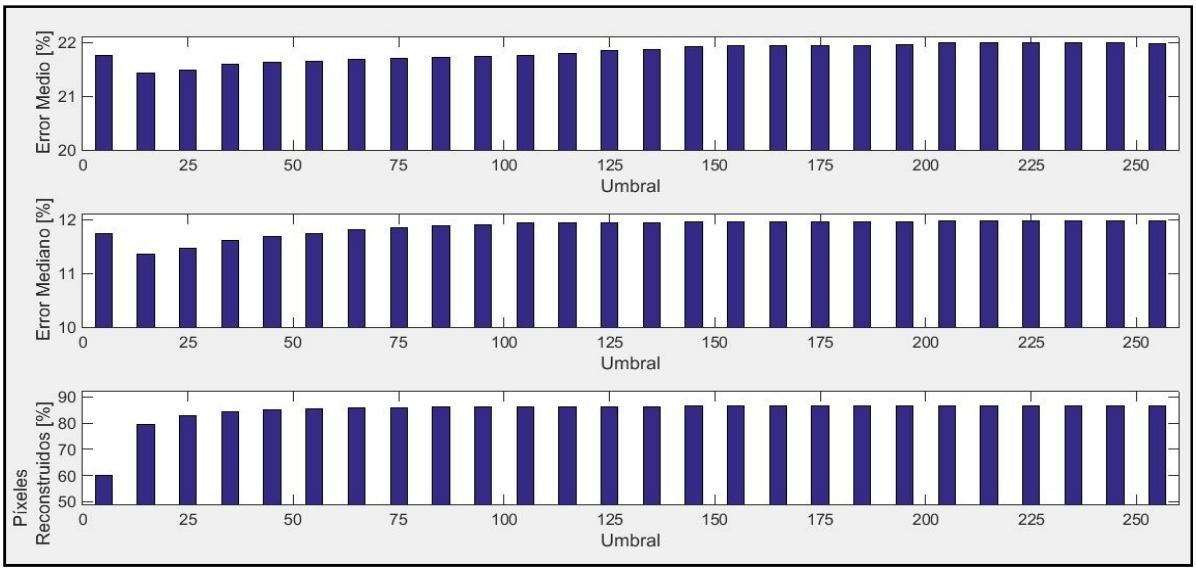

Ilustración 26: Variación del error y los puntos reconstruidos al variar el valor del umbral utilizado

Al introducir el umbral de valor 5, muchos puntos dejan de ser reconstruidos y los valores del error mejoran ligeramente, o no mejoran en algunos casos, respecto a utilizar un umbral de 255 (o no utilizar umbral). Cuando el umbral toma valores en torno a 15 hay muchos más puntos reconstruidos, y se aprecia la mejora del error medio y mediano. Como la parte más interesante de esta gráfica se encuentra entre los valores de umbral 5 y 25, en la ilustración 27 se muestra la misma gráfica con mayor detalle.

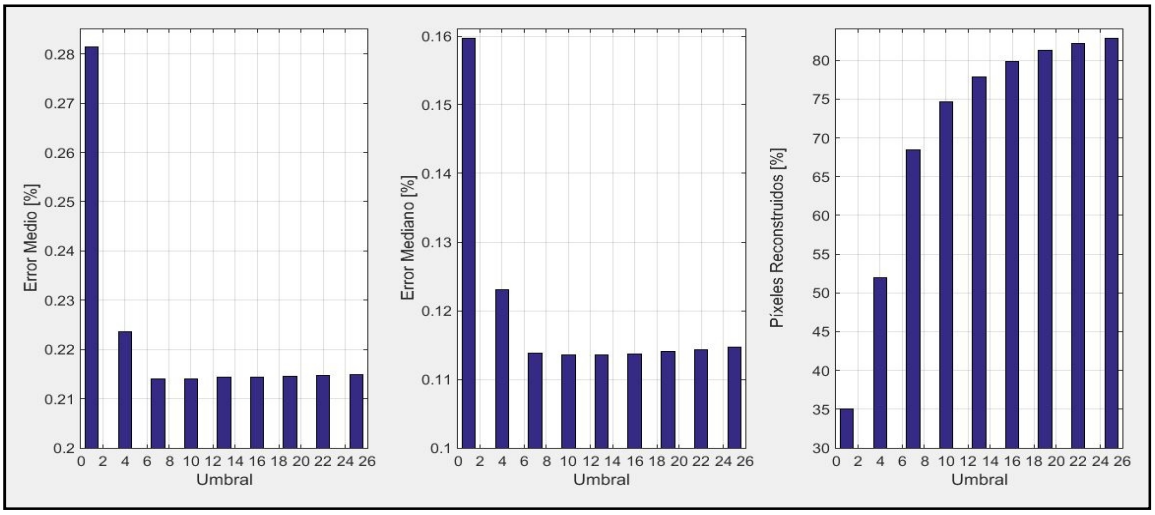

Ilustración 27: Variación del error y los puntos reconstruidos al variar el valor del umbral utilizado

El punto de inflexión en la evolución de los errores medio y mediano significa que aunque un píxel sea reconstruido con una diferencia de color de hasta 7 tonos (en el caso del ejemplo) no tiene por qué estar mal reconstruido. Serían los píxeles reconstruidos con diferencias de color mayores de 7 los que dan valores erróneos de profundidad.

El problema de coger el caso de umbral 7 es que hay muchos puntos que quedan sin reconstruir, por lo que el valor óptimo de umbral se situaría un poco superior a este punto de inflexión. El punto óptimo en este caso sería entonces un umbral de valor en torno a 10, cuya reconstrucción se muestra en la ilustración 28, comparada con la reconstrucción sin umbral.

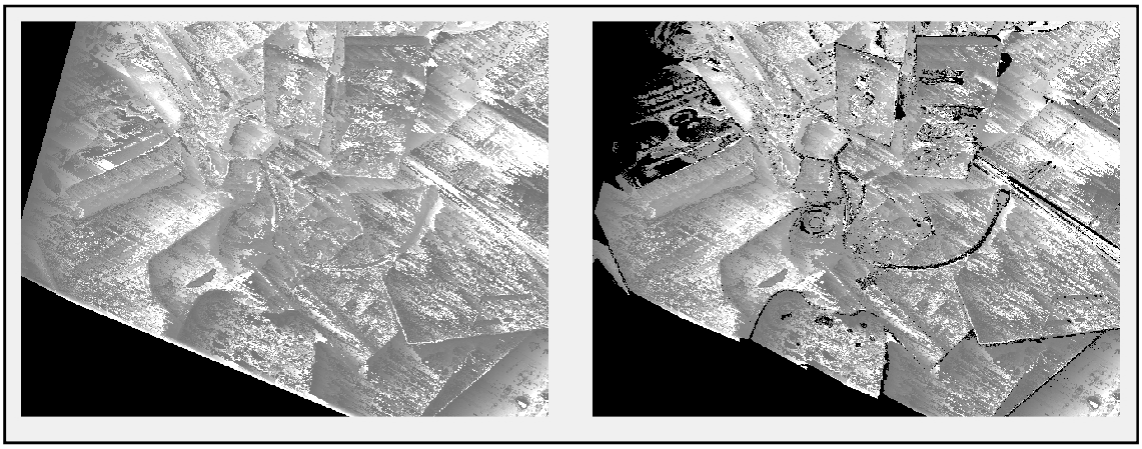

Ilustración 28: Comparación de los resultados sin umbral (izda) y con umbral (dcha)

Estos puntos que quedan sin reconstruir corresponden a los bordes de los objetos, donde los gradientes de color son muy altos, y/o principalmente a zonas que no tienen las dos imágenes en común, pero que han podido ser reconstruidas anteriormente por la elección de un punto erróneo de la recta epipolar.

Aquellos puntos de los que no se conocen las profundidades deberían ser reconstruidos de otra manera, ya sea mediante otras fotos o adoptando la profundidad de los píxeles de alrededor.

Todas las pruebas realizadas sobre diferentes pares de imágenes dan lugar al mismo tipo de resultados. A partir de estas pruebas, el umbral recomendado se sitúa siempre entre los valores 10 y 25, con el cual se obtiene una mejora mínima en la media y la mediana (Tabla 5).

Par de imágenes	Umbral Óptimo	Error Medio [%]		Error Mediano [%]	
		Con umbral	Sin umbral	Con umbral	Sin umbral
1	10	21.41	21.99	11.35	11.98
2	20	20.11	20.87	10.97	11.16
3	15	21.95	22.01	11.21	11.43
4	15	21.51	21.91	11.48	11.95
5	10	20.01	20.27	10.95	11.04
6	25	26.98	27.17	12.38	12.65
7	25	25.02	25.72	12.98	13.27

Tabla 5: Resultados de las pruebas realizadas con umbrales

Los pequeños cambios que se dan en el histograma del error se aprecian en la ilustración 29, donde se ve que no hay cambio alguno en los píxeles reconstruidos que tenían error bajo, pero que sí lo hay en los píxeles con errores grandes, aunque no muy apreciable.

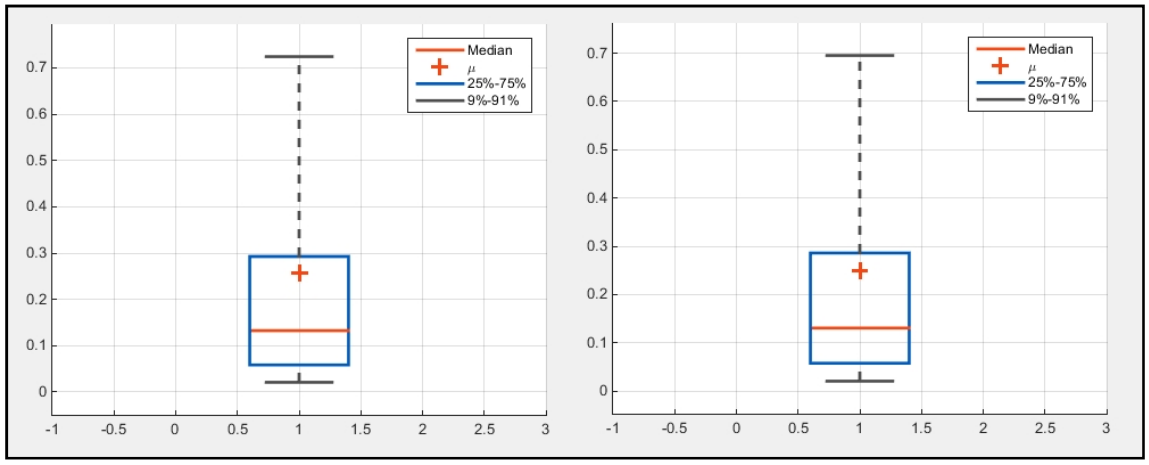

Ilustración 29: Cambio en la distribución del error al introducir el umbral óptimo en la reconstrucción (dcha)

Este método no mejora la reconstrucción, si no que permite detectar los errores más grandes para así poder ser corregidos mediante otros tipos de reconstrucción citados anteriormente.

Si se aplica este método junto con el de los parches el valor óptimo del umbral cambia: a medida que el tamaño del parche utilizado aumenta el valor del umbral óptimo también, de la manera que se muestra en la ilustración 30. Los errores medio y mediano disminuyen al mismo ritmo que si no se utilizaran umbrales.

La razón por la que el umbral óptimo aumenta al aumentar el tamaño del parche es la siguiente: para realizar esta prueba se calcula la media del color R, color G y color B de cada parche para la imagen de referencia y la imagen de apoyo, y la norma de la diferencia entre ambos dos es el umbral obtenido. Como con parches grandes se tienen en consideración más píxeles, es más probable que uno de ellos tenga un error fotométrico grande que afecte al error fotométrico del parche en cuestión.

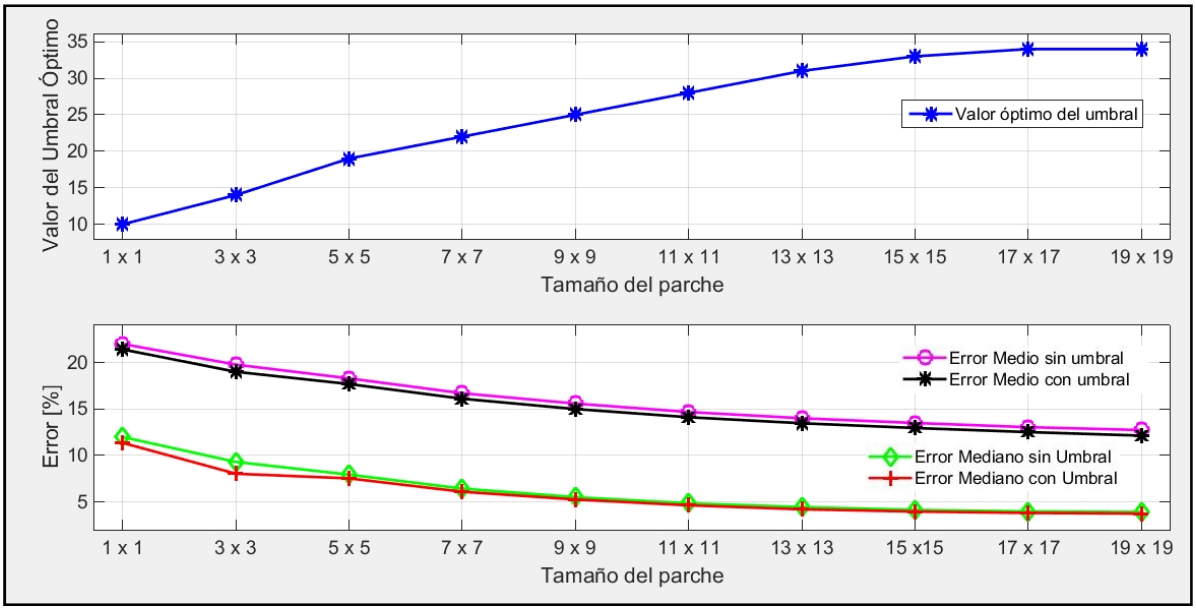

Ilustración 30: Variación del umbral óptimo al aumentar el tamaño del parche utilizado

Puesto que este método no supone una gran diferencia en los resultados, no se usará de ahora en adelante para las reconstrucciones fuera de este experimento.

Optimización mediante el uso de varias imágenes

Hasta ahora todas las imágenes utilizadas provenían de un dataset público (Sturm, 2009), en el cual también aparecían la traslación y rotación en formato cuaternio de la cámara al tomar cada imagen.

De ahora en adelante, la librería de imágenes utilizada proviene de una cámara particular, y las posiciones de las cámaras, la calibración de la cámara y las posiciones de algunos puntos salientes vienen dados por el software VisualSfM, que utiliza la técnica nombrada anteriormente de Sparse Mapping.

Las posiciones de la cámara al tomar las imágenes y los puntos característicos de éstas, hallados por el software VisualSfM aparecen en las ilustraciones 31 y 32.

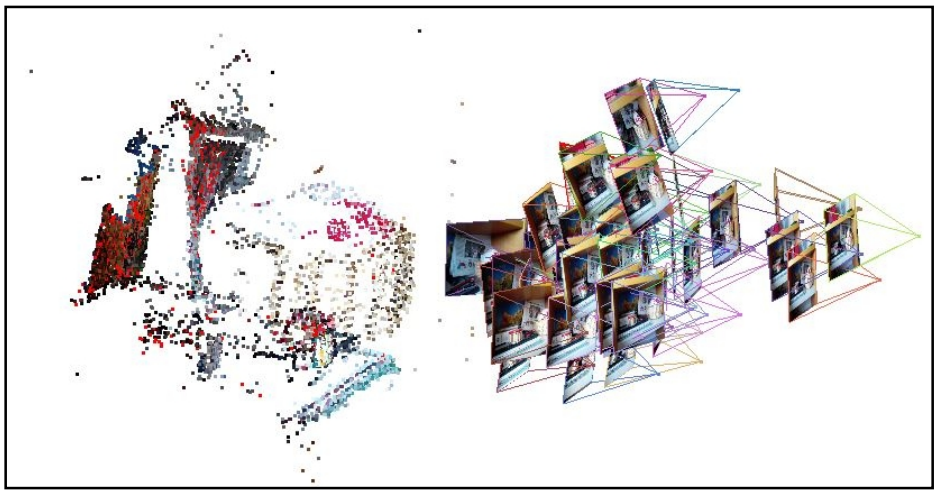

Ilustración 31: Interfaz de los resultados de Visual SFM (Ejemplo 1)

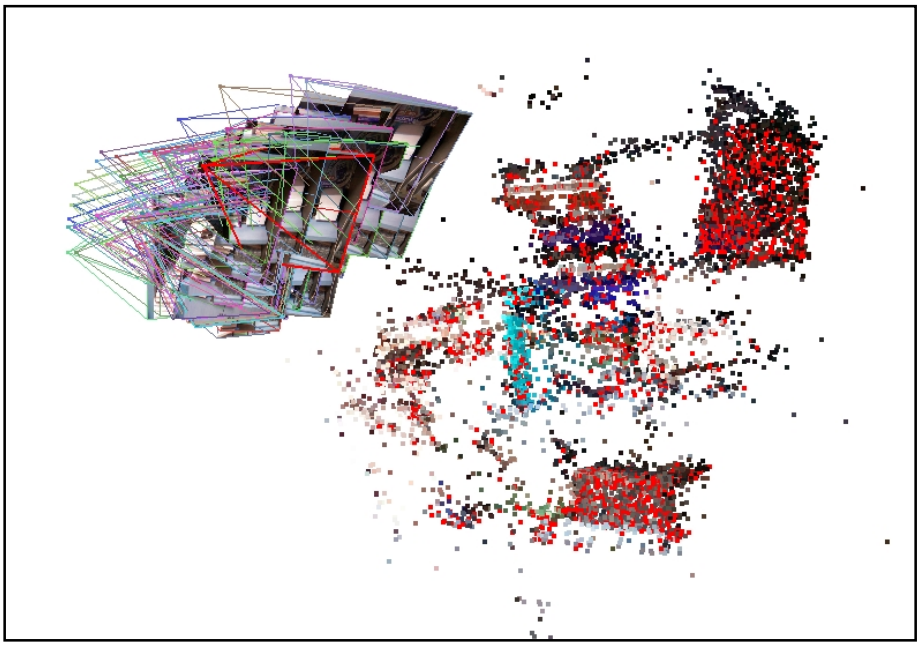

Ilustración 32: Interfaz de los resultados de Visual SFM (Ejemplo 2)

Este software, además de obtener la rotación y traslación de la cámara al tomar cada imagen, también obtiene la distancia focal de la cámara. La calibración en el eje x y la calibración en el eje y se obtienen de las hojas de características de la cámara particular, o del tamaño de las imágenes, dividiendo entre dos la resolución en píxeles horizontal y vertical.

La cámara utilizada también tiene un proyector de infrarrojos como la cámara Kinect, por lo que paralelamente se obtienen las profundidades reales. De esta forma, pueden ser comparadas con las profundidades obtenidas analizando las imágenes de la cámara RGB, y así poder medir el error.

Volviendo de forma específica a las mejoras a llevar a cabo, hasta ahora sólo se había utilizado una imagen de apoyo para cada imagen de referencia. Este experimento introduce más imágenes de apoyo para llevar a cabo las reconstrucciones. El número de fotos óptimo para la reconstrucción es la incógnita que queda por resolver ahora.

Dependiendo del número de fotos utilizadas se obtienen diferentes valores para las profundidades. Si se analiza el número óptimo de fotos necesarias para realizar la reconstrucción en 3D de una imagen, se obtienen los resultados de la ilustración 33.

Se observa que con diez imágenes el 75 % de los píxeles están medianamente bien reconstruidos, pero hay un 25 % de los píxeles con grandes errores. Al aumentar el número de fotos utilizadas la mejora del 75 % de los píxeles no es muy notoria, pero sin embargo los píxeles que estaban mal reconstruidos sufren una gran mejoría.

Esta evolución se estanca en todas las pruebas al llegar a 40 - 50 fotos en valores del error medio y mediano en torno al 6.5 % y 2.0 % respectivamente, por lo que el número óptimo para la reconstrucción de imágenes estaría en torno a 50 fotos.

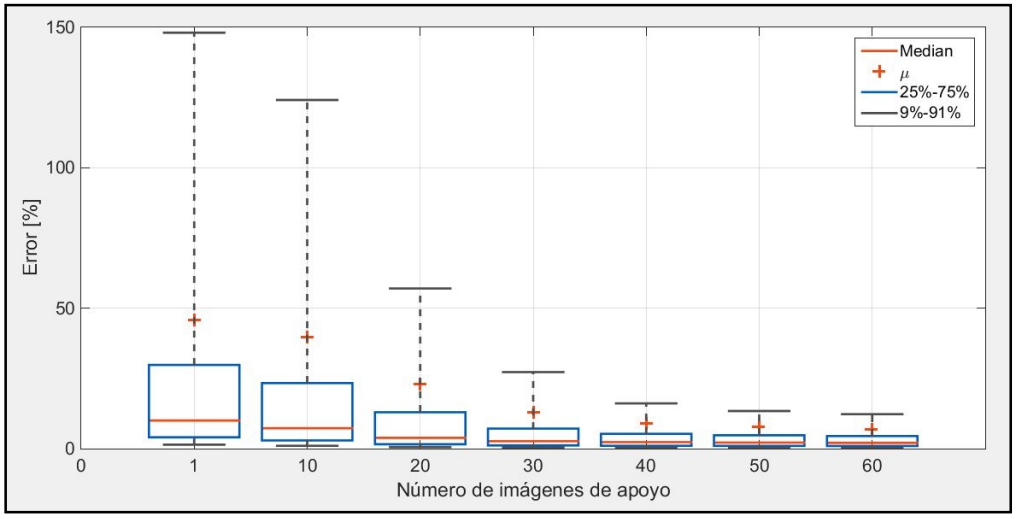

Ilustración 33: Variación del error al aumentar el número de imágenes utilizadas para la reconstrucción

Al introducir menos de 10 fotos no se ve una mejora clara en los resultados: dependiendo de las imágenes utilizadas los resultados mejoran o empeoran, es decir, no hay ninguna tendencia. Pero es a partir de la utilización de 10 fotos donde se ve una tendencia clara decreciente del error de la reconstrucción. El comportamiento medio de este método se ve en la tabla 6.

Número de fotos de apoyo	1		10		20		30		40		50		60	
	EM* [%]	EMo* [%]	EM [%]	EMo [%]	EM [%]	EMo [%]	EM [%]	EMo [%]	EM [%]	EMo [%]	EM [%]	EMo [%]	EM [%]	EMo [%]
Prueba	54.23	10.52	16.77	6.23	11.68	2.92	8.65	2.46	6.59	2.21	6.40	2.13	6.29	2.04

Tabla 6: Promedio de todas las pruebas realizadas al aumentar el número de imágenes de apoyo utilizadas
EM = Error Medio * EMo = Error Mediano *

Al utilizar una imagen de apoyo el error medio es más grande ahora que cuando se utilizaban las fotos provenientes del dataset, pero el error mediano es menor. Esta diferencia de valores puede deberse a particularidades de las fotos elegidas (ángulo de paralaje no óptimo) o a errores del VisualSfM al posicionar algún punto o cámara.

Los resultados de todas las pruebas realizadas se encuentran en el anexo VII.

Suavizado de la reconstrucción

El código utilizado para añadir esta mejora ha sido prestado de otro trabajo de fin de grado (Castillón Lacasa, 2015), por lo que no se va entrar en mucho detalle. La librería de imágenes es la misma que en el experimento anterior, y las posiciones de las cámaras, la calibración de la cámara y las posiciones de algunos puntos salientes vienen dados por el software VisualSfM.

En el caso de utilizar una imagen como apoyo, si no se utiliza la técnica de *Dense Mapping* completa los resultados son parecidos a los vistos anteriormente, y si se introduce el término del suavizado, el error disminuye notablemente. En la tabla 7 se ve un resumen de los resultados de las reconstrucciones con suavizado y sin suavizado aplicados a las mismas imágenes.

Par de imágenes	Sin suavizado		Con suavizado	
	Error Medio [%]	Error Mediano [%]	Error Medio [%]	Error Mediano [%]
1	20.24	12.37	23.56	3.56
2	62.87	8.91	43.06	2.98
3	64.86	11.02	46.10	3.49
4	59.31	10.83	38.31	3.78
5	56.80	10.84	35.41	3.71
6	50.50	9.13	29.66	3.13

Tabla 7: Comparación de los resultados con suavizado y sin suavizado

Un ejemplo de cómo queda una reconstrucción con y sin suavizado se ve en la ilustración 34, donde la imagen de la izquierda es la escena a reconstruir, la del medio es la reconstrucción sin suavizado, y la de la derecha es con suavizado.

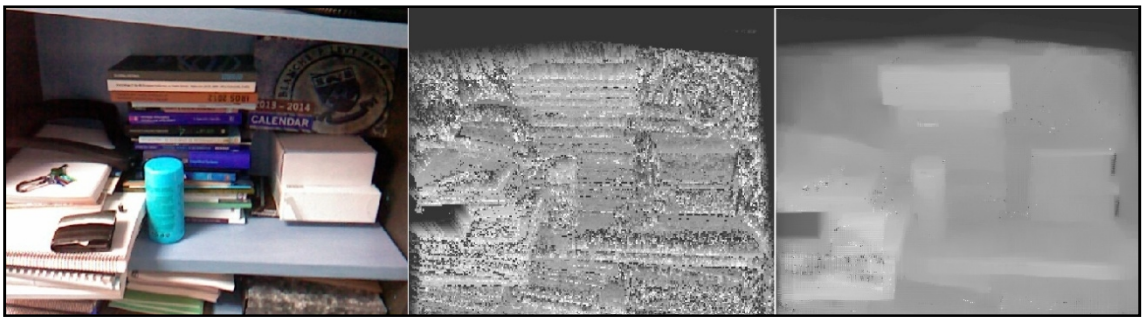

Ilustración 34: Mejora de la reconstrucción al introducir suavizado

En este caso en particular, la reconstrucción sin suavizado tiene un error medio de 56.80 %, y un error mediano del 10.84 %, mientras que la reconstrucción con suavizado tiene un error medio de 35.41 % y un error mediano de 3.71 %.

Si se utiliza suavizado con más de una imagen de apoyo, los resultados de un ejemplo son los que se ven en la ilustración 35, y los resultados promedio los que se leen en la tabla 8.

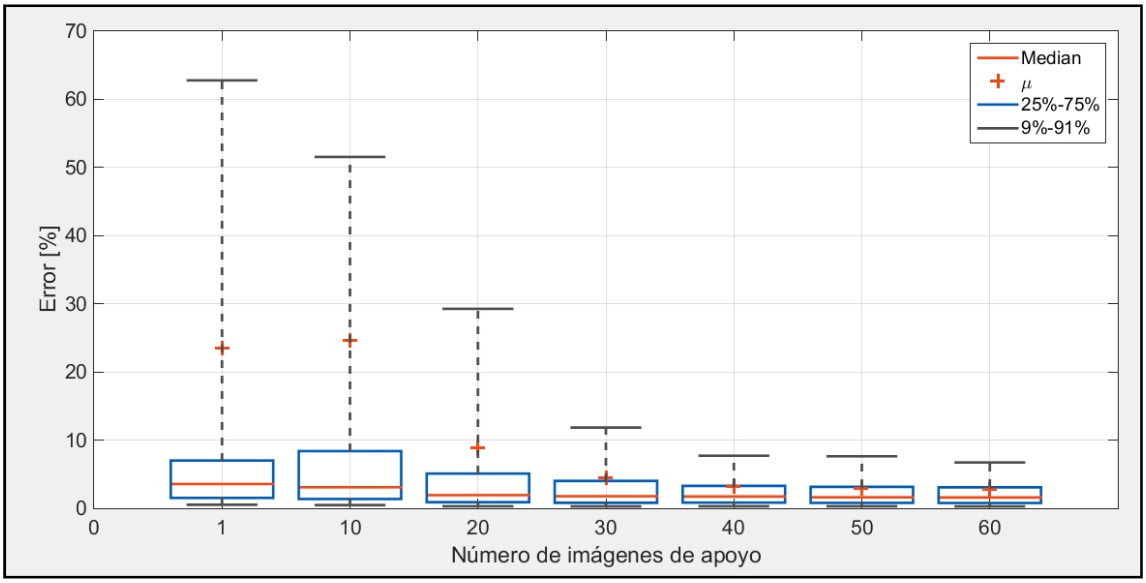

Ilustración 15: Mejora del error al introducir más de una imagen de apoyo con suavizado

La tendencia de mejora es la misma que la que había en el caso sin suavizado: la reducción del error se observa al utilizar más de 10 imágenes de apoyo, y lo que más mejora este método es el 25 % de los píxeles que tienen más error. Como ocurría anteriormente, el número óptimo de fotos está en torno a 40 - 50 fotos.

Número de fotos de apoyo	1		10		20		30		40		50		60	
	EM* [%]	EMo* [%]	EM [%]	EMo [%]	EM [%]	EMo [%]	EM [%]	EMo [%]	EM [%]	EMo [%]	EM [%]	EMo [%]	EM [%]	EMo [%]
Prueba	36.02	3.44	9.17	2.34	5.69	2.02	2.96	1.79	2.90	1.78	2.82	1.67	2.71	1.56

Tabla 8: Promedio de las pruebas realizadas al aumentar el número de imágenes de apoyo con suavizado
EM = Error Medio * EMo = Error Mediano *

En el caso sin suavizado se llegaban a obtener valores de 6.5 % en el error medio, y de 2.0 % en el error mediano. Con suavizado se llega a obtener un error medio de 3 % y un error mediano de 1.5 %. Los resultados de todas las pruebas realizadas se encuentran en el anexo VIII.

Finalmente, si se utilizan 50 imágenes de apoyo, suavizado y parches, los resultados promedio son los de la tabla 9, y los resultados de cada prueba se encuentran en el anexo IX.

Tamaño del Parche	Error Medio [%]	Error Mediano [%]
1 x 1	2.71	1.56
3 x 3	2.55	1.57
5 x 5	2.49	1.57
7 x 7	2.43	1.57
9 x 9	2.39	1.56
11 x 11	2.37	1.57
13 x 13	2.35	1.57
15 x 15	2.32	1.57
17 x 17	2.34	1.58
19 x 19	2.36	1.58

Tabla 9: Evolución del error con parches, suavizado y 50 fotos de apoyo

Al utilizar parches junto con el suavizado y 50 imágenes, la mejora que se veía al utilizar ambos métodos por separado se amortigua. Esto da lugar a una mejora del error poco apreciable hasta llegar a parches de 15 x 15 píxeles, y a partir de este tamaño de parche la reconstrucción empieza a empeorar.

Finalmente las mejores reconstrucciones se consiguen con 50 imágenes de apoyo, parches de 15 x 15 píxeles y suavizado. Para estas características los valores del error medio están en torno al 2.3 %, y los del error mediano en torno al 1.6 %. Una reconstrucción con estas características se puede ver en la ilustración 36.

Ilustración 36: Representación en escala de grises de las profundidades de una reconstrucción con suavizado, parches de 15 x 15 píxeles y 50 imágenes de apoyo

Para una mejor visualización de los resultados, las profundidades halladas de cada punto junto con el color de los píxeles son introducidas en el software Meshlab, que representa este mapa de profundidades como una nube de puntos (ilustración 37).

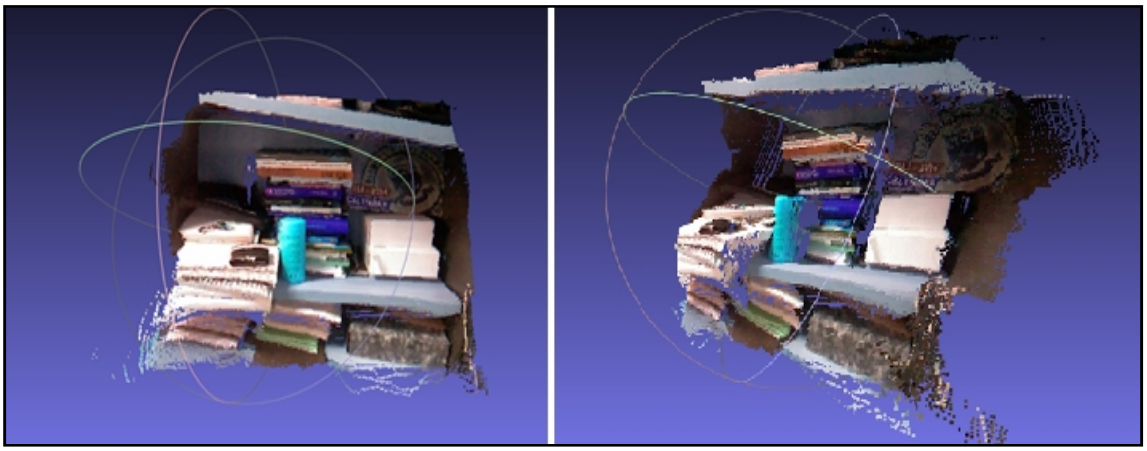

Ilustración 37: Representación con nube de puntos de la misma reconstrucción de la ilustración 36

Conclusiones

La elección de unas imágenes u otras para llevar a cabo la reconstrucción densa de imágenes supone uno de los puntos más importantes del problema.

Como se ha visto, lo que más influye en que las reconstrucciones llevadas a cabo den como resultado valores bajos de error es la elección de imágenes con alta textura, y un adecuado ángulo de paralaje entre las cámaras. Acorde con los experimentos llevados a cabo el ángulo de paralaje óptimo se encuentra entre 4 y 12 °.

Estas reconstrucciones tienen errores del orden del 29 % para el error medio, y del 18 % para el error mediano. Para mejorar estas reconstrucciones se pueden aplicar diferentes métodos de optimización:

- La aplicación de parches supone una gran mejora. En las pruebas realizadas, si se utilizan parches de 19 x 19 píxeles, la mejora de la media del error puede llegar a ser de hasta un 42 % respecto a no utilizar parches, y la mejora de la mediana hasta de un 66 %. Para obtener el máximo rendimiento con este método, las imágenes utilizadas deben tener una alta textura y haber sido tomadas con ángulos de paralaje óptimos para que así los píxeles a estudiar tengan los mismos píxeles alrededor.

- En cuanto a la introducción de umbrales, esto no supone una mejora en el error de las reconstrucciones, si no que detecta los píxeles que han sido mal reconstruidos para así poder mejorarlos con otros métodos. Estos píxeles mal reconstruidos suelen coincidir con bordes de objetos, y con zonas que no tienen en común las dos imágenes utilizadas. El valor del umbral depende del tamaño del parche utilizado: si no se utilizan parches el umbral recomendado se situaría en torno a 15, mientras que si se utilizan parches de 19 x 19 píxeles el umbral estaría en torno a 35.

Cuando se dispone de una cámara de infrarrojos como la cámara Kinect se obtienen directamente las posiciones de las cámaras al fotografiar una escena, que es como se ha hecho en los experimentos anteriores. Sin embargo, con una cámara de uso ordinario hay que procesar las imágenes tomadas con el software VisualSfM para obtener las posiciones. De esta manera los errores medio y mediano toman valores en torno al 54 % y al 10 % respectivamente. Los demás métodos de mejora se han aplicado sobre esta librería de imágenes:

- Utilizar múltiples imágenes de apoyo mejora la reconstrucción, siempre y cuando se utilicen más de 10 fotos de apoyo. El número óptimo de imágenes a utilizar está entonces entre 40 y 50, valor para el cual se estanca la tendencia de mejora del error. Se consiguen reconstrucciones con errores medios del 6.3 % y medianos del 2.0 %.

- Si se introduce la técnica de suavizado, la reconstrucción mejora notablemente, alcanzando valores del error medio en torno al 36 % y del error mediano en torno al 3.4 %.

- Al introducir conjuntamente la técnica de suavizado y más de una imagen de apoyo se llega a la conclusión de que el número óptimo de fotos sigue siendo alrededor de 50 fotos. De esta forma el error medio obtenido es del 2.7 % y el mediano del 1.6 %.

- Finalmente, si se aplica conjuntamente la técnica de suavizado, parches y 50 imágenes de apoyo se ve que la mejora que aportaban anteriormente los parches se amortigua y cambian muy ligeramente los resultados. Con un parche de 15 x 15 píxeles se llegan a obtener valores del error medio en torno a 2.3 % y del mediano en torno a 1.6 %, y sin embargo con píxeles de mayor tamaño el error empieza a aumentar.

Como conclusión final, la mejora óptima de la reconstrucción se obtiene utilizando parches de 15 x 15 píxeles, suavizado y 50 imágenes de apoyo. Sin embargo, si se quisiera ahorrar tiempo de ejecución se podrían utilizar parches más pequeños o incluso no utilizar, ya que la mejora que estos aportan es mínima.

En las tablas 10 y 11 se encuentra un resumen de los errores medios obtenidos con los diferentes métodos.

	Error Medio [%]	Error Mediano [%]
Estándar	29.0	18.0
Parches (19 x 19)	20.0	11.0
Umbrales	28.0	17.0

Tabla 10: Resumen de los resultados experimentales realizados con fotos provenientes de un dataset público (Sturm, 2009)

	Error Medio [%]	Error Mediano [%]
Estándar	54.2	10.5
50 fotos de apoyo	6.3	2.0
Suavizado	36.0	3.4
50 fotos de apoyo + Suavizado	2.7	1.6
Parches (15 x 15) + 50 fotos de apoyo + Suavizado	2.3	1.6

Tabla 10: Resumen de los resultados experimentales realizados con fotos de una cámara particular

En cuanto a las líneas de investigación futuras de este trabajo se propone la optimización del código utilizado para así reducir la velocidad de ejecución, ya que ha supuesto un importante inconveniente a la hora de trabajar. Otra opción podría ser trabajar en exteriores para ver cómo se comportan estos métodos, ya que en este trabajo sólo se han hecho experimentos en zonas cerradas.

Bibliografía

Blanes, F., Jiménez, L. M., Puerto, R., Ñeco, R., & Reinoso, O. (2005). Reconstrucción tridimensional de escenas con un par estereoscópico de cámaras.

Canny, J. (1986). *A computational approach to edge detection.*

Castillón Lacasa, R. (2015). Evaluación de métodos densos de reconstruccóon 3D a partir de imágenes.

Davis, L. (1975). *A survey of edge detection techniques.*

Goesele, M., Snavely, N., Curless, B., Hoppe, H., & Seitz, S. M. (2007). *Multi-view stereo for community photo collections.* Río de Janeiro.

Guerig, G. (2012). *Structured Lightning.* Utah.

Longuet-Higgins, H. C. (1981). *A computer algorithm for reconstructing a scene from two projections.* Nature.

Lowe, D. (2003). *Distinctive Image Features from Scale Invariant Keypoints.* Vancuver.

Martínez Montiel, J. M. *Apuntes de la asignatura Visión por Computador.* Zaragoza, España.

Pentland, A. P. (1984). *Local shading analyisis.*

Ponce, J., Hebert, M., Schmid, C., & Zisserman, A. *Toward Category - Level Object Recognition.* Nueva York.

Richard A. Newcombe, S. J. *DTAM: Dense Tracking and Mapping in Real-Time.* London, UK.

Rosenfeld, A., & Kak, A. C. (1976). *Digital Picture Processing.* New York: Academic Press.

Rosenfeld, A., & Pfaltz, J. (1966). *Sequential operations in digital picture processing.*

Seitz, S., & Szeliski, R. (1999). *Applications of computer vision to computer graphics.*

Sidenbladh, H., Black, M. J., & Fleet, D. J. (2000). *Stochastic tracking of 3D human figures using 2D image motion.* Dublin.

Sivic, J., Zitnick, C. L., & Szeliski, R. (2006). *Finding people in repeated shots of the same scene.* Edinburgh.

Snavely, N., Seitz, S. M., & Szeliski, R. (2007). Modeling the World from Internet Photo Collections.

Snavely, N., Seitz, S. M., & Szeliski, R. (2006). *Photo tourism: Exploring photo collections in 3D.* SIGGRAPH .

Sturm, J. (2009). *Technische Universität München*. Recuperado el Febrero de 2015

Szeliski, R. (2010). *Computer Vision: Algorithms and Applications.* Springer.

Ullman, S. (1979). *The interpretation of structure from motion.* Londres.

Anexos

Anexo I

Las iteraciones a seguir para encontrar el valor óptimo de la ecuación 1 se basan en principios de dualidad.

$$E_{\xi,\alpha} = \int_{\Omega} \{g(u) \cdot \|\nabla \xi(u)\|_{\in} + \frac{1}{2\theta}(\xi(u) - \alpha(u))^2 + \lambda C(u, \alpha(u))\} du$$

Ecuación 1

De esta forma se obtiene la ecuación dual de la ecuación 1, que es la que aparece en la ecuación 2. Aparecen varias variables auxiliares como q y δ_q. Además α y ξ son las variables correspondientes a a y d respectivamente en forma vectorial.

$$E(d,a,q) = \{\langle AGd, q\rangle + \frac{1}{2\theta}\|d-a\|_2^2 + \lambda C(a) - \delta_q(q) - \frac{\in}{2}\|q\|_2^2\}$$

Ecuación 2

Una vez que la variable auxiliar a está fijada correctamente ($a \to d$), la condición para que $E(d,a,q)$ sea óptima se obtiene al resolver $\partial_{d,q}(E(d,a,q)) = 0$. Para la variable dual q, la optimización se encuentra en la ecuación 3.

$$\frac{\partial E(d,a,q)}{\partial q} = AGd - \in q = 0$$

Ecuación 3

Sin embargo, para optimizar respecto de la variable dual d se necesita utilizar el teorema de divergencia, con el cual se puede concluir que $\langle AGd, q\rangle = \langle A^T q, Gd\rangle$. La optimización se reduce entonces a resolver la ecuación 4.

$$\frac{\partial E(d,a,q)}{\partial d} = GA^T q + \frac{1}{\theta}(d-a) = 0$$

Ecuación 4

El término G es una matriz diagonal que contiene el peso relativo de los píxeles, mientras que $\delta_q(q)$ es una función definida como $\delta_q(q) = \begin{cases} 0 & if \|q\|_1 \le 1 \\ \infty & otherwise \end{cases}$.

Para cada valor fijo d se obtiene la solución para cada $a_u = a(u)$ en la ecuación resultante de todas las anteriores (ecuación 5).

$$E^{aux}(u, d_u, a_u) = \frac{1}{2\theta}(d_u - a_u)^2 + \lambda C(u, a_u)$$

Ecuación 5

Para empezar a iterar hay que fijar unos valores iniciales para la primera iteración (n = 0):

- $q^0 = 0$

- $d_u^0 = a_u^0 = \arg\min_{a_u \in D} C(u, a_u)$

A partir de estos valores debe fijarse el actual valor de a^n e ir resolviendo las ecuaciones 6 y 7.

$$q^{n+1} = \prod_q ((q^n + \sigma_q GAd^n)/(1 + \sigma_q \in))$$

Ecuación 6

donde $\prod_q (x) = x / \max(1, \|x\|_2)$

$$d^{n+1} = (d^n + \sigma_d (GA^T q^{n+1} + \frac{1}{\theta^n} a^n))/(1 + \frac{\sigma_d}{\theta^n})$$

Ecuación 7

Anexo II

Teniendo dos cámaras C_1 y C_2 con su respectiva distancia focal f fotografiando una misma escena (ilustración A1), el punto P, situado a una distancia concreta (X y Z) del centro de la cámara C_1, se proyecta en las respectivas cámaras en P_1 y P_2.
Estos puntos P_1 y P_2 se sitúan a una distancia x_1 y x_2 respectivamente de sus centros de cámara.

Esta proyección de P sobre ambas cámaras se hace con las fórmulas $x_1 = f \cdot \frac{X_{C_1}}{Z} = f \cdot \frac{X}{Z}$ y $x_2 = f \cdot \frac{X_{C_2}}{Z} = f \cdot \frac{X - b}{Z}$, siendo b la distancia entre los dos centros de cámara.

Despejando ambas X y Z de las fórmulas se obtiene $X = \frac{x_1 b}{x_1 - x_2}$ y $Y = \frac{bf}{x_1 - x_2}$. Si se le llama a la distancia entre x_1 y x_2 disparidad d la profundidad Z queda $Z = \frac{bf}{d}$.

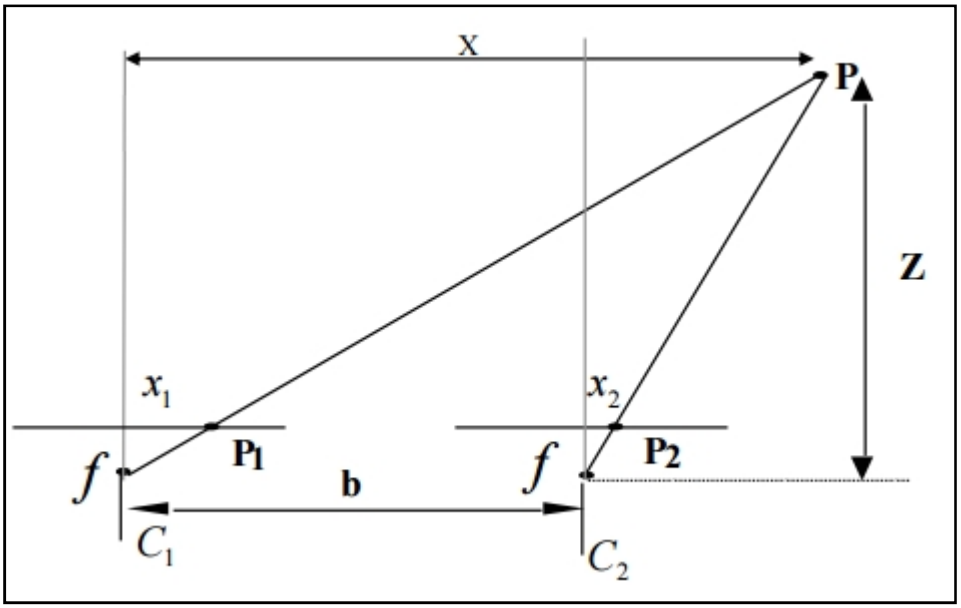

Ilustración A1: Esquema de visión en 2D (Martínez Montiel)

Si se hace un análisis de la sensibilidad de la profundidad Z respecto de la disparidad d se obtiene para un punto a una distancia $Z = Z_0$ la siguiente ecuación: $Z = Z_0 + \Delta d \frac{\partial Z}{\partial d} = Z_0 - \Delta d \frac{fb}{d^2} = Z_0 - \frac{\Delta d}{fb} Z_0^2$, y por lo tanto, el error ΔZ resulta $\Delta Z = -\frac{\Delta d}{fb} Z_0^2$. El error cometido al calcular la profundidad de un punto P varía de manera cuadrática con la distancia Z_0 a la que este punto se encuentra.

En el caso de utilizar el error en porcentaje ($\Delta Z(\%) = \frac{\Delta Z}{Z_0} = -\frac{d}{fb} Z_0$) se obtiene una función lineal del error.

Anexo III

El código de MATLAB utilizado para hacer la reconstrucción de una imagen a partir de otra de referencia sin parches y sin umbrales se encuentra en este anexo. Se parte de un dataset (Sturm, 2009) que contiene imágenes, sus profundidades reales, la posición de las cámaras al tomar todas las imágenes, y la calibración de la cámara.

```
clc
clear all
close all

images_path = 'C:\...\rgb\';              % Imágenes RGB
image_names = dir([images_path '*.png']);

dimages_path = 'C:\...\depth\';           % Profundidad de las escenas
dimage_names = dir([dimages_path '*.png']);

gt_poses_file = 'C:\...\groundtruth.txt';% Posición de las cámaras
gt_poses = dlmread(gt_poses_file);

factor = 1;
Camera.calibration.f = 525.0*factor;
Camera.calibration.cx = 319.5*factor;
Camera.calibration.cy = 239.5*factor;
Camera.K = [Camera.calibration.f 0 Camera.calibration.cx
    0 Camera.calibration.f Camera.calibration.cy
    0 0 1];

Im1 = 554;
Image1.rgb = imresize(imread([images_path
      image_names(Im1).name]),factor);
q_ref =
      interp1(gt_poses(:,1),gt_poses(:,5:8),str2double(image_names(Im1
      ).name(1:end-4)));
Image1.pose.R = q2r(q_ref)/norm(q2r(q_ref));
t_ref =
      interp1(gt_poses(:,1),gt_poses(:,2:4),str2double(image_names(Im1
      ).name(1:end-4)));
Image1.pose.t = t_ref';
Image1.T = [[Image1.pose.R,Image1.pose.t];[zeros(1,3),1]];

Im2 = 610;
Image2.rgb = imresize(imread([images_path
image_names(Im2).name]),factor);
q_ref =
      interp1(gt_poses(:,1),gt_poses(:,5:8),str2double(image_names(Im2
      ).name(1:end-4)));
Image2.pose.R = q2r(q_ref)/norm(q2r(q_ref));
t_ref =
      interp1(gt_poses(:,1),gt_poses(:,2:4),str2double(image_names(Im2
      ).name(1:end-4)));
Image2.pose.t = t_ref';
Image2.T = [[Image2.pose.R,Image2.pose.t];[zeros(1,3),1]];

depthimage = loadrealdepth(Im1,factor);
min_Z = 0.01;
max_Z = 3.50;
```

```
paso_Z = 0.01;
ancho = 640*factor;
alto = 480*factor;
Patch = 1;
x = (max_Z-min_Z)/paso_Z;
Position = zeros(x,4);
Diff = zeros(x,1);
CorrespondenciaPixeles = zeros(alto,ancho,4);

for i=1:alto
    i
    for j=1:ancho
        Z = 0;
        Punto.Imagen1 = [j
                i
                1];
        Punto.Color1.r = single(Image1.rgb(i,j,1));
        Punto.Color1.g = single(Image1.rgb(i,j,2));
        Punto.Color1.b = single(Image1.rgb(i,j,3));
        Punto.C1_3 = Camera.K^(-1)*Punto.Imagen1;
        for z = min_Z:paso_Z:max_Z
            Punto.C1_4 = [Punto.C1_3;1/z];
            Punto.W = Image1.T*Punto.C1_4;
            Punto.C2_4 = Image2.T^(-1)*Punto.W;
            Punto.Imagen2 = Camera.K*[eye(3),zeros(3,1)]*Punto.C2_4;
            Punto.Imagen2 = Punto.Imagen2/Punto.Imagen2(3,1);
            Punto.Imagen2(1:2,1) = fix(Punto.Imagen2(1:2,1))+1;
            if Punto.Imagen2(1,1)>0 && Punto.Imagen2(1,1)<=ancho &&
                  Punto.Imagen2(2,1)>0 && Punto.Imagen2(2,1)<=alto
                Z = Z+1;
                Punto.Color2.r =
      single(Image2.rgb(Punto.Imagen2(2,1),Punto.Imagen2(1,1),1));
                Punto.Color2.g =
      single(Image2.rgb(Punto.Imagen2(2,1),Punto.Imagen2(1,1),2));
                Punto.Color2.b =
      single(Image2.rgb(Punto.Imagen2(2,1),Punto.Imagen2(1,1),3));
                Resta = [abs(Punto.Color1.r -
      Punto.Color2.r),abs(Punto.Color1.g -
      Punto.Color2.g),abs(Punto.Color1.b - Punto.Color2.b)];
                Diff(Z,1) = norm (Resta);
                Position(Z,1) = Punto.W(1,1)*z;
                Position(Z,2) = Punto.W(2,1)*z;
                Position(Z,3) = Punto.W(3,1)*z;
                Position(Z,4) = z;
            end
        end
        if Z ~= 0
            Position = Position(1:Z,:);
            Diff = Diff(1:Z,:);
            [Min,pos] = min(Diff);
            CorrespondenciaPixeles(i,j,1) = Position(pos,1);
            CorrespondenciaPixeles(i,j,2) = Position(pos,2);
            CorrespondenciaPixeles(i,j,3) = Position(pos,3);
            CorrespondenciaPixeles(i,j,4) = Position(pos,4);
        end

    end
end
```

Anexo IV

En la memoria principal se expone el promedio de los resultados de todas las pruebas utilizadas. En este anexo se encuentra el resultado de cada prueba realizada en las tablas A1, A2 y A3.

Par de imágenes Librería 1	Error Medio [%]	Error Medio [cm]	Error Mediano [%]	Error Mediano [cm]
1	21.99	13.85	11.98	7.40
2	20.87	13.18	11.16	6.80
3	22.01	13.90	11.43	6.88
4	21.91	13.81	11.95	7.34
5	20.27	12.91	11.04	6.64
6	27.17	16.93	12.65	8.28
7	25.72	16.48	13.27	8.72

Tabla A1: Resultados de todos los experimentos llevados a cabo sobre la librería 1

Par de imágenes Librería 2	Error Medio [%]	Error Medio [cm]	Error Mediano [%]	Error Mediano [cm]
1	32.46	47.76	21.07	28.56
2	29.72	42.43	17.87	24.9
3	25.96	39.09	15.40	21.22
4	37.51	35.30	21.43	24.04
5	34.45	47.62	19.01	27.82
6	32.64	43.49	18.55	24.86
7	36.78	48.84	21.63	30.5

Tabla A2: Resultados de todos los experimentos llevados a cabo sobre la librería 2

Par de imágenes Librería 3	Error Medio [%]	Error Medio [cm]	Error Mediano [%]	Error Mediano [cm]
1	32.22	43.43	28.49	34.26
2	28.79	38.09	23.29	27.66
3	27.02	35.66	20.89	24.66
4	28.80	37.74	22.24	26.96
5	30.98	42.37	26.44	32.4
6	30.27	40.84	24.81	30.72
7	31.07	41.18	24.05	30.4

Tabla A3: Resultados de todos los experimentos llevados a cabo sobre la librería 3

Anexo V

En el anexo II se ha visto que el error de las reconstrucciones teóricos depende de la distancia entre cámaras o baseline b y la profundidad a analizar, de la siguiente forma: $\Delta Z = -\frac{d}{fb} Z_0^2$.

Si se quiere expresar el error en función del ángulo de paralaje α es necesario hacer la siguiente transformación: como se ve en la ilustración A2, la distancia X se puede expresar como $X = Z \cdot \tan\theta_1$ y como $X = Z \cdot \tan\theta_2 + b$.

Despejando la profundidad del punto P, Z, se obtiene $Z = \frac{b}{\tan\theta_1 - \tan\theta_2}$, y siendo que θ_1 y θ_2 son ángulos pequeños debido a la geometría de la cámara se pueden hacer las siguientes aproximaciones: $\tan\theta_1 \approx \theta_1$ y $\tan\theta_2 \approx \theta_2$. De esta forma, la profundidad Z queda expresada como $Z = \frac{b}{\theta_1 - \theta_2}$.

Analizando la geometría del triángulo $C_1 P C_2$, la suma de sus tres ángulos tiene que ser igual a 180 °, por lo que se obtiene que $\alpha = \theta_1 - \theta_2$, y por consiguiente $Z = \frac{b}{\alpha}$.

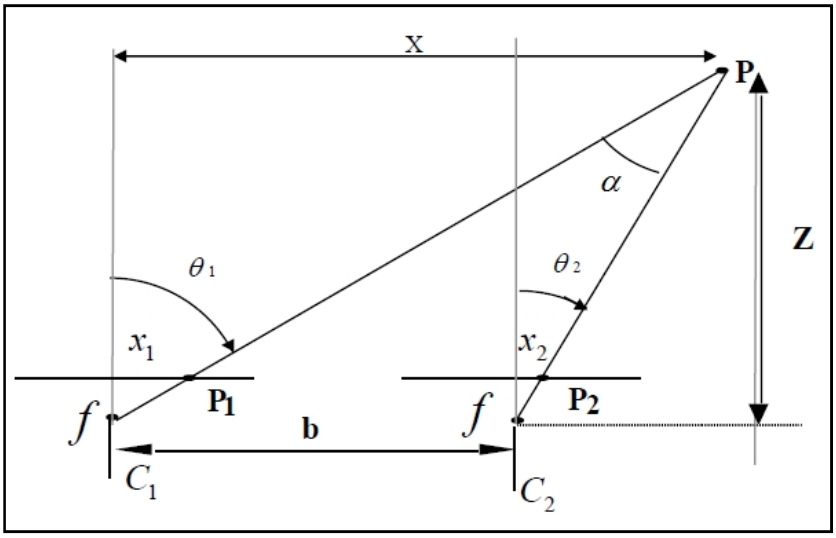

Ilustración A2: Esquema de visión en 2D (Martínez Montiel)

Teniendo en cuenta la anterior conclusión y haciendo un análisis de sensibilidad aproximado de la profundidad Z respecto al ángulo de paralaje α se llega a la siguiente conclusión: $\Delta Z = \frac{\sigma_0 \cdot \sqrt{2}}{\alpha^2} \cdot b$, siendo σ_0 el ángulo que abarca un píxel.

El error de la profundidad es proporcional a la distancia entre centros de cámaras b, e inversamente proporcional al cuadrado del ángulo de paralaje, α^2. (Martínez Montiel)

Anexo VI

En la memoria principal se expone el promedio de los resultados de todas las pruebas utilizadas para cada tamaño de parche. En este anexo, en las tablas A4 - A31 se encuentra el resultado de cada prueba realizada.

Librería 1: Parche 3 x 3 Par de imágenes	Error Medio [%]	Error Mediano [%]
1	19.77	9.3
2	18.18	7.93
3	20.04	8.93
4	19.81	9.06
5	16.95	7.81
6	24.88	8.41
7	23.52	10.61

Tabla A4: Resultados de todos los experimentos llevados a cabo sobre la librería 1 con parche 3 x 3

Librería 1: Parche 5 x 5 Par de imágenes	Error Medio [%]	Error Mediano [%]
1	18.29	7.93
2	16.51	6.31
3	18.81	7.73
4	18.36	7.51
5	15.20	6.24
6	22.92	6.39
7	21.61	9.32

Tabla A5: Resultados de todos los experimentos llevados a cabo sobre la librería 1 con parche 5 x 5

Librería 1: Parche 7 x 7 Par de imágenes	Error Medio [%]	Error Mediano [%]
1	16.69	6.43
2	14.92	4.90
3	17.54	6.39
4	16.99	5.93
5	13.65	4.97
6	21.04	5.17
7	19.55	8.37

Tabla A6: Resultados de todos los experimentos llevados a cabo sobre la librería 1 con parche 7 x 7

Librería 1: Parche 9 x 9 Par de imágenes	Error Medio [%]	Error Mediano [%]
1	15.58	5.51
2	13.93	4.20
3	16.62	5.47
4	16.00	5.06
5	12.68	4.20
6	19.88	4.48
7	18.22	7.70

Tabla A7: Resultados de todos los experimentos llevados a cabo sobre la librería 1 con parche 9 x 9

Librería 1: Parche 11 x 11 Par de imágenes	Error Medio [%]	Error Mediano [%]
1	14.67	4.85
2	13.15	3.86
3	15.74	4.88
4	15.09	4.48
5	11.93	3.80
6	18.97	3.97
7	17.12	7.26

Tabla A8: Resultados de todos los experimentos llevados a cabo sobre la librería 1 con parche 11 x 11

Librería 1: Parche 13 x 13 Par de imágenes	Error Medio [%]	Error Mediano [%]
1	14.00	4.44
2	12.62	3.66
3	15.03	4.50
4	14.27	4.12
5	11.35	3.58
6	18.31	3.64
7	16.50	7.00

Tabla A9: Resultados de todos los experimentos llevados a cabo sobre la librería 1 con parche 13 x 13

Librería 1: Parche 15 x 15 Par de imágenes	Error Medio [%]	Error Mediano [%]
1	13.49	4.14
2	12.14	3.53
3	14.53	4.22
4	13.71	3.93
5	10.89	3.44
6	17.70	3.39
7	15.46	6.67

Tabla A10: Resultados de todos los experimentos llevados a cabo sobre la librería 1 con parche 15 x 15

Librería 1: Parche 17 x 17 Par de imágenes	Error Medio [%]	Error Mediano [%]
1	13.04	3.97
2	11.83	3.51
3	14.17	4.09
4	13.33	3.85
5	10.44	3.39
6	17.28	3.24
7	14.90	6.50

Tabla A11: Resultados de todos los experimentos llevados a cabo sobre la librería 1 con parche 17 x 17

Librería 1: Parche 19 x 19 Par de imágenes	Error Medio [%]	Error Mediano [%]
1	12.73	3.91
2	11.64	3.49
3	13.89	4.03
4	13.16	3.80
5	10.17	3.37
6	16.86	3.14
7	14.40	6.35

Tabla A12: Resultados de todos los experimentos llevados a cabo sobre la librería 1 con parche 19 x 19

Librería 2: Parche 3 x 3 Par de imágenes	Error Medio [%]	Error Mediano [%]
1	30.02	17.61
2	26.48	14.95
3	21.98	10.95
4	34.53	18.21
5	30.86	15.90
6	29.27	15.80
7	34.11	18.73

Tabla A13: Resultados de todos los experimentos llevados a cabo sobre la librería 2 con parche 3 x 3

Librería 2: Parche 5 x 5 Par de imágenes	Error Medio [%]	Error Mediano [%]
1	28.04	15.52
2	24.09	13.44
3	19.61	8.93
4	32.42	16.05
5	29.00	13.88
6	27.83	14.25
7	32.25	17.13

Tabla A14: Resultados de todos los experimentos llevados a cabo sobre la librería 2 con parche 5 x 5

Librería 2: Parche 7 x 7 Par de imágenes	Error Medio [%]	Error Mediano [%]
1	25.75	13.38
2	21.73	12.03
3	17.16	7.34
4	30.08	13.55
5	27.08	11.73
6	25.99	12.15
7	30.20	15.08

Tabla A15: Resultados de todos los experimentos llevados a cabo sobre la librería 2 con parche 7 x 7

Librería 2: Parche 9 x 9 Par de imágenes	Error Medio [%]	Error Mediano [%]
1	24.27	12.02
2	20.23	11.08
3	15.67	6.56
4	28.44	12.10
5	25.90	10.29
6	24.74	10.88
7	29.13	13.75

Tabla A16: Resultados de todos los experimentos llevados a cabo sobre la librería 2 con parche 9 x 9

Librería 2: Parche 11 x 11 Par de imágenes	Error Medio [%]	Error Mediano [%]
1	23.10	11.06
2	19.19	10.44
3	14.50	6.08
4	27.36	11.04
5	24.94	9.35
6	23.84	10.02
7	27.83	12.60

Tabla A17: Resultados de todos los experimentos llevados a cabo sobre la librería 2 con parche 11 x 11

Librería 2: Parche 13 x 13 Par de imágenes	Error Medio [%]	Error Mediano [%]
1	22.01	10.31
2	18.44	9.99
3	13.74	5.83
4	26.35	10.18
5	24.30	8.75
6	23.33	9.48
7	26.98	11.67

Tabla A18: Resultados de todos los experimentos llevados a cabo sobre la librería 2 con parche 13 x 13

Librería 2: Parche 15 x 15 Par de imágenes	Error Medio [%]	Error Mediano [%]
1	21.16	9.65
2	17.99	9.66
3	13.13	5.59
4	25.59	9.50
5	23.87	8.23
6	22.83	9.00
7	26.23	11.01

Tabla A19: Resultados de todos los experimentos llevados a cabo sobre la librería 2 con parche 15 x 15

Librería 2: Parche 17 x 17 Par de imágenes	Error Medio [%]	Error Mediano [%]
1	20.34	9.15
2	17.44	9.31
3	12.66	5.47
4	24.82	8.97
5	23.51	7.75
6	22.41	8.59
7	25.48	10.43

Tabla A20: Resultados de todos los experimentos llevados a cabo sobre la librería 2 con parche 17 x 17

Librería 2: Parche 19 x 19 Par de imágenes	Error Medio [%]	Error Mediano [%]
1	19.59	8.80
2	16.97	9.08
3	12.28	5.35
4	24.19	8.48
5	23.20	7.39
6	22.11	8.34
7	24.82	9.91

Tabla A21: Resultados de todos los experimentos llevados a cabo sobre la librería 2 con parche 19 x 19

Librería 3: Parche 1 x 1 Par de imágenes	Error Medio [%]	Error Mediano [%]
1	32.22	28.49
2	28.79	23.29
3	27.02	20.89
4	28.80	22.24
5	30.98	26.44
6	30.27	24.81
7	31.07	24.05

Tabla A22: Resultados de todos los experimentos llevados a cabo sobre la librería 3 con parche 1 x 1

Librería 3: Parche 3 x 3 Par de imágenes	Error Medio [%]	Error Mediano [%]
1	32.57	28.69
2	28.17	21.98
3	25.58	18.67
4	28.78	21.32
5	31.07	26.19
6	30.00	23.81
7	31.78	23.83

Tabla A23: Resultados de todos los experimentos llevados a cabo sobre la librería 3 con parche 3 x 3

Librería 3: Parche 5 x 5 Par de imágenes	Error Medio [%]	Error Mediano [%]
1	32.30	28.69
2	27.25	21.04
3	24.38	17.17
4	28.32	20.65
5	30.65	25.98
6	29.40	23.01
7	31.77	23.43

Tabla A24: Resultados de todos los experimentos llevados a cabo sobre la librería 3 con parche 5 x 5

Librería 3: Parche 7 x 7 Par de imágenes	Error Medio [%]	Error Mediano [%]
1	32.06	28.69
2	26.30	20.09
3	23.05	15.60
4	27.78	19.94
5	30.02	25.71
6	28.77	22.30
7	31.57	22.98

Tabla A25: Resultados de todos los experimentos llevados a cabo sobre la librería 3 con parche 7 x 7

Librería 3: Parche 9 x 9 Par de imágenes	Error Medio [%]	Error Mediano [%]
1	31.87	28.69
2	25.53	19.21
3	22.13	14.41
4	27.16	19.28
5	29.51	25.40
6	28.22	21.71
7	31.19	22.45

Tabla A26: Resultados de todos los experimentos llevados a cabo sobre la librería 3 con parche 9 x 9

Librería 3: Parche 11 x 11 Par de imágenes	Error Medio [%]	Error Mediano [%]
1	31.71	28.79
2	25.01	18.42
3	21.47	13.49
4	26.66	18.77
5	29.13	25.14
6	27.72	21.30
7	30.73	21.99

Tabla A27: Resultados de todos los experimentos llevados a cabo sobre la librería 3 con parche 11 x 11

Librería 3: Parche 13 x 13 Par de imágenes	Error Medio [%]	Error Mediano [%]
1	31.62	28.89
2	24.52	17.73
3	20.82	12.58
4	26.34	18.30
5	28.93	25.02
6	27.22	21.02
7	30.30	21.63

Tabla A28: Resultados de todos los experimentos llevados a cabo sobre la librería 3 con parche 13 x 13

Librería 3: Parche 15 x 15 Par de imágenes	Error Medio [%]	Error Mediano [%]
1	31.06	28.24
2	24.06	17.04
3	20.33	11.91
4	26.16	17.90
5	28.87	24.98
6	26.82	20.62
7	29.85	21.26

Tabla A29: Resultados de todos los experimentos llevados a cabo sobre la librería 3 con parche 15 x 15

Librería 3: Parche 17 x 17 Par de imágenes	Error Medio [%]	Error Mediano [%]
1	31.18	28.42
2	23.79	16.58
3	20.02	11.41
4	26.01	17.74
5	28.84	24.92
6	26.26	20.26
7	29.29	20.80

Tabla A30: Resultados de todos los experimentos llevados a cabo sobre la librería 3 con parche 17 x 17

Librería 3: Parche 19 x 19 Par de imágenes	Error Medio [%]	Error Mediano [%]
1	31.24	28.50
2	23.57	16.16
3	19.70	10.98
4	25.83	17.52
5	28.86	24.85
6	26.20	20.08
7	29.26	20.79

Tabla A31: Resultados de todos los experimentos llevados a cabo sobre la librería 3 con parche 19 x 19

Anexo VII

Para corroborar que lo expuesto en la parte principal de la memoria se cumple para cualquier caso es necesario realizar las mismas pruebas sobre diferentes imágenes. En este anexo en la tabla A32 se encuentran todas las pruebas realizadas para determinar el número de fotos óptimo para las reconstrucciones.

Número de fotos / Conjunto de imágenes	1		10		20		30		40		50		60	
	Error Medio [%]	Error Mediano [%]	Error Medio [%]	Error Mediano [%]	Error Medio [%]	Error Mediano [%]	Error Medio [%]	Error Mediano [%]	Error Medio [%]	Error Mediano [%]	Error Medio [%]	Error Mediano [%]	Error Medio [%]	Error Mediano [%]
1	20.24	12.37	27.90	19.70	28.04	4.36	14.01	2.68	9.75	2.36	7.83	2.16	6.89	2.04
2	62.87	8.91	22.75	4.70	9.52	2.90	6.78	2.49	7.09	2.23	7.61	2.16	6.79	2.02
3	64.86	11.02	10.36	3.25	6.85	2.61	11.32	2.62	6.19	2.22	6.22	2.12	6.02	2.05
4	59.31	10.83	14.79	3.24	9.68	2.57	6.43	2.45	5.76	2.23	5.81	2.09	6.95	2.05
5	56.80	10.84	14.26	3.26	8.57	2.60	6.02	2.21	5.40	2.11	5.64	2.12	5.98	2.00
6	50.50	9.13	10.54	3.20	7.42	2.46	7.36	2.28	5.36	2.13	5.29	2.10	5.08	2.06

Tabla A32: Resultados de todas las pruebas realizadas para diferente número de imágenes de apoyo

Anexo VIII

En este anexo, en la tabla A33, se encuentran todas las pruebas realizadas para determinar el número de fotos óptimo para las reconstrucciones en las que se ha utilizado suavizado. Al realizar numerosas pruebas se puede confirmar con cierta seguridad que lo que ocurre en éstas pasará en todas las pruebas.

Número de fotos / Conjunto de imágenes	1		10		20		30		40		50		60	
	Error Medio [%]	Error Mediano [%]	Error Medio [%]	Error Mediano [%]	Error Medio [%]	Error Mediano [%]	Error Medio [%]	Error Mediano [%]	Error Medio [%]	Error Mediano [%]	Error Medio [%]	Error Mediano [%]	Error Medio [%]	Error Mediano [%]
1	23.56	3.56	26.26	3.14	12.02	2.02	4.79	1.72	3.30	1.74	3.05	1.67	2.70	1.55
2	43.06	2.98	8.35	2.72	6.70	1.94	3.18	1.97	2.86	1.74	2.93	1.62	2.67	1.54
3	46.10	3.49	5.30	2.16	4.35	2.09	2.73	1.78	2.93	1.86	2.79	1.66	2.73	1.60
4	38.31	3.78	5.81	2.33	4.12	2.22	3.08	1.70	2.93	1.89	2.72	1.65	2.78	1.55
5	35.41	3.71	5.59	1.95	3.32	1.96	2.73	1.78	2.64	1.70	2.68	1.68	2.67	1.51
6	29.66	3.13	3.70	1.71	3.63	1.86	3.08	1.70	2.71	1.75	2.72	1.75	2.73	1.61

Tabla A33: Resultados de todas las pruebas realizadas con suavizado para diferente número de imágenes de apoyo

Anexo IX

En este anexo, en las tablas A34 - A43, se encuentran los resultados de todas las pruebas realizadas con 50 imágenes de apoyo, suavizado y parches.

Parche 1 x 1	Error Medio [%]	Error Mediano [%]
Prueba 1	2.70	1.55
Prueba 2	2.67	1.54
Prueba 3	2.73	1.60
Prueba 4	2.78	1.55
Prueba 5	2.67	1.51
Prueba 6	2.73	1.61

Tabla A34: Resultados de las pruebas realizadas con 50 imágenes de apoyo, suavizado y parches de 1 x 1 píxeles

Parche 3 x 3	Error Medio [%]	Error Mediano [%]
Prueba 1	2.55	1.53
Prueba 2	2.53	1.56
Prueba 3	2.57	1.62
Prueba 4	2.56	1.59
Prueba 5	2.54	1.55
Prueba 6	2.55	1.57

Tabla A35: Resultados de las pruebas realizadas con 50 imágenes de apoyo, suavizado y parches de 3 x 3 píxeles

Parche 5 x 5	Error Medio [%]	Error Mediano [%]
Prueba 1	2.48	1.53
Prueba 2	2.47	1.56
Prueba 3	2.53	1.63
Prueba 4	2.51	1.57
Prueba 5	2.47	1.55
Prueba 6	2.49	1.59

Tabla A36: Resultados de las pruebas realizadas con 50 imágenes de apoyo, suavizado y parches de 5 x 5 píxeles

Parche 7 x 7	Error Medio [%]	Error Mediano [%]
Prueba 1	2.42	1.56
Prueba 2	2.40	1.53
Prueba 3	2.47	1.63
Prueba 4	2.43	1.6
Prueba 5	2.41	1.57
Prueba 6	2.45	1.54

Tabla A37: Resultados de las pruebas realizadas con 50 imágenes de apoyo, suavizado y parches de 7 x 7 píxeles

Parche 9 x 9	Error Medio [%]	Error Mediano [%]
Prueba 1	2.37	1.52
Prueba 2	2.37	1.55
Prueba 3	2.42	1.62
Prueba 4	2.42	1.56
Prueba 5	2.39	1.6
Prueba 6	2.36	1.52

Tabla A38: Resultados de las pruebas realizadas con 50 imágenes de apoyo, suavizado y parches de 9 x 9 píxeles

Parche 11 x 11	Error Medio [%]	Error Mediano [%]
Prueba 1	2.37	1.53
Prueba 2	2.34	1.55
Prueba 3	2.39	1.61
Prueba 4	2.37	1.56
Prueba 5	2.33	1.57
Prueba 6	2.41	1.58

Tabla A39: Resultados de las pruebas realizadas con 50 imágenes de apoyo, suavizado y parches de 11 x 11 píxeles

Parche 13 x 13	Error Medio [%]	Error Mediano [%]
Prueba 1	2.34	1.54
Prueba 2	2.33	1.56
Prueba 3	2.37	1.61
Prueba 4	2.35	1.57
Prueba 5	2.34	1.59
Prueba 6	2.36	1.55

Tabla A40: Resultados de las pruebas realizadas con 50 imágenes de apoyo. suavizado y parches de 13 x 13 píxeles

Parche 15 x 15	Error Medio [%]	Error Mediano [%]
Prueba 1	2.30	1.53
Prueba 2	2.31	1.56
Prueba 3	2.35	1.62
Prueba 4	2.32	1.57
Prueba 5	2.30	1.60
Prueba 6	2.34	1.54

Tabla A41: Resultados de las pruebas realizadas con 50 imágenes de apoyo. suavizado y parches de 15 x 15 píxeles

Parche 17 x 17	Error Medio [%]	Error Mediano [%]
Prueba 1	2.33	1.55
Prueba 2	2.33	1.57
Prueba 3	2.37	1.62
Prueba 4	2.34	1.58
Prueba 5	2.33	1.56
Prueba 6	2.36	1.60

Tabla A42: Resultados de las pruebas realizadas con 50 imágenes de apoyo. suavizado y parches de 17 x 17 píxeles

Parche 19 x 19	Error Medio [%]	Error Mediano [%]
Prueba 1	2.35	1.56
Prueba 2	2.35	1.57
Prueba 3	2.39	1.62
Prueba 4	2.36	1.58
Prueba 5	2.35	1.56
Prueba 6	2.37	1.60

Tabla A43: Resultados de las pruebas realizadas con 50 imágenes de apoyo. suavizado y parches de 19 x 19 píxeles