



Escuela de
Ingeniería y Arquitectura
Universidad Zaragoza



Universidad
Zaragoza

Proyecto Final de Carrera
Ingeniería Industrial
Curso 2012-2013

**Reconstrucción 3D en tiempo real
a partir de imágenes de endoscopio**

Rubén Lumbreras Roche

Director: José María Martínez Montiel

Departamento de Informática e Ingeniería de Sistemas
Escuela de Ingeniería y Arquitectura
Universidad de Zaragoza

Marzo 2013

Agradecimientos

- Agradecimiento a Oscar García Grasa y al equipo de cirujanos del Hospital Clínico Lozano Blesa (Ismael Gil, Ernesto Bernal y Santiago Casado) por las secuencias in-vivo de operaciones de hernia ventral en humanos.
- A George Klein por su generosidad al permitir el empleo del software "PTAM" para uso de investigación.
- Y una mención especial a José María Martínez Montiel, profesor titular de la universidad de Zaragoza, por dirigir todo este proyecto y por el gran interés puesto en él.

Reconstrucción 3D en tiempo real a partir de imágenes de endoscopio

Resumen

En la actualidad la intervención con cirugía laparoscópica se ha convertido en una técnica habitual en las diferentes especialidades quirúrgicas. El empleo de esta técnica en sustitución a la cirugía convencional, que se realiza mediante una incisión en el abdomen, presenta grandes beneficios al ser una técnica menos invasiva.

El uso que se le da a las imágenes que proporciona el endoscopio durante la intervención se limita a su visualización por el monitor, para ser inmediatamente desechadas. El objetivo es analizar de qué manera los recientes resultados de visión por computador desarrollados para la robótica móvil pueden explotarse en la cirugía laparoscópica.

Se ha focalizado el trabajo en los métodos de construcción de mapas simultáneos a la localización a partir de imágenes porque presenta gran similitud entre la robótica y la cirugía: una cámara móvil (endoscopio), observa una escena (cavidad abdominal) por primera vez, siendo el objetivo construir, en tiempo real a frecuencia de video, un mapa de la escena y estimar simultáneamente la localización de la cámara (endoscopio) respecto del mapa construido. Una vez obtenido este resultado pueden realizarse medidas geométricas sobre la escena, dar soporte a anotaciones de realidad aumentada. También permite mostrar una reconstrucción de la escena que cubre la unión de todas las observaciones parciales que ha detectado el endoscopio consiguiendo así aumentar el campo de vista del endoscopio.

El empleo de técnicas de SLAM visual permiten conocer la trayectoria seguida por la cámara durante toda la toma de imágenes así como obtener el mapa que conforma la estructura de la escena. El problema de estos algoritmos es que en la práctica, la estimación de la trayectoria de la cámara es correcta pero la estimación del modelo 3D de la escena es errónea debida a la aparición de puntos espurios, puntos 3D erróneamente estimados en el mapa.

Este proyecto se centra en mejorar la calidad del mapa, por una parte mediante la reducción de la tasa de espurios en tiempo real y por otra mediante el procesamiento del mapa a posteriori para una eliminación final de espurios y la construcción de un mapa de apariencia fotorrealista.

Esta reconstrucción se realiza aplicando técnicas de mallado triangular sobre el mapa semidenso estimado de manera que aplicando un texturizado sobre la superficie creada se genere la reconstrucción fotorrealista de la escena. La presencia de puntos espurios arruina este modelo por lo que es preciso obtener un mapa 3D con baja densidad de espurios.

Los métodos propuestos son validados sobre secuencias reales de eventroplastia por laparoscopia in-vivo en humanos. Pudiendo concluir que las técnicas de SLAM pueden aplicarse con éxito en la cirugía laparoscópica.

Este trabajo se enmarca dentro del proyecto de investigación del Plan Nacional: "DPI2009-07310. Visión 3D robusta y en tiempo real. Aplicación de la realidad aumentada en cirugía con endoscopia." desarrollado por el Grupo de Robótica, Percepción y Tiempo Real de la Universidad de Zaragoza que tiene como uno de sus objetivos la aplicación de técnicas de visión por computador a la cirugía con endoscopia para la reparación de hernias ventrales.

Índice general

Agradecimientos	I
Resumen	III
1. Introducción	1
1.1. Estado del Arte	1
1.2. Motivación y Objetivos	2
1.3. Estructura de la memoria	3
2. Algoritmo de SLAM visual: PTAM	5
2.1. Seguimiento de la posición de la cámara: Tracking	5
2.2. Estimación del mapa de la escena: Mapping	6
2.3. Resultados obtenidos	7
3. Eliminación de espurios a posteriori	9
3.1. Eliminación según Error de Reproyección	9
3.2. Eliminación por ángulo de paralaje	10
3.3. Eliminación por Error de Mínima Mediana en profundidad	11
4. Eliminación de espurios en tiempo real	15
4.1. Sintonización del algoritmos de eliminación de puntos con Error de Reproyección excesivo	15
4.2. Eliminación de puntos no reobservables en frames	17
5. Reconstrucción fotorealista de la escena	19
5.1. Algoritmo de triangulación implementado	20
6. Validación y Resultados	25
6.1. Eliminación de espurios a posteriori	26
6.2. Eliminación de espurios en Tiempo Real	29
6.3. Reconstrucciones 3D	32
6.4. Tiempos de cómputo	35

7. Conclusiones y Trabajo Futuro	37
7.1. Conclusiones	37
7.2. Líneas de investigación futuras	38
Bibliografía	41
A. Resultados y Reconstrucciones	43
A.1. Secuencia 2011-02-28	44
A.2. Secuencia 2011-04-04	46
A.3. Secuencia 2011-09-27	48
A.4. Secuencia 2011-10-03	50
A.5. Secuencia 2011-10-17	52
A.6. Secuencia 2011-10-18	54
A.7. Secuencia 2011-11-29	56
A.8. Secuencia 2012-01-23	58
A.9. Secuencia 2012-02-06	60
A.10. Secuencia 2012-03-06	62
A.11. Secuencia 2012-05-21(1)	64
A.12. Secuencia 2012-05-21(2)	66
A.13. Secuencia 2012-06-04	68
A.14. Secuencia 2012-07-02(1)	69
A.15. Secuencia 2012-07-02(2)	71
Lista de figuras	73
Lista de tablas	76

Capítulo 1

Introducción

1.1. Estado del Arte

La visión por computador es la disciplina que tiene como objetivo la tarea de programar un computador para que "entienda" una escena o interprete las características de una imagen. Abarca muchos campos de trabajo como la detección, localización y reconocimiento de objetos en imágenes, seguimiento de objetos o mapeo de una escena para la generación de modelos tridimensionales.

Existen variedad de algoritmos que permiten la obtención de modelos 3D a partir de secuencias de imágenes de manera secuencial. Agarwal [30] presenta la posibilidad de obtener la reconstrucción 3D de ciudades enteras mediante el empleo de miles de cientos de imágenes desordenadas de la misma. Furukawa [31] plantean la posibilidad de obtener modelos densos de objetos y lugares también a partir de un conjunto de imágenes de la escena aplicando técnicas de emparejamiento sobre imágenes y su expansión sobre píxeles cercanos, dotando al algoritmo de la capacidad de discernir puntos outliers en la escena.

En concreto, este proyecto se va a centrar en el empleo de técnicas de SLAM visual (Simultaneous camera Localization And Mapping) para la generación de modelos 3D de la escena a partir de una secuencia real de cirugía de reparación de hernia ventral en humanos.

Será en los trabajos de Stoyanow [23] y Mourgues [24] donde se puedan ver algunos de los resultados de aplicar este tipo de técnicas de visión por computador sobre secuencias médicas para la obtención de modelos 3D haciendo uso de endoscopios monoculares. Koppel [22], estima la profundidad de la escena en una operación colonoscópica a partir de la disparidad entre los puntos en las diferentes imágenes de la secuencia generando un modelo 3D preciso de la escena.

En la actualidad existen diversos softwares libres al alcance del usuario que permiten realizar reconstrucciones fotorealistas no secuenciales a partir de una secuencia de imágenes. Un ejemplo de este tipo de softwares es 123D Catch desarrollado por Autodesk.

A partir de una secuencia de imágenes de la escena que se desea reconstruir, este software permite llevar a cabo un modelizado de la misma, generando una malla densa texturizada que puede ser visualizada (fig. 1.1). Además permite realizar mediciones sobre la escena una vez se ha realizado el modelo 3D.

Como se puede ver en los resultados obtenidos, es posible generar un modelo 3D fidedigno a partir de una secuencia de imágenes de una operación real de cirugía de reparación de hernia ventral.

Sin embargo, este tipo de programas y algoritmos no cubren las necesidades que se requieren en el empleo de este tipo de técnicas en cirugía laparoscópica. Todos estos trabajos como se ha visto permiten obtener modelos 3D pero no lo hacen en tiempo real a frecuencia de vídeo.

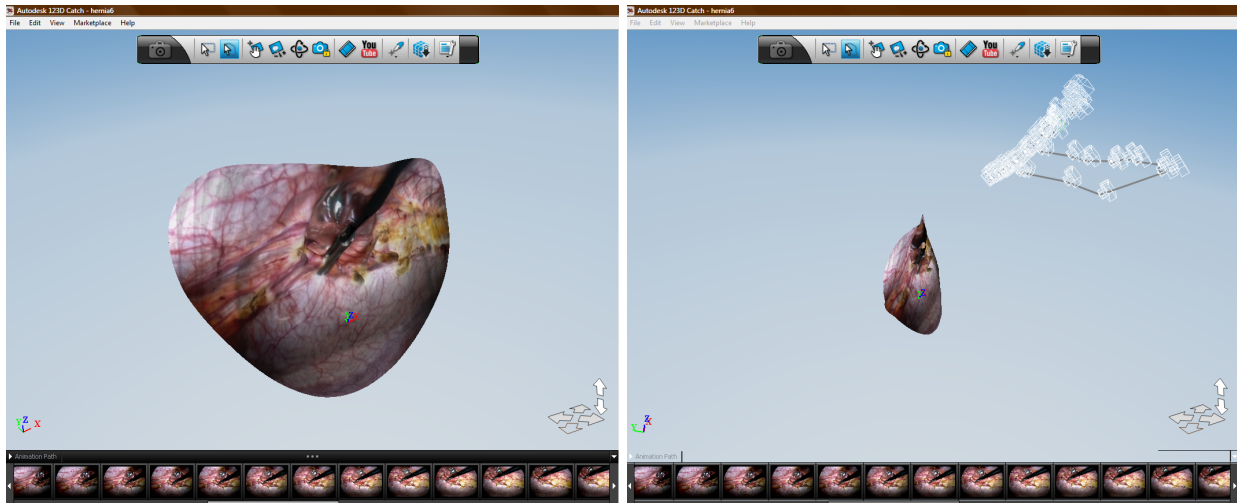


Figura 1.1: Modelo fotorealista obtenido aplicando el software 123D Catch.

O. García [8] presenta un algoritmo que permite a partir de una secuencia médica real tomada en quirófano con endoscopio obtener modelos 3D y aplicar realidad aumentada, todo ello en tiempo real. Los mapas que se manejan en este trabajo son del orden de las decenas de puntos.

El objetivo de este proyecto es aplicar este tipo técnicas de visión por computador en tiempo real recientemente desarrollados en el campo de la robótica para la obtención de modelos 3D que permitan reconstruir la escena a partir de una secuencia médica de endoscopio.

1.2. Motivación y Objetivos

La técnica de SLAM visual utiliza las imágenes proporcionadas por una cámara de 6 grados de libertad para estimar la localización de la cámara en relación con el entorno en el que se encuentra y al mismo tiempo estimar un mapa de la escena, todo ello en tiempo real a frecuencia de vídeo.

Actualmente, todos los trabajos relacionados con técnicas de visión slam están más enfocados en la correcta estimación de la trayectoria de la cámara que en la correcta estimación del mapa 3D que estime con precisión la estructura de la escena.

En este proyecto concreto, se va a trabajar con el algoritmo PTAM (Parallel Tracking And Mapping)[1], uno de los recientes algoritmos que mejores resultados están dando en SLAM visual.

La clave de estos algoritmos de SLAM visual es la estimación de puntos homólogos, correspondencias, entre las imágenes de la secuencia. A partir de un conjunto de correspondencias bien estimadas, el problema de estimar la escena y la trayectoria de la cámara se simplifica a resolver un problema de optimización no lineal bien conocido.

PTAM por su parte emplea el algoritmo de optimización no lineal de Levenbeg-Marquardt que tolera la existencia de puntos espurios (puntos erróneamente estimados en el mapa) mediante el empleo de funciones robustas [4]. El resultado final obtenido de la aplicación de estas técnicas es que la trayectoria que sigue la cámara es correcta pero la estimación del modelo 3D de la escena no, debido a la incursión de puntos espurios en el mapa.

El objetivo principal en el que se centra todo este proyecto es en mejorar la calidad del mapa, reduciendo la tasa de puntos espurios que perturben la escena para la obtención del modelo 3D fotorealista.

Por una parte, la consecución de este objetivo se centrará en la creación de un algoritmo que lleve a cabo una detección y eliminación de espurios mediante el procesado a posteriori del mapa

estimado por PTAM. De esta manera, se asegurará la existencia de un mapa libre de espurios que permita obtener un modelo 3D de la escena lo más preciso posible.

Por otra parte, se presentarán una serie de mejoras relacionadas con la detección y eliminación de espurios en tiempo real y a frecuencia de vídeo que permitan reducir la tasa de espurios todo ello durante la ejecución del algoritmo PTAM.

Así, el resultado final permitirá obtener una reconstrucción 3D fotorealista de la escena que represente la cavidad abdominal del paciente de la manera más fidedigna posible.

PTAM genera un mapa semi-denso de puntos 3D (del orden del millar de ptos) que estima la forma y dimensión de la cavidad. A partir de este mapa se propone crear un modelo 3D fotorealista basado en técnicas de triangulación Delaunay, de manera que el cirujano pueda visualizar la cavidad completa y así facilitar su comprensión.

Los resultados serán validados sobre 15 secuencias correspondientes a cirugías de reparación de hernia en humanos. Mostrando así las ventajas de los algoritmos propuestos. La obtención de unos resultados favorables en cuanto a reducción de la tasa de espurios y generación de modelos 3D se refiere, permitirá decir que los algoritmos empleados son válidos y pueden generalizarse a cualquier tipo de secuencias de laparoscopia cumplan las mismas condiciones en la escena.

1.3. Estructura de la memoria

En el capítulo 2 se realiza un análisis del algoritmo PTAM. Algoritmo slam empleado para la obtención del modelo 3D de la cavidad abdominal.

Con el fin de garantizar en todo momento un mapa final libre de espurios, en el capítulo 3 se propone un algoritmo de detección y eliminación que realice un post-procesado del mapa obtenido por PTAM de manera que se asegure un mapa final libre de espurios con el fin de obtener el modelo 3D de la escena lo más preciso posible.

El objetivo principal es conseguir que esta eliminación se lleve a cabo en tiempo real durante la ejecución del algoritmo PTAM, por ello en el capítulo 4 se proponen una serie de mejoras que reduzcan la tasa de espurios en el mapa estimado.

En el capítulo 5 se detalla el funcionamiento del algoritmo implementado para la obtención de modelos 3D fotorealistas a partir del mapa procesado por el algoritmo PTAM y el conjunto de imágenes seleccionadas por el propio algoritmo de la secuencia de reparación de hernia ventral.

Los resultados obtenidos para las mejoras propuestas en la detección y eliminación de espurios y las reconstrucciones 3D para las distintas secuencias sobre las que han sido validadas se presentan y analizan en el capítulo 6.

Por último, las conclusiones obtenidas de este proyecto y futuras líneas de desarrollo se tratan en el capítulo 7.

Capítulo 2

Algoritmo de SLAM visual: PTAM

PTAM es un algoritmo que permite estimar de manera simultánea la posición de la cámara y la estructura de una escena desconocida. La característica que diferencia este algoritmo del resto de algoritmos de SLAM (Simultaneous Localization And Mapping) es la separación de las tareas de Tracking y Mapping en dos procesos paralelos. El Tracking se centrará en la obtención de una trayectoria robusta de la cámara y el Mapping en la obtención de un mapa 3D a partir de los frames de la secuencia previamente analizados.

El objetivo de este capítulo es detallar el funcionamiento del algoritmo que permite obtener el mapa de la escena para su posterior reconstrucción 3D. Para ello se van a explicar en rasgos generales el funcionamiento de ambos procesos (Tracking y Mapping).

2.1. Seguimiento de la posición de la cámara: Tracking

La tarea de tracking se encarga de mantener en tiempo real la estimación de la posición de la cámara relativa al mapa 3D de la escena. Este mapa 3D ha sido previamente inicializado empleando el algoritmo estéreo de 5 puntos [29].

El método a proceder durante la tarea de tracking es:

1. Proyección del mapa estimado sobre el frame.
2. Búsqueda de los puntos del mapa vistos en el frame.
3. Estimación de la posición final del frame a partir de los emparejamientos encontrados.

El mapa 3D es transformado en coordenadas homogéneas $p_{jW} = (x_{jW} \ y_{jW} \ z_{jW} \ 1)^T$. Para proyectar los puntos 3D sobre las imágenes se llevará a cabo una transformación desde la posición mundo (frame inicial W) hasta la posición de la cámara (C) en la que se van a comprobar los errores. La matriz de transformación que representa la posición de la cámara se denota (T_{CW}).

$$p_{jC} = T_{CW}p_{jW} \quad (2.1)$$

El modelo de proyección de cámara calibrada permite transformar el punto de coordenadas 3D a coordenadas en la imagen:

$$\begin{pmatrix} u_i \\ v_i \end{pmatrix} = FOVProj(p_{jC}) \quad (2.2)$$

El modelo de distorsión radial de FOV [2] utilizado por PTAM es el que se utiliza para este mismo proceso:

$$FOVProj \begin{pmatrix} x_i \\ y_i \\ z_i \end{pmatrix} = \begin{pmatrix} u_0 \\ v_0 \end{pmatrix} + \begin{bmatrix} f_u & 0 \\ 0 & f_v \end{bmatrix} \frac{r'}{r} \begin{pmatrix} \frac{x}{z} \\ \frac{y}{z} \end{pmatrix} \quad (2.3)$$

$$r = \sqrt{\frac{x^2 + y^2}{z^2}} \quad r' = \frac{1}{w} \arctan \left(2r \tan \frac{w}{2} \right) \quad (2.4)$$

donde la distancia focal (f_u, f_v) , principal point (u_0, v_0) y la distorsión (w) de la cámara han sido estimadas y calibradas antes de las ejecuciones.

Para cada KeyFrame (KF) se obtienen los puntos FAST-09 [14] característicos salientes de la imagen. De tal manera que para cada punto del mapa, exista un punto homólogo detectado por el algoritmo.

La estimación de la localización del frame se lleva a cabo mediante la minimización de la función objetivo del error de reproyección de los puntos del mapa y sus correspondientes puntos homólogos (ec. 2.5). Si para alguno de los puntos no se ha encontrado emparejamiento, este punto no intervendrá en la estimación de la localización del frame.

$$\mu' = \underset{\mu}{\operatorname{argmin}} \sum_{j \in S} \operatorname{Obj} \left(\frac{|e_j|}{\sigma_j}, \sigma_T \right) \quad (2.5)$$

$\operatorname{Obj}(\cdot, \sigma_T)$ es la función objetivo de doble peso de Tukey [3] y σ_T un estimador de mediana de la desviación estándar de los residuos.

El cálculo del residuo (error de reproyección) entre un punto 3D y sus correspondientes emparejamientos se computa de la siguiente manera:

$$e_j = \begin{pmatrix} u_m \\ v_m \end{pmatrix} - FOVProj \begin{pmatrix} x_i \\ y_i \\ z_i \\ 1 \end{pmatrix} \quad (2.6)$$

2.2. Estimación del mapa de la escena: Mapping

La tarea de mapping se encarga estimar el mapa 3D de la escena, mediante un continuo refinamiento del mapa inicialmente estimado mediante técnicas estéreo y su correspondiente expansión a partir de los nuevos keyframes añadidos por el tracking.

El ajuste del mapa (Bundle Adjustment) se realiza de la misma manera que en el tracking. Se busca minimizar la función robusta objetivo del error de reproyección pero en esta ocasión el ajuste se realizará tanto de las cámaras como de la estimación de la escena.

$$\{\{\mu_2 \dots \mu_N\}, \{p'_1, p'_M\}\} = \underset{\{\mu\}, \{p\}}{\operatorname{argmin}} \sum_{i=1}^N \sum_{j \in S_i} \operatorname{Obj} \left(\frac{|e_j|}{\sigma_j}, \sigma_T \right) \quad (2.7)$$

Además del empleo del M-estimador de Tukey, se utiliza Levenberg-Marquardt (como se describe en su apéndice 6.6 de [4]) en el bundle adjustment global con la que se persigue estimar la localización de la cámara y los puntos 3D minimizando la función del error de reproyección robusta.

El tiempo de cómputo que supone un ajuste global de toda la escena, plantea la idea de realizar un ajuste local empleando únicamente 5 keyframes. El último de los keyframes incluidos al conjunto junto con sus vecinos más cercanos permitirá un ajuste local de la escena y del mapa antes de proceder a un ajuste global.

Cuando el Bundle Adjustment ha concluido y no hay nuevos keyframes que incluir al conjunto para su ajuste, el proceso tiene tiempo de buscar nuevos emparejamientos. La creación de un nuevo punto se llevará a cabo mediante correspondencia entre keyframes usando búsqueda epipolar. Este nuevo punto comenzará siendo visto por únicamente 2 keyframes y se posteriormente se procederá a buscar en el resto, de forma que se consiga un mayor número de emparejamientos que permitan un ajuste del mapa mejor.

2.3. Resultados obtenidos

Se testeó el algoritmo sobre una de las secuencias reales de reparación de hernia ventral. Los datos que se obtienen del algoritmo PTAM son: un mapa de puntos 3D de la escena y una selección de Keyframes de los que se conoce su posición con respecto a la escena y los correspondientes puntos homólogos del mapa 3D que han sido vistos en cada uno de los KFs.

Para la prueba se utilizó la secuencia (b) de la fig. 6.1 y en la imagen siguiente (fig. 2.1) se pueden ver alguno de los keyframes obtenidos por PTAM.

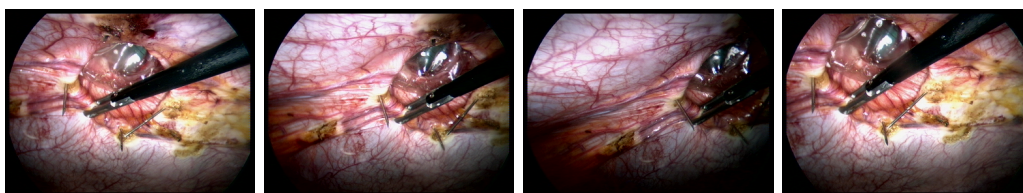


Figura 2.1: Selección de frames de una operación real de reparación de hernia ventral.

Empleando el software Matlab se realizó una primera reconstrucción de la escena mediante técnicas de triangulación Delaunay 2D. Empleando técnicas de geometría proyectiva [9] (generación de Homografías) para texturizar los triángulos de la malla creada, se obtuvo una primera reconstrucción 3D de la cavidad abdominal (fig. 2.2).

Un análisis detallado del mapa 3D llevado a cabo manualmente, verificó la existencia de puntos espurios en el mapa 3D estimado. Los puntos espurios pueden detectarse por su excesivo error de reproyección y por la existencia de puntos homólogos creados erróneamente. Un error de reproyección aceptable sería entorno al pixel pero en este caso aparecen errores de reproyección de hasta 12 píxeles o más.

Para asegurar un modelo fotorealista que represente a la perfección la escena, es preciso analizar cuáles son las causas que incurren en la existencia de puntos espurios en el mapa final obtenido por PTAM.

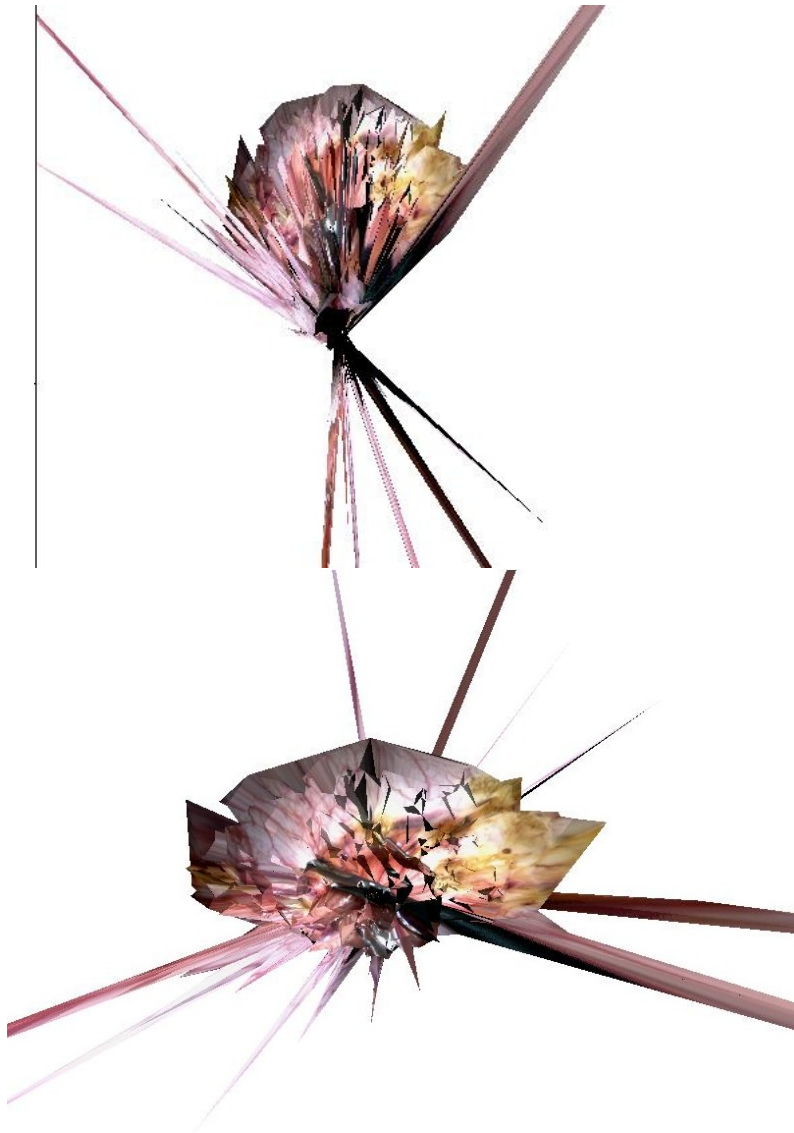


Figura 2.2: Reconstrucción 3D de la cavidad abdominal a partir del mapa estimado por PTAM.

En el siguiente capítulo 3 se presenta un algoritmo que asegura la obtención de un modelo 3D libre de espurios. Una vez se haya conseguido un método que asegure la eliminación total de puntos espurios del mapa estimado, se analizará el algoritmo actual de detección y eliminación de espurios implementado en PTAM para verificar su correcto funcionamiento, modificándolo convenientemente para conseguir un mapa libres de puntos espurios en tiempo real y a frecuencia de vídeo. Estas mejoras se presentan el capítulo 4.

Capítulo 3

Eliminación de espurios a posteriori

El objetivo que se persigue a lo largo de todo el proyecto es obtener un mapa de puntos 3D estimado de la escena (cavidad abdominal) libre de espurios que permita obtener un modelo 3D lo más fidedigno posible.

Para garantizarlo se ha desarrollado un algoritmo que realice un tratamiento a posteriori de los datos obtenidos por el algoritmo PTAM eliminando todos los puntos que perturban la escena y dañan la reconstrucción 3D.

El algoritmo consta de tres etapas: la primera de ellas se centra en la eliminación de puntos espurios a partir de su error de reproyección. La segunda de las etapas eliminará puntos espurios según su ángulo de paralaje. Como último paso en la eliminación de espurios aparece un filtrado de puntos espurios en función de la profundidad relativa de los puntos con respecto a la de sus vecinos (relación entre la coordenada Z del punto y la de sus vecinos).

Los resultados que se obtienen pueden verse en el capítulo 6 y en el anexo A.

3.1. Eliminación según Error de Reproyección

La primera etapa del algoritmo se centra en la detección automatizada de emparejamientos espurios y su correspondiente eliminación en función del error de reproyección estimado.

A partir del mapa 3D estimado y la información que aporta cada KF (puntos homólogos y posición de la cámara) es posible llevar a cabo un análisis a posteriori de la información. El procedimiento a seguir en esta etapa será:

1. Reproyectar el mapa de puntos 3D estimado sobre los diferentes KFs.
2. Determinar el error de reproyección entre el punto y sus correspondencias.
3. Eliminación de aquellos puntos con un error de reproyección por encima del threshold σ .

El modelo de proyección empleado para determinar el error de reproyección de los puntos es el mismo que se emplea en el algoritmo PTAM (ver cap. 2).

Será el módulo de este error de reproyección e_j el que se emplee en el algoritmo de eliminación para aceptar los puntos inliers del mapa (ec. 3.1):

$$|e_j| = \sqrt{e_{jx}^2 + e_{jy}^2} \quad (3.1)$$

El error de reproyección varía según el nivel de la escala en la que fue detectada la característica. Por ese motivo, es necesario ajustar todos los puntos del mapa al mismo nivel. Para ello, el error

de reproyección se divide para un valor de ruido $\sigma = 2^l$ como sucede en el cálculo de la función objetivo de Tukey ec. 2.7. Siendo l igual al nivel de la pirámide (donde $l = 0$ equivale a escala fina y $l = 3$ a la escala basta) en el que fue detectado.

Normalizados los valores al mismo nivel de escala se procede a determinar si el punto debe eliminarse del mapa. En el capítulo siguiente (Eliminación de espurios en tiempo real) se analiza el algoritmo de detección y eliminación implementado en PTAM en tiempo real pero como pequeño adelanto comentar que el algoritmo original implementado se focaliza también en la detección y eliminación de espurios en función del error de reproyección. Por esa razón, el threshold empleado en esta etapa de eliminación es el mismo que en el empleado en la detección y eliminación de espurios en tiempo real por PTAM (ver cap.4). Concretamente se empleará el threshold (valor de corte σ) que se propone como mejora y que se comentará en el capítulo siguiente.

$$|e_j| \leq \sigma \quad (3.2)$$

No resulta coherente emplear un valor más restrictivo de σ , por ejemplo $\sigma = 1$, para esta etapa de eliminación de espurios siendo que el sistema de detección y eliminación en tiempo real implementado en PTAM no elimina puntos con error de reproyección menor a un threshold fijado. Además, resulta innecesario ser tan restrictivos eliminando puntos según error de reproyección para garantizar un mapa final que reconstruya de manera fidedigna la escena 3D.

3.2. Eliminación por ángulo de paralaje

Además de la existencia de espurios debida al error de reproyección, se pudo comprobar como una sección de aquellos puntos considerados espurios durante el análisis inicial, cumplían una serie de condiciones comunes: eran puntos vistos por cámaras muy próximas entre sí (en alguna ocasión consecutivas) y además eran únicamente vistos por un par de KFs de la secuencia. Como eliminar todos los puntos vistos únicamente por dos cámaras o cámaras que estuvieran demasiado próximas originaba que se eliminaran mayor cantidad de puntos inliers que outliers y no todos los espurios eran eliminados, se planteó que el algoritmo eliminara puntos espurios en función de su ángulo de paralaje.

El método para la eliminación de puntos espurios con respecto a su ángulo de paralaje es el siguiente:

1. Búsqueda de los keyframes que tienen correspondencia con el punto.
2. Cálculo de los rayos de proyección del punto respecto a las cámaras con emparejamiento.
3. Cálculo del ángulo de paralaje para los distintos pares de rayos de proyección.
4. Eliminación del punto que posee un ángulo de paralaje menor a un threshold.

El ángulo de paralaje es el ángulo que forma la intersección de los rayos de luz que unen el punto del mapa 3D con el centro óptico de la cámara que observa la característica [10]. Se calculan los vectores directores de estos rayos (\vec{u} y \vec{v}) y se determina el ángulo α (ec. 3.3) que se crea como intersección entre rectas (fig. 3.1(b)).

$$\alpha = \arccos \frac{\vec{u}\vec{v}}{|\vec{u}||\vec{v}|} \quad (3.3)$$

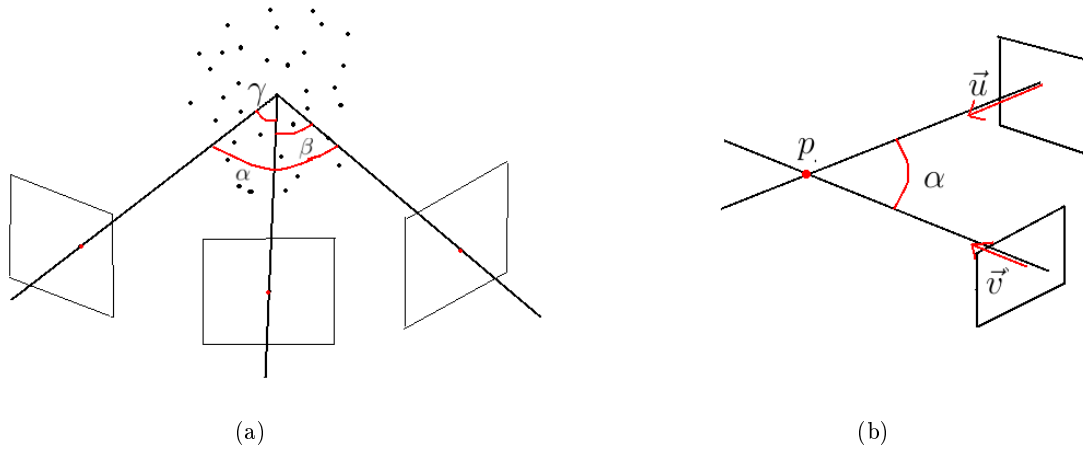


Figura 3.1: (a) Búsqueda de los distintos KFs que ven el punto. (b) Cálculo del ángulo entre los rayos de los KFs.

Se han convertido los ángulos obtenidos de grados a píxeles, para mantener una coherencia entre unidades en el proceso de eliminación de espurios. El ángulo de visión de la cámara endoscópica es (70°) y el tamaño de la imagen de captura de (384×288 px). Al no ser el pixel cuadrado, se hace una pequeña aproximación suponiéndolo de 310×310 px para obtener la relación de conversión de grados a píxeles (ec. 3.4).

$$\frac{\text{grados}}{\text{pixel}} = \frac{\text{ang_endoscopio}}{\text{Tam_imagen}} \quad (3.4)$$

El mayor de todos los ángulos de paralaje (fig. 3.1(a)) calculados para un punto, será el elegido para determinar si el punto cumple o no la condición de ángulo de paralaje ya que será el que tenga mayor peso de entre todos los ángulos calculados. Todo punto cuyo mayor ángulo de paralaje sea menor a 10 píxel (threshold de eliminación) será considerado espurio y eliminado del mapa.

Como ocurría en la etapa de eliminación de espurios con error de reproyección, es preciso ajustar el ángulo de paralaje de todos los puntos al mismo nivel de escala. Es por eso que se multiplica el ángulo de paralaje por su nivel de ruido correspondiente al nivel de escala en el que ha sido visto ($\sigma = 2^l$, con l el nivel de escala).

3.3. Eliminación por Error de Mínima Mediana en profundidad

Se analizaron las reconstrucciones fotorealistas que se obtenían aplicando los algoritmos de eliminación anteriores y se comprobó que sobre el mapa final seguían apareciendo puntos espurios. Estos puntos espurios aparecían de manera aleatoria por toda la reconstrucción por lo que se pensó en hacer un análisis local de todos los puntos del mapa.

La solución que se plantea es analizar de manera individual las características de cada punto con la de sus vecinos. En concreto la característica empleada para la determinación de si un punto es espurio o no la marcará la coordenada Z del mismo (profundidad en la escena).

El método de eliminación que se propone es el siguiente:

1. Comprobar si la característica se encuentra entre los umbrales de profundidad del KF ($dSceneDepthMean$) en el que se observa.

2. Proyección del punto 3D sobre el primer KF en el que se ve el punto.
3. Búsqueda de los puntos del mapa 3D que se encuentran en un radio próximo al punto.
4. Determinación de las coordenadas Z de los puntos seleccionados y comparación con la del punto analizado ($dSceneDepthMedian$).
5. Eliminación del punto si el valor de la coordenada Z no se encuentra entre unos umbrales fijados.

La primera parte de este algoritmo hace una comparación entre la profundidad de la escena ($DepthMean$) y la coordenada Z del punto p_j . Una vez se ha ajustado la posición de la cámara mediante la minimización de la función robusta del error de reproyección durante el Tracking, se calcula el valor medio de la profundidad de la escena (ec. 3.5 y ec. 3.6).

$$DepthMean = \frac{\sum_{k=1}^N z_k}{N} \quad (3.5)$$

$$\sigma^2 = \frac{\sum_{k=1}^N z_k^2 - N * DepthMean^2}{N} \quad (3.6)$$

Donde N es el número de puntos empleados para determinar la profundidad del frame, z_j es el valor de la coordenada z de cada punto k del conjunto.

La condición de eliminación según la profundidad media de la escena es poco restrictiva ya que para el cálculo de $DepthMean$ no se ha empleado todo el mapa estimado sino una pequeña porción de este conjunto (ec. 3.7 y 3.8) por lo que se da cierta permisividad al resultado.

$$p_j(z) \geq DepthMean - 3 * \sigma_{media} \quad (3.7)$$

$$p_j(z) \leq DepthMean + 3 * \sigma_{media} \quad (3.8)$$

Sobre la figura 3.2 se puede ver un ejemplo aplicado sobre una de las secuencias reales con las que se ha trabajado. El mapa de puntos representados es el conjunto de puntos que se observan desde uno de los KFs. Aquellos puntos que se encuentren fuera de los umbrales establecidos a partir de la $dSceneDepthMean$ (línea verde) y la desviación estándar σ_{media} (líneas rojas) de la cámara serán eliminados (puntos azules).

En la segunda etapa se realiza una eliminación local de los puntos 3D. Se buscará para cada punto, en qué KF ha sido visto el punto por primera vez y se proyectará el mapa estimado sobre él. Iterativamente, se buscarán aquellos puntos que se encuentren dentro del radio de búsqueda R que determinará si el punto es outlier.

El mínimo número de puntos para llevar a cabo el análisis es de 10 puntos inliers. De manera progresiva, se irá incrementado el valor de R hasta que se hayan detectado el mínimo de puntos inliers. Si superado el valor máximo de R (50 píxeles) no se detectan el suficiente número de puntos, el punto p_j será considerado espurio y eliminado del mapa.

Si por el contrario para alguno de los radios de búsqueda se alcanza el mínimo de puntos, se procederá a calcular el error de mínima mediana. Se emplea la mediana porque es robusta siempre

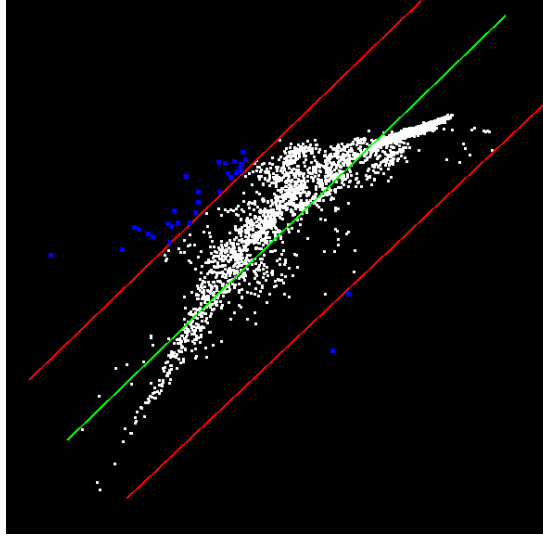


Figura 3.2: Eliminación de puntos outliers en función de la profundidad media del KF

y cuando el porcentaje de espurios sea inferior al 50 %. Algo que no ocurre con la media donde un valor espurio perturba los resultados de este estimador estadístico.

Para cada punto p_z de los puntos detectados durante la búsqueda radial, se determinará el error de mediana de las diferencias absolutas de las coordenadas Z del conjunto de puntos P (ec. 3.9).

$$error_{prof} = med(|p_z - z_j| : z_j \in P) \quad (3.9)$$

ErrorMinMedian será aquél que haga mínimo el error de mediana de las diferencias entre las coordenadas de los puntos (ec. 3.10).

$$ErrorMinMedian = min(error_{prof}) \quad (3.10)$$

$$\sigma_{median} = 1.4826 * (1 + \frac{5}{N * 2 - 2}) * \sqrt{ErrorMinMedian^2} \quad (3.11)$$

La desviación estándar robusta se calculará a partir a partir de la ecuación planteada por Rousseeuw [11] en la que el cálculo de la desviación estándar estimado está relacionado con la mediana de los errores absolutos de profundidad (ec. 3.11), siendo N igual a 10.

Como condición adicional es necesario que el punto todavía siga siendo considerado inlier para que se tenga en cuenta a la hora de la búsqueda de los vecinos más próximos. Si el valor del error de mediana en profundidad Z para el punto analizado ($error_{prof}(j)$) no se encuentra dentro del umbral de aceptación, el punto se considerará outlier y será eliminado del mapa final (ec. 3.12).

$$\frac{|error_{prof}(j)|}{\sigma_{median}} > 2.5 \quad (3.12)$$

Este método propuesto permite la eliminación de parte de los puntos correspondientes al instrumental médico-quirúrgico empleado en la operación. Estos puntos verdaderamente no están mal estimados pero a la hora de llevar a cabo la reconstrucción 3D puede perturbar la escena y generar una interpretación errónea de la misma.

Capítulo 4

Eliminación de espurios en tiempo real

En el capítulo anterior se ha presentado un algoritmo que asegura la obtención de mapas finales libres de espurios para su posterior procesado en la obtención de modelos 3D fotorealistas.

El siguiente objetivo es conseguir que sea el propio algoritmo PTAM el que lleve a cabo la detección y eliminación de espurios durante la ejecución de la secuencia, en tiempo real y a frecuencia de vídeo.

Cuanta menor sea la necesidad de empleo del algoritmo de eliminación a posteriori, se demostrará que las modificaciones propuestas en el algoritmo original mejoran la precisión del mapa, eliminando mayor fracción de espurios durante la ejecución de la secuencia.

PTAM cuenta con su propio algoritmo de detección y eliminación de espurios, que focaliza su eliminación en el error de reproyección de los puntos 3D. En la primera parte de este capítulo se analizará el algoritmo original implementando para la eliminación de espurios y se propondrán una serie de mejoras que posibiliten la reducción de la tasa de espurios.

Como complemento a estas mejoras sobre el algoritmo actual de detección de espurios, se presenta una nueva condición de detección y eliminación según la observabilidad de los puntos en los frames de la secuencia. Esta mejora pretende cubrir las necesidades que aparecen tras comprobar que existe la posibilidad de que no todos los puntos espurios se eliminen únicamente con el sistema actual de eliminación.

4.1. Sintonización del algoritmos de eliminación de puntos con Error de Reproyección excesivo

La detección y eliminación de espurios se lleva de manera conjunta por los procesos de mapping y tracking. La tarea de mapping se encarga de la eliminación mientras que el tracking se centra en la detección y determinación de qué puntos deben ser considerados espurios.

Para cada punto estimado del mapa se computa el número de veces que ese punto ha sido visto como espurio ($n_{SeenOutlier}$) y cuántas ha sido visto como bueno ($n_{SeenInlier}$) en los frames de la secuencia. En cada iteración de la tarea del mapping se comprueban estos valores para proceder a la eliminación del punto del mapa. La condición que rige si un punto p_j debe mantenerse en el mapa 3D o por el contrario debe eliminarse del conjunto, viene determinada por la ecuación 4.1.

$$\begin{aligned} \text{if } n_{SeenOutlier} \geq 20 \ \&\& \ \frac{n_{SeenInlier}}{n_{SeenInlier} + n_{SeenOutlier}} \leq 50 \% \\ &\Downarrow \\ p_j &= Outlier \end{aligned} \tag{4.1}$$

Al menos el **50 %** de las veces que p_j ha sido visto y detectado en los frames de la secuencia, debe computarse como inlier para que el punto continúe en el mapa. Como condición adicional, a

cada punto se le otorga una vida previa antes de proceder a su eliminación del mapa. Debe ser visto al menos 20 veces como outlier antes de proceder a su eliminación, de tal forma que se permite a un punto recién creado en el mapa ajustar su posición en la escena tras un ajuste local y global (Bundle Adjustment) sin proceder a su instantánea eliminación.

Como mejora en la detección y eliminación de espurios, se propone modificar esta condición de aceptación de puntos inliers en el mapa (**sintonización de la puntuación acumulada de observaciones espurias**). Haciendo más restrictiva la condición que se refiere al número de veces que debe ser un punto visto y detectado como inlier. El objetivo que se busca es conseguir eliminar mayor fracción de espurios durante la ejecución y así conseguir disminuir la tasa de espurios en el mapa estimado.

En este caso, aumentar en exceso esta condición reduce la densidad del mapa sin la certeza de asegurar que los puntos eliminados hubieran sido ajustados en un posible ajuste de haces (Bundle Adjustment Local y Global) posterior. Por esta razón es necesario encontrar un equilibrio entre lo estricto que se quiere ser eliminando y la densidad que se desea en el mapa final. Se ha modificado la fracción de forma que ahora será necesario que un punto haya sido contabilizado como inlier por lo menos el 80 % de las veces que ha sido visto (ec. 4.2) en los frames.

$$\begin{aligned} \text{if } nSeenOutlier \geq 20 \ \&\& \ \frac{nSeenInlier}{nSeenInlier+nSeenOutlier} \leq 80 \% \\ \Downarrow \\ p_j = Outlier \end{aligned} \quad (4.2)$$

El Tracking por su parte, se encarga de ir estimando la posición local de la cámara para cada frame y a su vez determinar si los puntos observados son espurios o no. Es un proceso iterativo en el que por medio de técnicas de optimización no lineal se estima la posición del frame en función del mapa que observa (ec. 2.5).

El peso que cada punto tiene en el ajuste del frame viene determinado por el estimador de mediana robusto (σ_T) de la distribución de la desviación estándar de los residuos de los puntos que intervienen en el ajuste del frame. Aquellos puntos cuyo residuo supera el valor del estimador robusto se les atribuye un peso nulo en el ajuste de la posición del frame. Así mismo, aquellos puntos cuyo peso en el ajuste sea nulo serán considerados espurios y se añadirán a la cuenta de espurios.

La determinación de si un punto es espurio o no, se realiza en la última de las diez iteraciones del ajuste de la posición de la cámara. Como los valores que se obtienen del estimador robusto de mediana son muy elevados, en las últimas iteraciones del cálculo de la posición de la cámara, este valor se fija a 4 píxeles. De manera que aquellos puntos cuyo error de reproyección sea mayor a 4 píxeles tendrán un peso nulo en la estimación de la posición de la cámara y por tanto, considerados espurios.

Obligados a ser lo más precisos posibles para su posterior tratamiento, errores de reproyección de esa magnitud son inaceptables. Por esta razón como segunda mejora, se propone separar el threshold de detección de espurios (**sintonización del umbral de marcado de observaciones espurias**) del threshold empleado en la estimación de la posición de la cámara, manteniendo los umbrales de aceptación utilizados para tal efecto ya que la estimación de la trayectoria de la cámara no se ve perturbada por la aparición de puntos espurios.

Tras varios experimentos realizados con secuencias reales, se ha comprobado que reducir el valor de σ en exceso implica que el sistema se pierda y no sea posible obtener un mapa estimado de la escena. Estos resultados plantean la necesidad de encontrar un equilibrio entre lo estricto que se quiere ser a la hora de detectar puntos espurios y la necesidad de que el sistema no se pierda. De tal forma que el valor mínimo aceptado hasta el que se puede reducir el valor de *sigma* sin que el sistema se pierda en este tipo de secuencias es de 2 píxeles. Aquellos puntos cuyo error de

reproyección sea menor a 2 píxeles serán considerados inliers. Este valor de σ será el mismo que se emplee en el algoritmo de detección y eliminación de espurios a posteriori en la etapa de eliminación de espurios explicado en el capítulo anterior (3).

Estas pequeñas modificaciones además de mejorar los resultados en el mapa estimado, permiten mejorar la estimación de la trayectoria ya que se reduce el número de espurios utilizados para estimar la posición de la cámara durante el Tracking permitiendo ser lo más precisos posibles en el cálculo de la trayectoria.

4.2. Eliminación de puntos no reobservables en frames

Durante una ejecución in-vivo sobre una secuencia de cirugía de hernia ventral se puede ir observando sobre cada frame el mapa estimado. Con el algoritmo trabajando a frecuencia de vídeo (30Hz o 30 imágenes/s) se observa cómo algunos puntos no encuentran correspondencia entre frames consecutivos.

Si el punto está bien estimado cabría esperar que hasta que el endoscopio no terminara de barrer la zona en la que ese punto es visto, el punto fuera detectado en la imagen. En la figura 4.1 se puede ver un ejemplo de que este hecho no ocurre así.

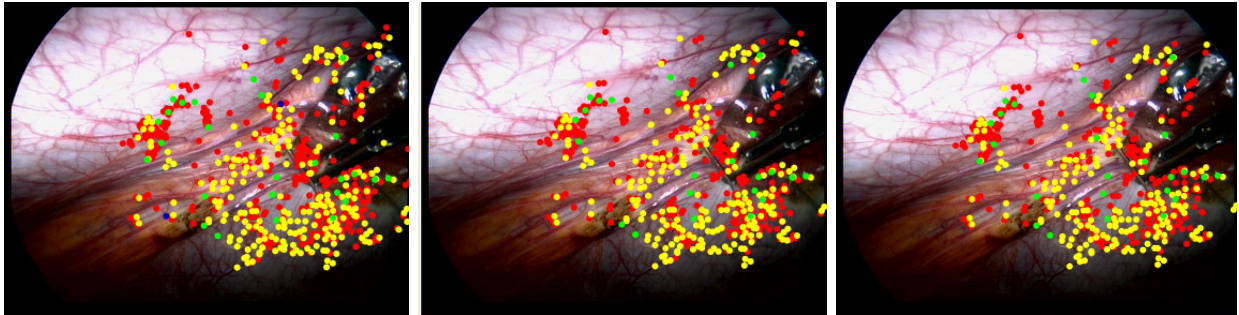


Figura 4.1: Frames consecutivos en los que se puede apreciar cómo no todo el mapa estimado es proyectado sobre ellos.

Que ciertos puntos del mapa no encuentren correspondencia cuando son proyectados en ciertos frames, plantea la posibilidad de que esos puntos hayan sido mal estimados y por esa razón no obtengan correspondencia alguna.

Por este motivo se ha creado una nueva condición que elimine aquellos puntos que siendo potencialmente visibles en el frame, no han sido detectados (**condición de reobservabilidad de puntos en frame**). De esta forma lo que se quiere evitar es la aparición de emparejamientos espurios que únicamente se vean en 2 cámaras consecutivas ya que para este tipo de emparejamientos, el error de reproyección que generan es 0 y por tanto, por medio del anterior algoritmo de eliminación, no serían detectados y eliminados del mapa.

El punto de inclusión de esta nueva condición de detección, será en la sección de código que se encarga de determinar si un punto potencialmente visible en el frame ha sido detectado o no (proceso de tracking). Se computarán las veces que ha sido detectado y visto a lo largo de la secuencia de tal forma que puntos que debiendo estar en la imagen (visibles) y que no han sido detectados, se considerarán espurios.

Al igual que ocurría con el algoritmo de eliminación según el error de reproyección en la que el mapping se encarga de la eliminación de los puntos que no cumplan la condición, en esta ocasión el proceso de mapping se encargará también de eliminar los puntos que no cumplan la condición de reobservabilidad en el frame (visible y detectado).

La fracción de aceptación del punto como inlier será de que al menos un 30 % de las veces que ha sido visible, haya sido detectado (ec. 4.3).

$$\frac{Detected}{Detected + NotDetected} \geq 30 \% \quad (4.3)$$

El porcentaje que se propone para esta condición es sustancialmente menor que en la condición de eliminación por error de reproyección ya que durante el barrido que se hace sobre la cavidad abdominal, los puntos internos del defecto pueden ser potencialmente visibles sobre la imagen pero no van a ser vistos (debido a oclusión) por lo que exigir una restricción mayor reduciría el mapa sin la certeza de saber si todos los puntos eliminados son realmente espurios.

Capítulo 5

Reconstrucción fotorealista de la escena

El output generado por PTAM es una nube semi-densa de puntos que representa la forma de la cavidad abdominal. Visualizar únicamente una nube puntos puede resultar útil desde el punto de vista de la visión y la robótica, pero puede resultar insuficiente desde un punto de vista quirúrgico donde resulta interesante generar un mapa denso de la escena que facilite la interpretación del modelo y aporte información entendible para el cirujano.

Aplicando sobre esta nube de punto un algoritmo de triangulación que permita texturizar la superficie generada durante el mallado, se pretende conseguir un modelo fotorealista de la cavidad que aporte esa información que un mapa puntos 3D y una secuencia de imágenes de laparoscopia por separado no aportan al cirujano.

Para explicar el método propuesto, se van a emplear de manera conjunta las secuencias (b) y (f) de la figura 6.1. En la fig.5.1 se muestra un adelanto de los resultados que se van a obtener al aplicar el algoritmo. Se puede ver cómo los puntos correspondientes al defecto están correctamente triangulados y unidos a lo que sería la cavidad abdominal por la zona del defecto herniario.

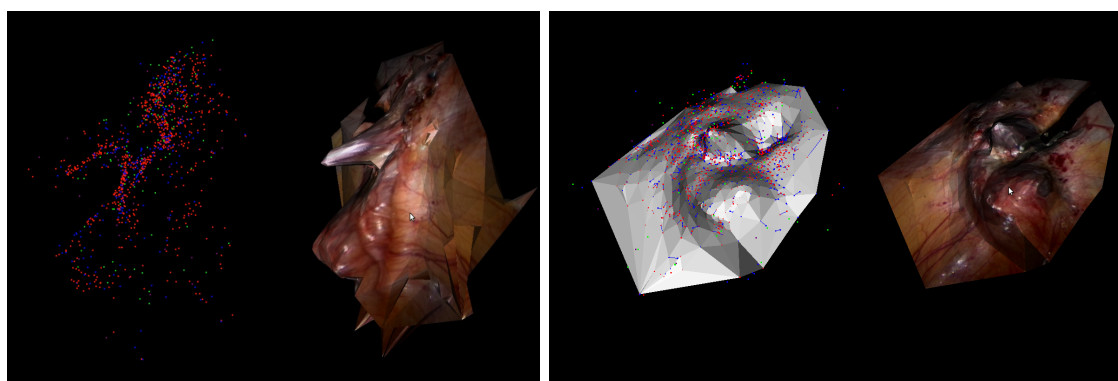


Figura 5.1: Reconstrucción 3D de la cavidad abdominal a partir de una secuencia real de cirugía

Se pensó en utilizar la Point Cloud Library (PCL) como punto de partida para la generación de las conexiones de la malla de la reconstrucción ya que cuenta con una gran variedad de funciones para la manipulación y trabajo con nubes de puntos. Greedy Projection Triangulation [5] y Poisson [6] son sus principales métodos de triangulación. El método más adecuado para generación de mallas 3D a partir de nubes de puntos es Greedy Projection Triangulation.

Greedy Projection Triangulation (GPT) es un algoritmo de triangulación de puntos 3D basado en la proyección local del mapa en la cual la triangulación de los puntos se va realizando de forma incremental (fig. 5.2): i) para cada punto p del mapa se buscan los k -vecinos más próximos que se encuentran en el interior de una esfera con radio r (valor que puede ser adaptado a la densidad del mapa). ii) Después son proyectados sobre un plano que es aproximadamente tangencial a la superficie que conforman los k -vecinos de p .

iii) Finalmente se aceptarán los nuevos triángulos que den validez al criterio de aceptación por ángulo (El ángulo de inclinación del triángulo se encuentra entre unos valores máximos y mínimos pre-fijados).

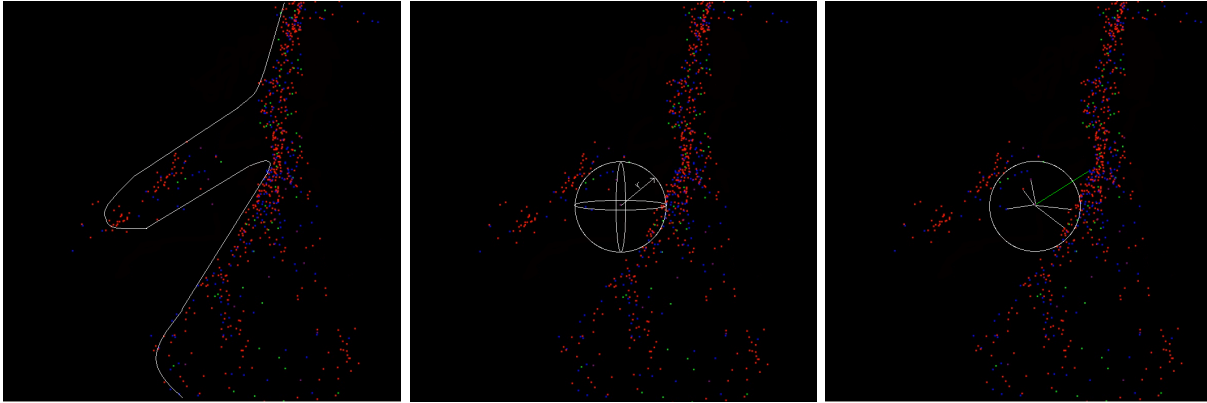


Figura 5.2: Método de búsqueda empleado para la triangulación Greedy. (a) silueta del contorno que debería dar forma a la cavidad abdominal. (b) Búsqueda de los k -vecinos que se encuentran dentro del radio de búsqueda. (c) Selección de los puntos que cumplen la condición de aceptación y que serán triangulados. En verde, la línea que debería seguir el mallado correcto.

Radios de búsqueda r de los k -vecinos más próximos pequeños, generan una triangulación con una alta densidad de zonas vacías. Por el contrario, un aumento del valor de r genera triángulos entre zonas de la cavidad abdominal que no deberían surgir.

Este proceso resultaría interesante si la nube de puntos fuera más densa. Permitiría que la distancia entre un punto p y sus k -vecinos más próximos fuera menor y por tanto permitiría reducir r en la esfera de búsqueda en el proceso de triangulación. El principal problema que surge a la hora triangular aparece en la zona interna del defecto herniario donde puntos que pertenecen a la oquedad resultan triangulados a la pared abdominal en vez de al defecto herniario que es lo esperado.

Descartada la utilización de PCL para obtener la reconstrucción de la cavidad se ha optado por desarrollar un algoritmo basado en técnicas tradicionales de triangulación (Delaunay 2D) que de manera secuencial permita obtener el modelo fotorealista con un mallado entre puntos correcto.

5.1. Algoritmo de triangulación implementado

Para su implementación se emplea la librería VTK [12, 13] otro de los referentes en el tratamiento de nubes de puntos. Cuenta con su propia función de triangulación Delaunay que permite realizar triangulaciones sobre un plano determinado. En la figura 5.3(a) se puede ver la reconstrucción obtenida empleando el plano XY como base para la triangulación.

La fig. 5.3(b) representa la forma de la cavidad real esperada. Como se puede observar en la fig 5.3(a) , una triangulación predeterminada a un único plano genera un mallado erróneo, uniendo puntos del mapa no contiguos. Se ha empleado el plano XY que el algoritmo emplea por defecto pero modificar este plano a uno predicho por el usuario no presenta mejora alguna. La dificultad del problema radica en encontrar un método que discierna qué puntos pertenecen al interior del defecto de los puntos de la pared para que el método de triangulación genere una malla realista a lo observado en la secuencia.

El método que se ha seguido para la implementación del algoritmo 1 es una adaptación de la de Civera [7] y la de Óscar Gracia [8]. En el algoritmo propuesto por Civera, la cámara únicamente rotaba y la triangulación de los puntos se llevaba a cabo sobre el plano X-Y del sistema mundo.

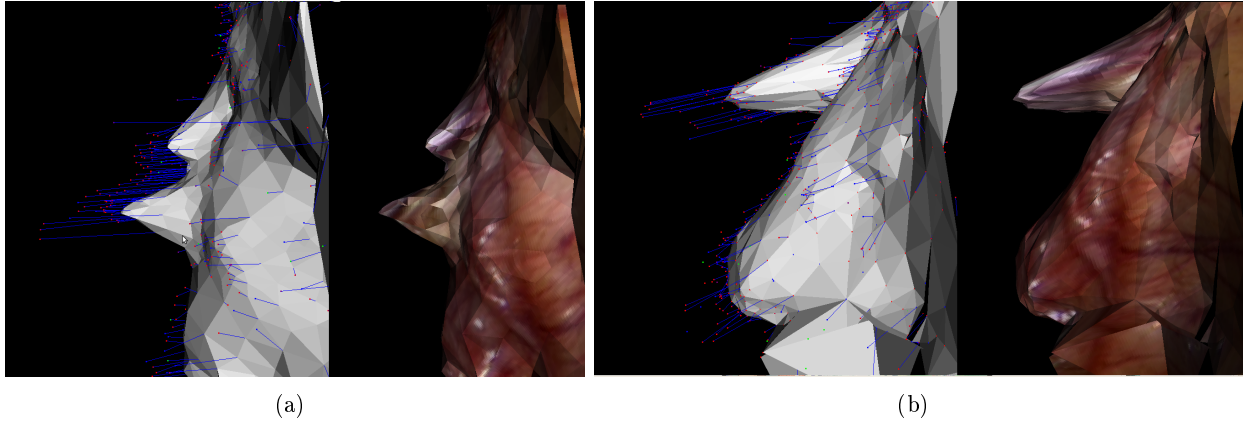


Figura 5.3: Reconstrucción fotorealista obtenida para la secuencia (f) de la fig. 6.1. La imagen de la izq. muestra el resultado tras aplicar una triangulación DELAUNAY sobre el plano XY. La de la derecha es el resultado final obtenido tras la aplicación del algoritmo desarrollado de triangulación.

Óscar G. propone un algoritmo que permite una reconstrucción densa de la cavidad abdominal en el caso de que la cámara se traslade sobre el mismo plano X-Y del sistema mundo. En este proyecto, lo que se propone es reconstruir la cavidad abdominal triangulando el mapa sobre los planos X-Y de cada KF.

Algorithm 1 Triangulación y Texturización de la Escena Fotorealista

```

1: Input: Mapa puntos 3D  $Map3D$ ; Secuencia de Imágenes  $KF$ ;
2: Lista de triángulos  $T \leftarrow$  Lista vacía
3: Lista de puntos triangulados  $TL \leftarrow$  Lista vacía
4: Hull2D de puntos ya triangulados  $Hull2D \leftarrow$  Lista vacía
   {Iniciación de la Triangulación}
5:  $TL \leftarrow$  Matches primer  $KF$ .
6:  $T \leftarrow$  Triangulación Delaunay de  $TL$ .
   {Inclusión de nuevos puntos en la triangulación}
7: for all  $k \in KF$  do
8:    $Hull2D \leftarrow$  Hull de las proyecciones de los puntos ya triangulados sobre el frame  $k$ .
9:   for all  $p \in Map3D$  do
10:    if  $p$  aún no incluido en  $TL$  then
11:      if  $p$  dentro del hull de los puntos ya triangulados then
12:        Proyectar  $TL$  sobre  $KF$  correspondientes.
13:        Búsqueda del triángulo sobre el que se incluye  $p$  de todo el conjunto  $T$ .
14:        Eliminar de  $T$  triángulos reparticionado y añadir nuevos triángulos.
15:        Añadir a  $TL$  el punto  $p$ .
16:      end if
17:    end if
18:  end for
   {Inclusión de puntos exteriores al Hull2D}
19:  Proyectar todos los puntos que son vistos en el  $KF$ .
20:  Incluir triángulos en los que alguno de sus vértices sea un punto aún no triangulado.
21:  Añadir a  $TL$  los puntos triangulados.
22: end for
23:  $TL \leftarrow$  Suavizado de la malla originada en la triangulación  $TL$ .
   {Selección de textura}
24: for all  $t \in T$  do
25:   Obtener textura del  $KF$  original que ve los 3 vértices del triángulo  $t$ .
26: end for

```

El proceso se divide en 3 etapas. La primera inicializa el mallado triangular mediante una triangulación Delaunay de los puntos vistos sobre el primer KF (fig. 5.4). Todos los puntos del mapa no son emparejados, por lo que existen puntos que han quedado excluidos fuera de esta inicialización.

Éstos puntos excluidos de la triangulación inicial pueden englobarse en dos categorías diferentes: puntos que se encuentran dentro de la triangulación cuando ésta es re proyectada sobre el KF y puntos que quedan fuera. Será en la segunda etapa del algoritmo 1 (Inclusión de nuevos puntos en la triangulación) donde se procederá a su incursión.

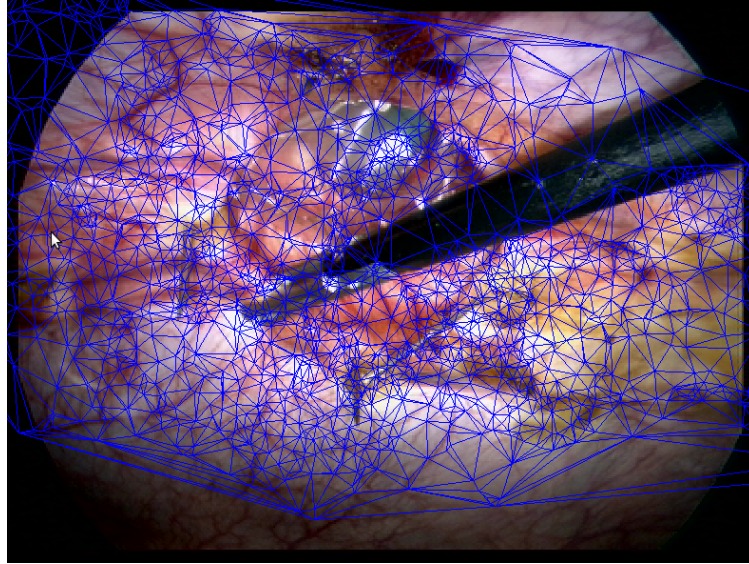


Figura 5.4: Triangulación Delaunay del mapa estimado sobre el primer KF de la secuencia 6.1(b).

La manera de determinar si un punto se encuentra o no dentro de alguno de los triángulos ya creados, será mediante la creación del Convex Hull2D [19] de los puntos incluidos en TL hasta el momento. Si el punto se encuentra dentro del polígono que encierra a todo el mapa ya triangulado, este será buscado para proceder a la retriangulación. Si no es así, se almacenará para su posterior tratamiento como punto externo a la triangulación. El algoritmo implementado para la generación del Hull de la malla es una adaptación del propuesto en [20, 21].

La manera de proceder con aquellos puntos internos al Hull será: proyectar la triangulación sobre el KF que el punto ha sido visto. Iterativamente, buscar sobre qué triángulo el punto se encuentra incluido. Ese triángulo será eliminado y será reparticionado en 3 triángulos nuevos que serán incluidos en la lista T . Únicamente si los 3 vértices del triángulo son vistos en el KF se incluirán los nuevos triángulos y el punto al conjunto T . Si esto no ocurre, se seguirá buscando otro posible triángulo al que pertenezca en otro KF. En la figura 5.5 puede apreciarse el método seguido para la obtención de nuevos triángulos por repartición de los ya existentes en T .

Aquellos puntos que no han sido incluidos dentro del Hull de la malla triangular serán incluidos en TL por medio de una nueva triangulación. La idea surge a partir de la propuesta que hace L. Arteaga en [25] en la que explica un algoritmo de triangulación incremental a partir de una triangulación con preordenación. Se proyectarán todos los puntos del mapa ya triangulado y vistos en el frame y se realizará una triangulación Delaunay2D de los puntos. A continuación se añadirán el resto de puntos vistos en el KF y aún no triángulos fuera del mallado. De esta forma se generará una nueva triangulación restringida por los triángulos formados inicialmente. Aquellos triángulos que generen los puntos aún no incluidos en T serán los incluidos en TL .

El inconveniente que surge es la posible superposición de triángulos a la hora de adicionar

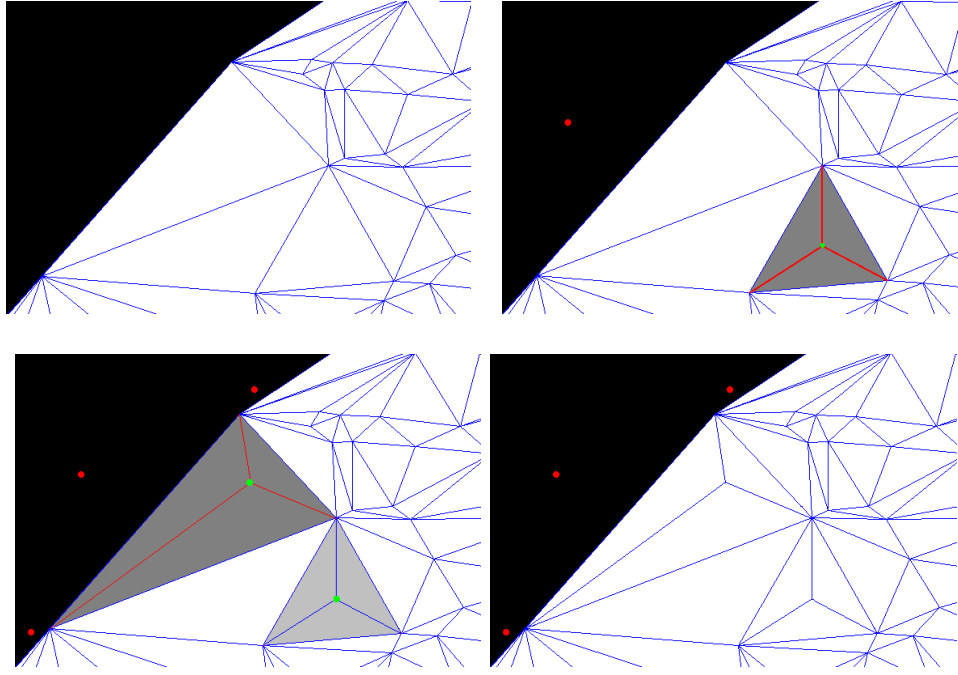


Figura 5.5: Método de triangulación de puntos interiores a la Triangulación Delaunay inicial. Si un punto del mapa se encuentra en el interior de un triángulo ya creado, este se reparticiona eliminando el triángulo original y generando 3 nuevos triángulos al conjunto.

triángulos externos ya que se está suponiendo que todos los puntos que son vistos por el KF, ya han sido triangulados pero la realidad indica que no todos los puntos son vistos por lo que pueden existir interconexiones entre puntos nuevos.

Antes de la selección de la región de la imagen a añadir sobre cada triángulo se aplica un suavizado sobre la malla creada.

De esta manera se pretende reducir las discontinuidades que pudieran surgir entre puntos contiguos provocados por las herramientas empleadas durante la operación (pinzas y agujas). Como alternativas para el suavizado se presentan la utilización de splines [15] y el suavizado Laplaciano explicado en [16, 17]. Se empleará el suavizado Laplaciano propuesto por Nealen en [18] para la obtención de modelos 3D el escogido, ya que VTK cuenta con su propia función del método.

Los resultados de la aplicación del suavizado pueden verse en la fig. 6.9 donde las líneas que unen puntos (el color es según la escala del punto) corresponde al desplazamiento sufrido por los puntos en relación al suavizado llevado a cabo.

La última de las etapas corresponde al proceso de texturización de la malla extrayendo de cada KF la imagen de la región del espacio que encierra el triángulo para después pegarla sobre la superficie del mismo.

Todas las reconstrucciones obtenidas para las distintas secuencias sobre las que ha sido testeado el algoritmo se pueden ver en el capítulo 6 y en el Anexo A.

Capítulo 6

Validación y Resultados

Las secuencias empleadas para la consecución de los experimentos constan de unos cuantos cientos de frames y han sido extraídas de cirugías de eventroplastia laparoscópica in-vivo en humanos grabadas en el Hospital Clínico Universitario Lozano Blesa de Zaragoza. En la figura 6.1 se puede ver un frame de cada una de las cavidades abdominales empleadas para la validación de los resultados.

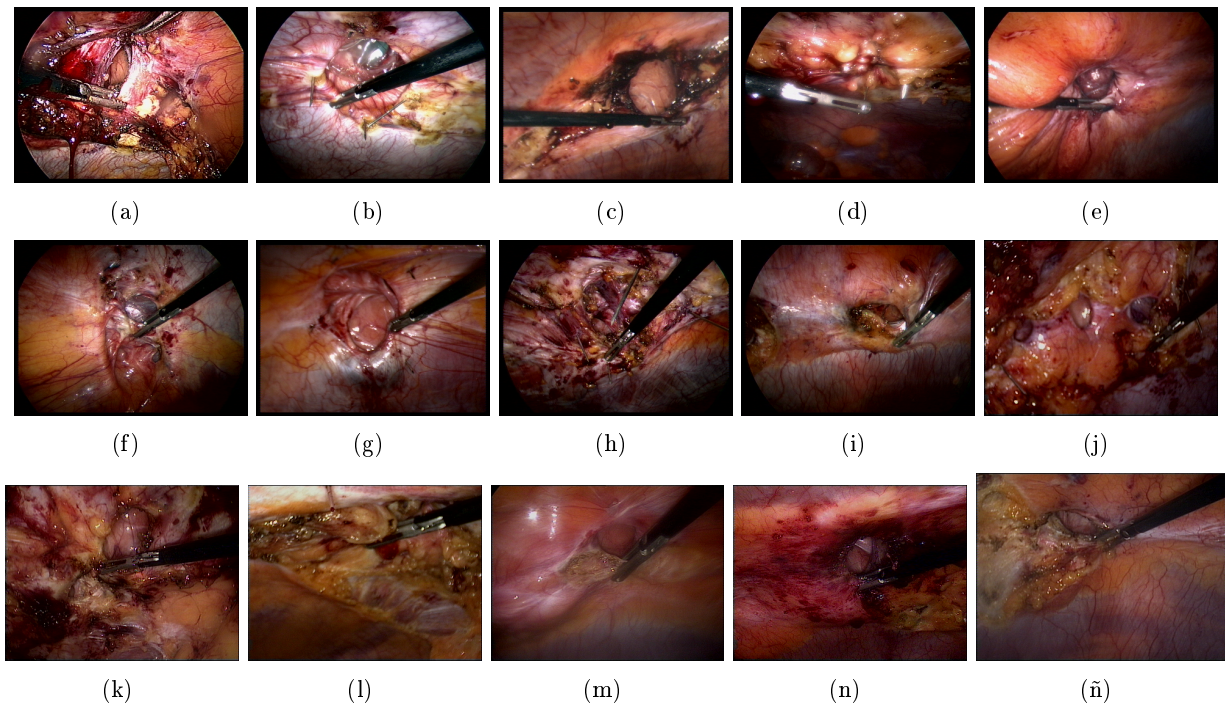


Figura 6.1: Secuencias reales de operaciones con endoscopio proporcionadas por el Hospital clínico Lozano Blesa para verificar los resultados.

Este tipo de secuencias analizadas presentan un común denominador entre ellas: la estructura de la escena es prácticamente rígida en su totalidad, no existe movimiento debido a la respiración del paciente o cualquier movimiento de los órganos. Además, no contienen grandes oclusiones que limiten la visión durante su ejecución que puedan llevar a pérdida por parte del algoritmo PTAM.

Ante la diversidad de texturas de las hernias ventrales utilizadas para la validación, se puede razonar que si los experimentos propuestos son válidos para estas secuencias, el método empleado de detección y eliminación de espurios y el algoritmo de reconstrucción 3D podrán emplearse de manera generalizada en otras secuencias de endoscopio.

Todas las secuencias han sido probadas con las diferentes propuestas en la detección y eliminación de espurios y la generación de modelos 3D fotorealistas. En este capítulo se presentan los

resultados obtenidos para una de ellas en concreto y un comentario general del resto que verifique el correcto funcionamiento. El resto de resultados y modelos 3D quedan recogidos en el anexo A.

Los resultados se centrarán en verificar el correcto funcionamiento de los distintos algoritmos implementados: algoritmo de eliminación a posteriori que garantice un mapa final libre de espurios, algoritmo de eliminación en tiempo real que reduzca la tasa de espurios durante la ejecución de la secuencia a frecuencia de vídeo y algoritmo de triangulación en la generación de modelos 3D. Además se comprobarán los tiempos de cómputo de manera que se pueda garantizar que el sistema funciona en todo momento en tiempo real.

6.1. Eliminación de espurios a posteriori

En primer lugar se va a comprobar la efectividad del algoritmo de detección y eliminación de espurios a posteriori. El objetivo es asegurar que el algoritmo implementando permite obtener un mapa final libre de espurios que puede ser procesado para obtener un modelo 3D de la escena.

Cada etapa se aplica de manera progresiva sobre el mapa estimado, de manera que los resultados que se obtengan de las etapas anteriores tengan validez en las siguientes. Si un punto ha sido considerado espurio en la etapa anterior, no volverá a ser tenido en cuenta en el resto de etapas. El orden de aplicación de las etapas de detección y eliminación será:

1. Eliminación según error de reproyección de los puntos 3D y sus correspondencias.
2. Eliminación según ángulo de paralaje de los puntos estimados del mapa.
3. Eliminación según error de mínima mediana en profundidad.

La primera de las etapas corresponde a la eliminación según el error de reproyección de los puntos. El algoritmo será válido si del mapa final se eliminan todos aquellos puntos que tengan un error de reproyección en alguno de sus emparejamientos, mayor a 2 píxeles. En la figura 6.2 se muestran los errores de reproyección acumulados de los puntos del mapa estimado tras finalizar la ejecución del PTAM y tras su paso por la etapa de eliminación según el error de reproyección.

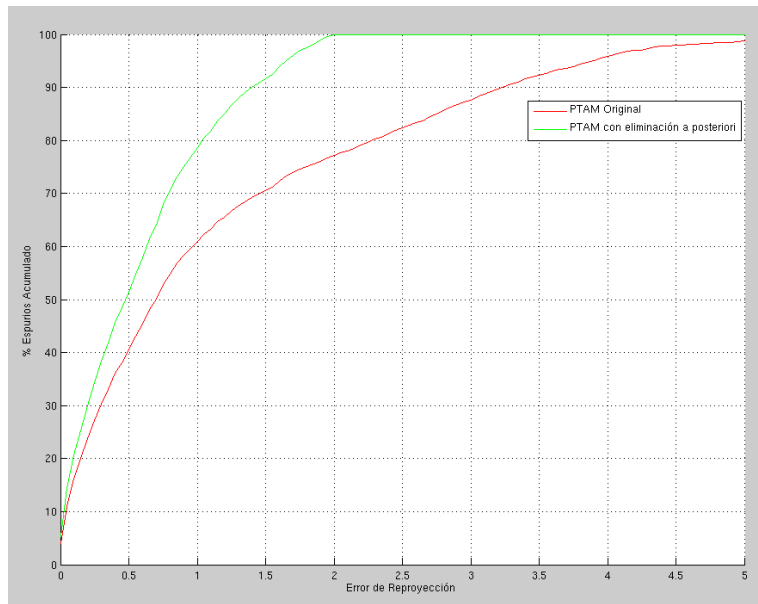


Figura 6.2: Porcentaje acumulado del Error de Reproyección para la secuencia de (b) de la fig. 6.1 según el criterio de eliminación de error de reproyección a posteriori.

Como se puede observar, los errores de reproyección tras la aplicación del algoritmo a posteriori, aseguran un mapa final de puntos con errores de reproyección menores a los 2 píxeles.

Para la segunda de las etapas (ángulo de paralaje) se comprobó que todos aquellos puntos que pasaban la etapa de eliminación cumplían la condición de aceptación, su ángulo de paralaje era mayor a 10 píxeles que era el threshold de corte fijado (sección 3.2).

Se comprobó manual y visualmente qué características tenían los puntos eliminados tras aplicar esta etapa de eliminación. En la figura 6.3 se puede ver el mapa final obtenido tras la ejecución del algoritmo (puntos blancos) junto con los puntos eliminados por la condición de ángulo de paralaje (puntos rojos).

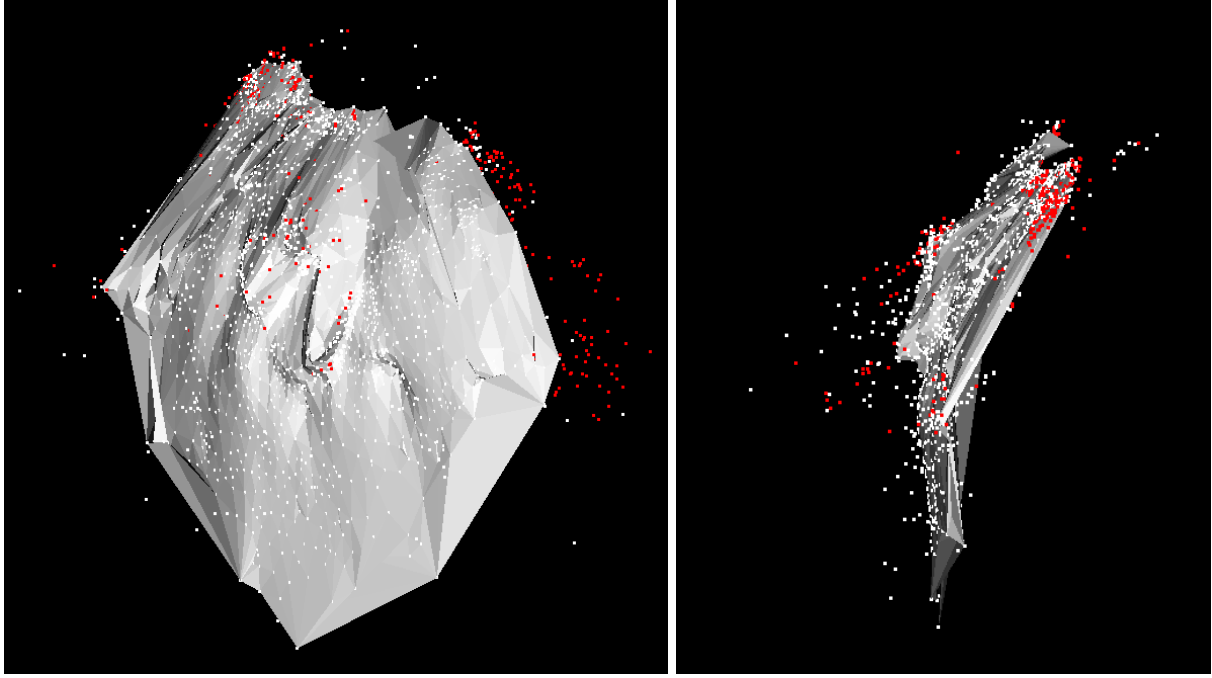


Figura 6.3: Mapa final tras aplicar la etapa de eliminación según ángulo de paralaje. En rojo, los puntos eliminados en la etapa de ángulo de paralaje.

Visualmente se puede apreciar cómo los puntos eliminados por la etapa de eliminación según ángulo de paralaje se concentran en zonas de la pared abdominal que a priori no dan la impresión de estar erróneamente estimadas. El problema que surge es que son puntos vistos por cámaras muy próximas y que generan ángulos de paralaje pequeños y, por esa razón, son eliminados del mapa. Además, se pudo ver que no todos los puntos espurios eran eliminados por esta condición. Se vio la necesidad de implementar la última etapa de las etapas de eliminación de espurios, error de mínima mediana de profundidad (sec. 3.3) para verificar que es posible obtener un mapa final libre de espurios.

Esta tercera etapa tiene como objetivo eliminar los puntos remanentes aún no eliminados por las etapas anteriores, de manera que se consiga un mapa libre de espurios. En la figura 6.4 se pueden ver los puntos eliminados por las etapas de ángulo de paralaje y error de profundidad así como el mapa final generado para el modelo 3D.

Visto que esta última etapa cubre las carencias que presenta la etapa de eliminación por ángulo de paralaje, se plantea la posibilidad de que el algoritmo de eliminación tenga únicamente 2 etapas: eliminación por error de reproyección y por error de mínima mediana en profundidad. De manera que la etapa de eliminación según ángulo de paralaje quede excluida al comprobarse que elimina mayor tasa de puntos inliers que outliers.

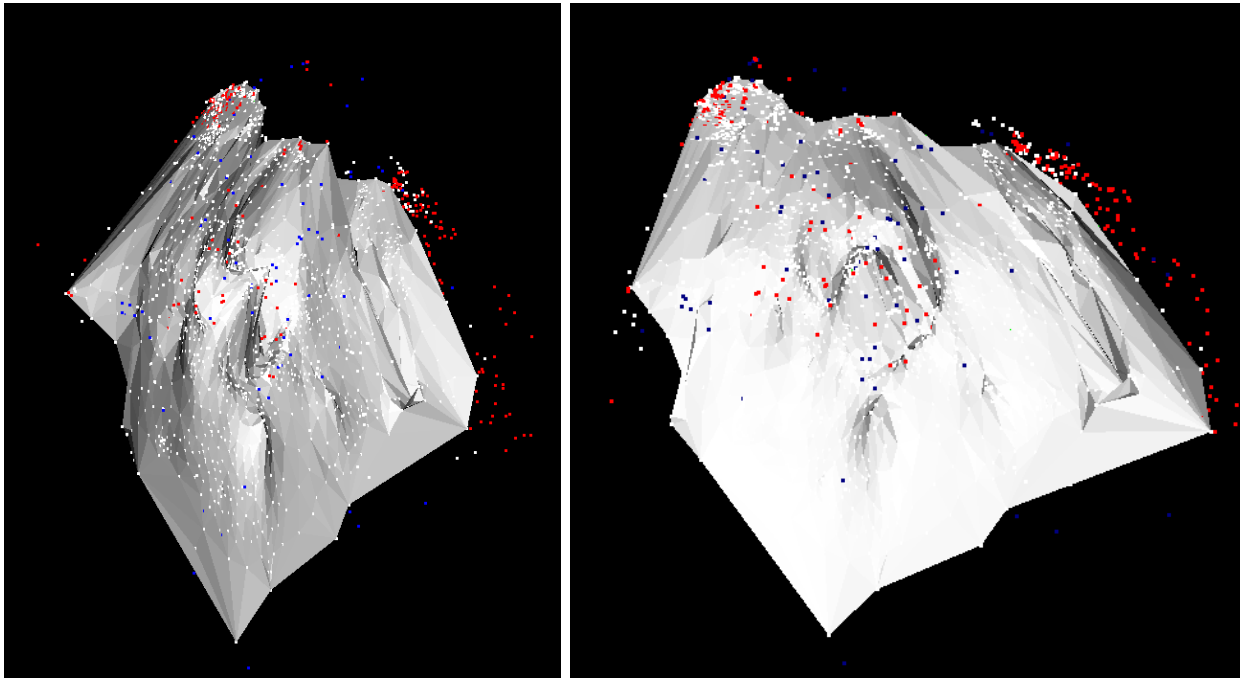


Figura 6.4: Mapa final tras aplicar la etapa de eliminación según ángulo de paralaje y error de mínima mediana. En rojo, puntos eliminados en la etapa de ángulo de paralaje. En azul, puntos eliminados en la etapa de error de mínima mediana.

En la figura 6.5 se puede ver el mapa final que se obtiene de aplicar únicamente la etapa de error de mínima mediana y suprimir la etapa de eliminación según ángulo de paralaje. El resultado muestra que se consiguen eliminar todos los puntos espurios que perturban la escena. Además destacar que, aunque en esta etapa también son eliminados algunos puntos inliers, la tasa de espurios eliminada es mayor que en el caso de aplicar la eliminación por ángulo de paralaje.

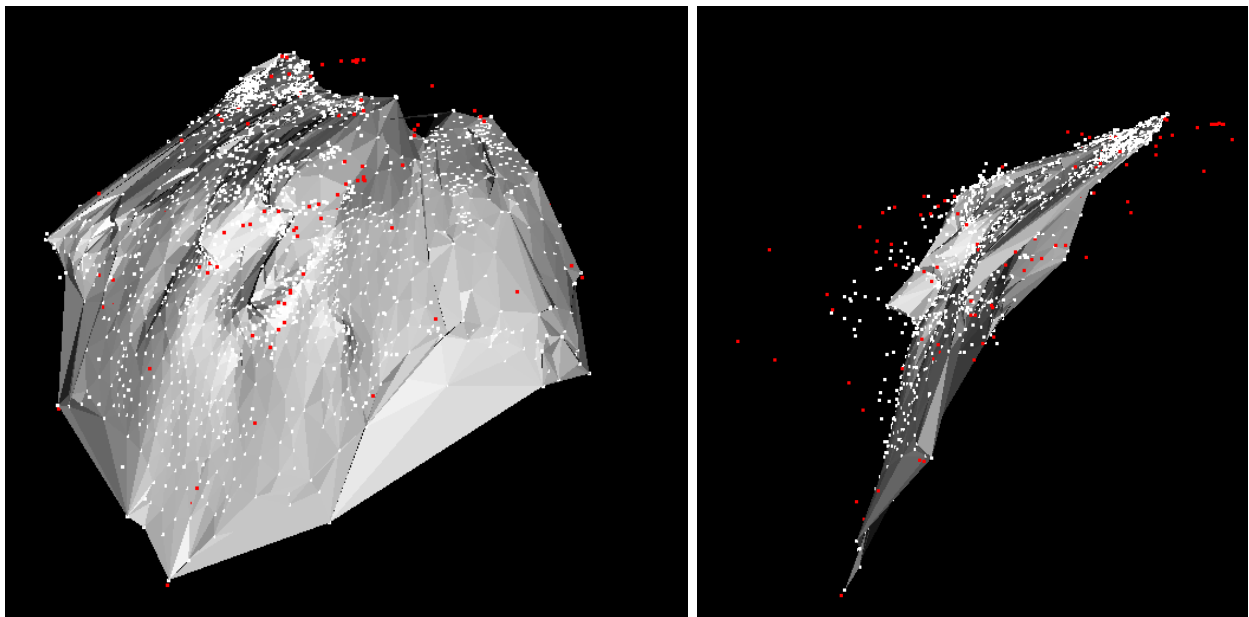


Figura 6.5: Mapa final tras aplicar la etapa de eliminación según ángulo de paralaje. En rojo, los puntos eliminados en la etapa de ángulo de paralaje.

En resumen, aplicando las condiciones de eliminación según error de reproyección y error de mínima mediana en profundidad, es posible garantizar un mapa final libre de espurios con el que conseguir un modelo 3D fotorealista de la escena. Aplicar la etapa de eliminación según ángulo de paralaje, no aporta ningún beneficio ya que reduce la densidad del mapa final, eliminando más puntos inliers que espurios.

6.2. Eliminación de espurios en Tiempo Real

Se comparará el algoritmo original del PTAM con las modificaciones propuestas. Estas mejoras se aplicarán de manera acumulativa de forma que el efecto de la modificación anterior tenga validez sobre la siguiente. El orden en el que se incluirán las mejoras será:

1. Sintonización del umbral para el marcado de una observación espúrea ($\sigma = 2$).
2. Sintonización de la puntuación acumulada de observaciones espurias ($\text{Inlier} > 50 \% \Rightarrow \text{Inlier} > 80 \%$).
3. Adición de la condición de reobservación en la detección y eliminación de espurios (Detectados y Vistos $> 30 \%$).

El objetivo que se persigue con la eliminación en tiempo real es conseguir que todo el peso de la detección y eliminación de espurios recaiga sobre el algoritmo PTAM y no sobre el algoritmo de detección y eliminación a posteriori. De manera que los resultados que se obtengan tras la ejecución de PTAM con las diferentes mejoras, se aproximen lo máximo posible al caso de aplicar el algoritmo a posteriori que es el que asegura un mapa final libre de espurios.

La manera de proceder para en la comprobación de los resultados, será determinar en primer lugar los errores de reproyección de cada uno de los puntos del mapa estimado por PTAM para cada una de las mejoras. En la gráfica 6.6 se puede ver el error de reproyección acumulado para los distintos casos planteados y los errores de reproyección tras la aplicación del algoritmo de eliminación a posteriori. Los errores de reproyección de cada punto corresponden al error máximo del punto con respecto a todos sus emparejamientos.

La curva en color rojo, corresponde al algoritmo PTAM original. Como se puede ver, la curva se encuentra muy desplazada hacia la derecha de lo que sería el caso ideal (curva verde de raya discontinua) que corresponde a los resultados obtenidos tras la aplicación del algoritmo a posteriori. Se puede ver como una reducción en el umbral de marcado de puntos espurios (curva azul) consigue una mejor aproximación al caso de eliminación a posteriori.

Tomando como unidad de referencia comparativa los 2 píxeles de error, se puede ver como el porcentaje de espurios se ve reducido en un 7 % para el caso de sintonización del umbral de marcado. Añadir la sintonización de la puntuación acumulada a la sintonización del umbral de marcado (línea verde continua) presenta una reducción del 15 % de puntos espurios con error mayor a los 2 píxeles. En el caso de la adición de la condición de reobservabilidad (línea negra), se puede ver cómo los resultados que se obtienen son prácticamente idénticos a los del caso anterior, incluso en este caso se puede ver como el porcentaje de puntos con un error pequeño (0.5 píxeles) es menor que en el caso de no aplicar la condición de reobservabilidad.

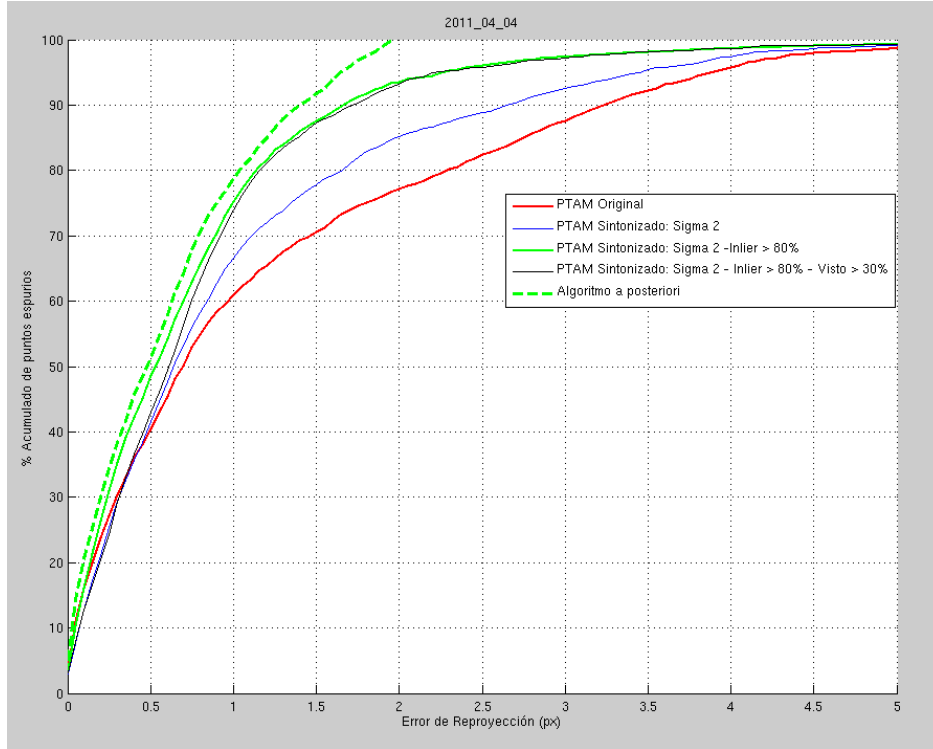


Figura 6.6: Porcentaje acumulado del Error de Reproyección para la secuencia de (b) de la fig. 6.1.

Estas dos últimas propuestas (umbral de puntuación y condición de reobservabilidad), son las que consiguen aproximar mejor la curva obtenida con el caso de eliminación a posteriori. Sin embargo la adición de la condición de reobservabilidad plantea un inconveniente. Además de no verse reducido el porcentaje de espurios eliminados por la condición de reobservabilidad, no se aproxima mejor a la curva del algoritmo de eliminación a posteriori, el número de puntos final del mapa 3D estimado se ve reducido en mayor medida (tabla 6.1).

	PTAM Original	PTAM $\sigma = 2$	Inlier $> 80\%$	Visto $> 30\%$
Mapa Final (ptos)	2548	2143	2124	1758

Tabla 6.1: Tamaño del mapa final estimado tras la aplicación del algoritmo de eliminación a posteriori aplicando los tres criterios de eliminación: error de reproyección, ángulo de paralaje y error de mínima mediana en profundidad.

Esto plantea la posibilidad de que realmente con esta nueva condición de reobservabilidad lo que se esté haciendo sea eliminar más puntos inliers que espurios del mapa. En la figura 6.7 se muestra el mapa final estimado para la secuencia tras la ejecución de PTAM (puntos blancos) y los puntos eliminados durante la ejecución según la condición de reobservabilidad (puntos rojos).

Verdaderamente esta condición elimina puntos que a priori sí son espurios pero, sin embargo, elimina también puntos que pertenecen a la zona interna del defecto herniario sin ser espurios. La fracción de puntos espurios eliminados por esta condición es menor que la fracción de inliers eliminada. El problema que plantea esta condición, como ya se comentó en la sección 4.2, es que puntos que se encuentran en el interior del defecto, debido al movimiento lateral de la cámara en la trayectoria exploratoria, sean no detectados cuando si que son potencialmente visibles en la imagen.

Esta nueva condición no supone una mejora real en los resultados, llegando incluso en algunas ocasiones a obtener unos resultados más contraproducentes para alguna de las secuencias (aumento

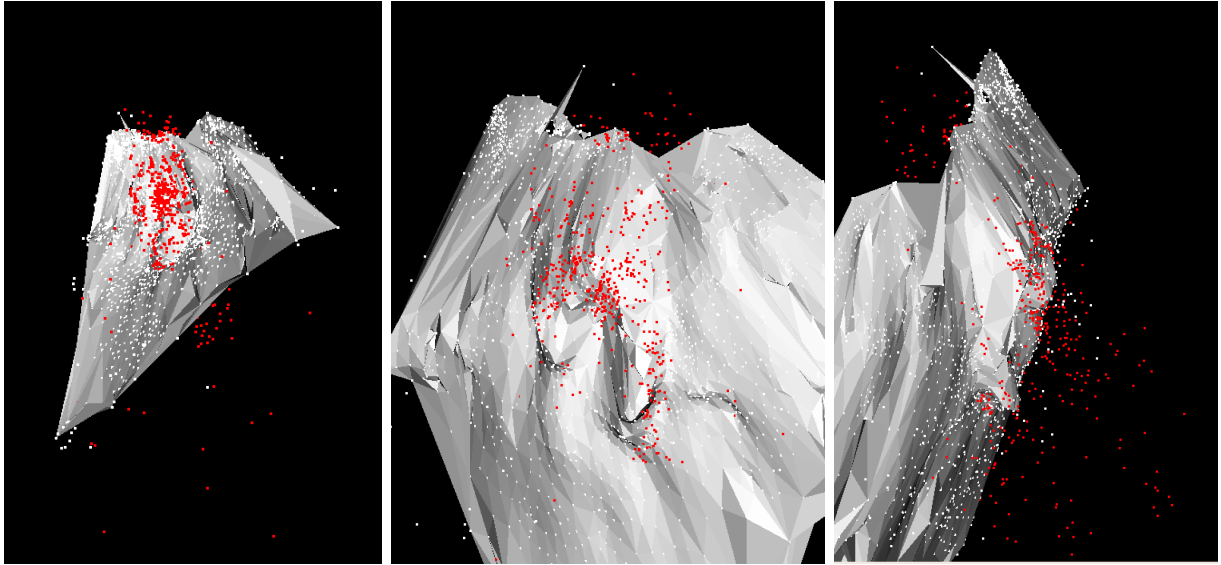


Figura 6.7: Mallado obtenido tras aplicar la condición de reobservabilidad. En blanco aparecen los puntos considerados inliers tras la aplicación del algoritmo a posteriori y que generan el mallado 3D. En rojo los puntos eliminados durante la ejecución según la condición de reobservabilidad.

de los porcentajes de espurios con error mayor a 2 píxeles). Por esa razón y a la vista de los resultados obtenidos, las mejoras que proporcionan una mejor aproximación a la situación idónea en la que se obtendría un mapa libre de espurios que sería tras la aplicación del algoritmo a posteriori, son la sintonización del umbral de marcado y de la puntuación acumulada.

En la tabla 6.2, se recogen los porcentajes de puntos eliminados por cada una de las etapas del algoritmo de eliminación a posteriori y el tamaño de mapa obtenido para las distintas propuestas de mejora en el algoritmo de eliminación en tiempo real.

	% Error > 2px	% Mediana	Mapa Final(ptos)
PTAM Original	23 %	6 %	2951
PTAM $\sigma = 2$	16 %	5 %	2677
Inlier > 80 %	8 %	7 %	2311
Visto > 30 %	8 %	6 %	2033

Tabla 6.2: Porcentajes de puntos espurios tras aplicar las condiciones de error de reproyección y profundidad del punto. Se incluye también el tamaño del mapa obtenido al final de la simulación.

Se puede comprobar como la condición de eliminación de reobservabilidad de los puntos no reduce los porcentajes de espurios eliminados sino que además disminuye la densidad del mapa final.

Comprobados los resultados obtenidos para una de las secuencias de laparoscopia, se lleva a cabo el mismo análisis para el resto de ejecuciones. Los resultados para todas ellas pueden verse en el Anexo A (Resultados y Reconstrucciones).

En la figura 6.8 se recogen los porcentajes de puntos del mapa 3D estimado cuyo error máximo de reproyección es mayor a 2 píxeles. La situación ideal sería obtener un error de reproyección del 0 % para asegurar una correcta efectividad del algoritmo de detección y eliminación en tiempo real según error de reproyección.

La primera barra corresponde al algoritmo original de detección y eliminación de espurios y la segunda barra a la modificación del algoritmo tras una sintonización del umbral de marcado de

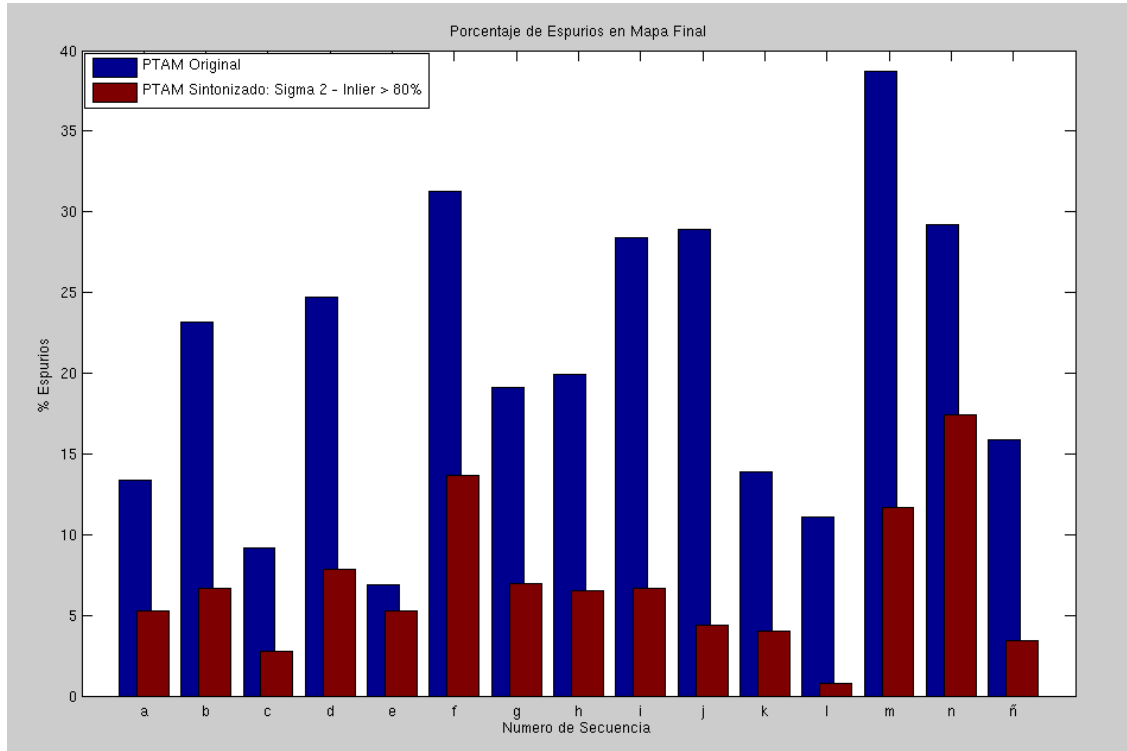


Figura 6.8: Porcentajes de puntos espurios con error de reproyección mayor a 2 píxeles para las distintas secuencias.

espurios y la sintonización de la puntuación acumulada de eliminación que es la solución que mejores resultados genera. Se puede ver cómo el porcentaje se ve reducido para todas y cada una de las secuencias sobre las que se ha analizado.

Si bien es cierto comentar que para las secuencias (e) y (m) se han tenido ciertas dificultades a la hora de procesar las secuencias con el algoritmo PTAM. En el caso de la secuencia (e) para poder obtener un mapa 3D se debe detener la ejecución antes de que acabe, ya que si se permite que el vídeo termine, el algoritmo PTAM se pierde y es imposible obtener un resultado fiable de la estructura de la escena.

En el caso de la secuencia (m) el problema que surge es que su mapa estimado tiene tan baja densidad de puntos que en ocasiones el sistema se pierde y como ocurría en el caso de la secuencia (e) es imposible obtener un mapa final correcto de la escena.

Como resumen general se puede concluir que la mejor solución de entre las distintas mejoras en la detección y eliminación de espurios en tiempo real planteadas es la correcta sintonización del umbral de aceptación de puntos inliers ($\sigma = 2$) y la sintonización de la puntuación acumulada en eliminación (Inlier > 80 %). La nueva condición de reobservación para este tipo de secuencia no aporta mejoras en lo que a la reducción de puntos espurios se refiere y además reduce la densidad de los mapas finales.

6.3. Reconstrucciones 3D

En esta sección se presentan los modelos 3D obtenidos a partir del algoritmo de triangulación implementado. El mapa empleado para su obtención ha sido procesado por el algoritmo de detección y eliminación a posteriori de manera que se pueda asegurar un mapa final libre de espurios.

La figura 6.9 corresponde a la reconstrucción 3D obtenida para la secuencia 6.1(b). En la imagen de la izquierda se puede observar el proceso de texturización seguido para la reconstrucción. Los puntos que se observan en la imagen están coloreados según la escala más fina en la que fueron vistos. La relación de colores en función de la escala en la que fueron cogidos desde la más fina a la más basta es: rojo, azul, verde, morado. Para todas las reconstrucciones fotorealistas únicamente se van a emplear los puntos correspondientes a la escala fina y subfina (rojos y azules) que son aquellos puntos que aportan mayor precisión geométrica.

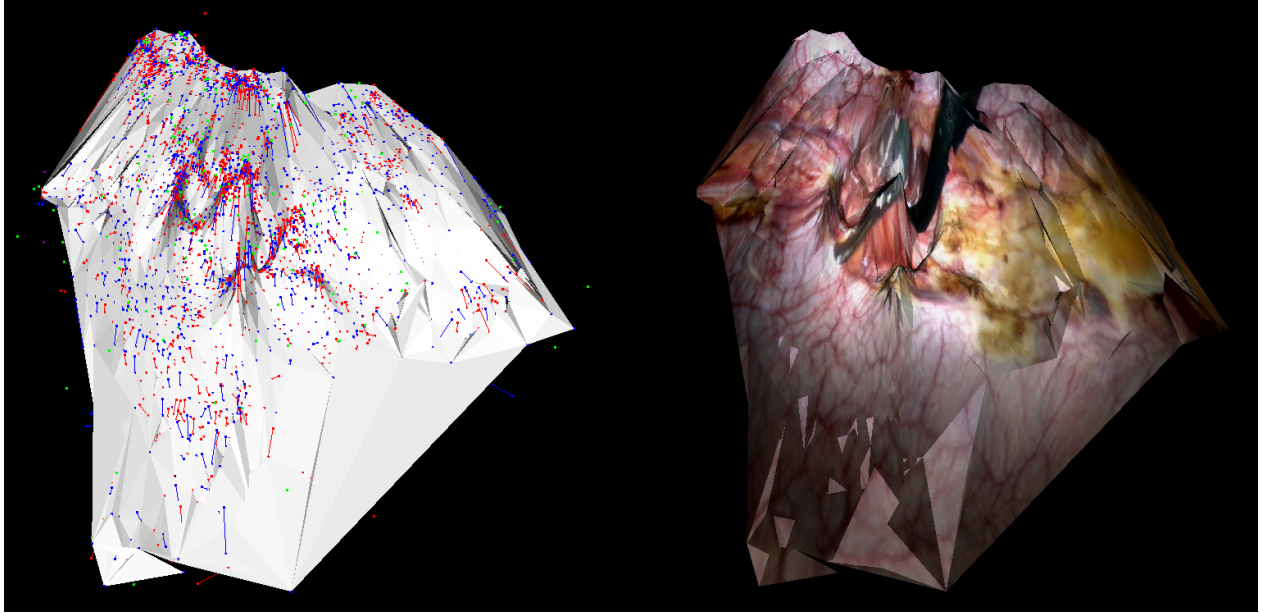


Figura 6.9: Reconstrucción 3D de la secuencia (b) de la fig. 6.1

En la imagen de la izquierda puede verse también el mallado triangular obtenido antes de aplicar la texturización. También puede verse la posición final de los puntos del mapa, una vez se ha aplicado el suavizado Laplaciano a la malla (cap. 5). El desplazamiento que se produce sobre cada punto queda reflejado mediante las líneas que aparecen en la imagen que unen la posición inicial y final de los puntos empleados en la reconstrucción de la escena.

Las líneas generadas durante el paso de suavizado no unen puntos muy distantes del mapa 3D por lo que puede decirse que la eliminación de espurios en tiempo real y el algoritmo de eliminación de espurios a posteriori han eliminado todos los puntos outliers del mapa que pudiera haber.

En la figura de la derecha se puede ver el resultado final de la reconstrucción 3D, aplicada sobre la malla la textura correspondiente de la escena. Como se puede observar el resultado muestra una reconstrucción de la escena 3D muy fidedigna a lo que se visualiza en la secuencia laparoscópica. En la fig. 6.10 se pueden ver otras vistas del modelo 3D obtenido para la cavidad abdominal.

Las reconstrucciones fotorealistas para las diferentes secuencias laparoscópicas se encuentran incluidas en el anexo A. Ha sido posible obtener modelos 3D para todas las secuencias sobre las que ha sido probado el algoritmo de triangulación excepto para 2 de ellas. El tamaño medio de los mapas finales obtenidos libres de espurios es de alrededor del millar de puntos.

Para la secuencia (m) el mapa estimado de la escena contiene una densidad baja de puntos (decenas de puntos) y a la hora de aplicar los métodos de eliminación de espurios del algoritmo de eliminación a posteriori, en concreto el ajuste de error de mínima mediana, se elimina todo el mapa ya que la distancia entre puntos en el mapa es superior al radio de búsqueda creado y no se encuentran suficientes emparejamientos. Además, en ocasiones, el sistema también se pierde debida

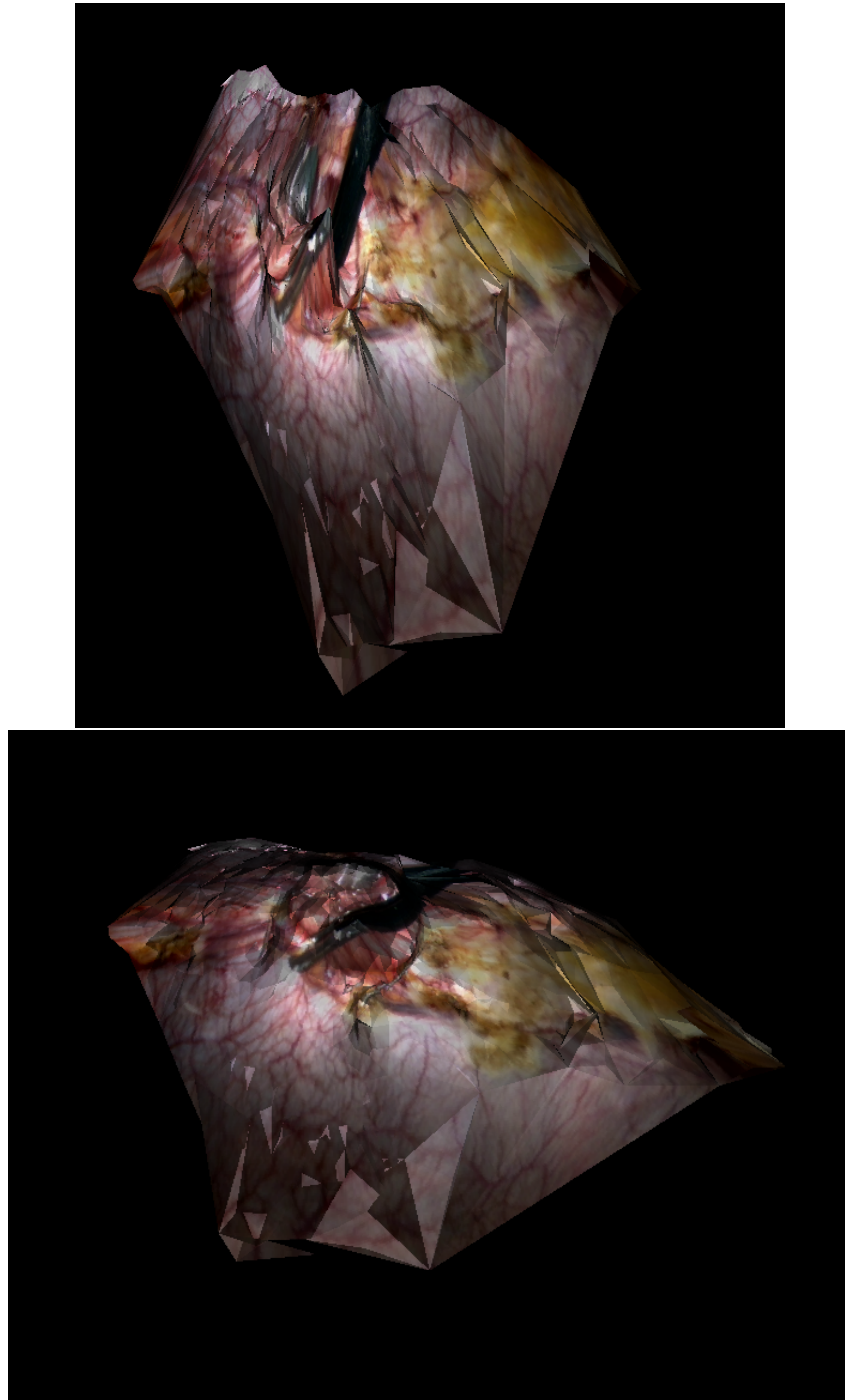


Figura 6.10: Vistas de la reconstrucción final de la secuencia (b) de la fig. 6.1

a esa baja densidad de correspondencias que aparecen entre las imágenes de la secuencia que resulta imposible obtener un mapa 3D de la escena. Estos resultados ocurren tanto en la situación original como para las mejoras propuestas en la eliminación de espurios.

En el caso de la secuencia (e) también existen dificultades a la hora de obtener una reconstrucción fotorealista. Si se permite que toda la secuencia del vídeo sea procesada por PTAM, no es posible obtener un modelo fidedigno ya que el algoritmo se pierde y es incapaz de obtener un mapa de la escena. En cambio si se detiene antes de que esto suceda, es posible obtener la reconstrucción de la cavidad abdominal.

6.4. Tiempos de cómputo

La condición necesaria con la que se trabaja durante todo el proyecto es que el algoritmo PTAM debe estar en todo momento funcionando en tiempo real y a frecuencia de vídeo. Las mejoras propuestas en la reducción de la tasa de espurios en tiempo real implementadas, únicamente varían las condiciones de restricción, y por tanto no se ven afectados los tiempos de cómputo en el análisis. De manera que se puede asegurar que el algoritmo PTAM trabaje a tiempo real y frecuencia de vídeo.

En cuanto al tratamiento del mapa a posteriori, en la siguiente tabla (tabla A.30) se recogen los tiempos de cómputo empleados para las etapas de eliminación según error de reproyección y error de mínima mediana que son las etapas que aseguran un mapa final libre de espurios. La densidad del mapa sobre el que se está trabajando tiene un volumen de 2800 ptos. Factor a tener en cuenta ya que en función del número de puntos en el mapa, se precisa un mayor tiempo de procesado de mapa.

	% Error > 2px	% Error Mínima Mediana	Tiempo Final
Tiempo de cómputo (seg)	1.5	0.33	1.83

Tabla 6.3: Tiempo de cómputo de las diferentes etapas de eliminación de espurios del algoritmo a posteriori.

La etapa del algoritmo de eliminación a posteriori que mayor tiempo emplea es la etapa de eliminación según error de reproyección. El número de emparejamientos que esta etapa debe comprobar para garantizar un mapa final con puntos con error de reproyección menor a 2 píxeles es de 17700 emparejamientos. Una gran cantidad de puntos y emparejamientos a comprobar teniendo en cuenta que se analizan todos y cada uno de ellos.

En cuanto a tiempo en la obtención de la reconstrucción 3D fotorealista, en la tabla 6.4 puede verse el tiempo medio empleado en la generación del modelo 3D para la secuencia con un mapa final libre de espurios de densidad 2300ptos.

	Tiempo de Cómputo (seg.)
Generación Modelo 3D	1.19

Tabla 6.4: Tiempo de cómputo empleado en la generación del modelo 3D de la escena.

La duración de la secuencia de estudio es de 27 seg. Si se realiza un análisis comparativo entre la duración del vídeo de la secuencia con el tiempo empleado en la obtención del modelo 3D se puede afirmar que el algoritmo de triangulación permite obtener un modelo 3D de la escena en tiempo real.

Como se propondrá en el siguiente capítulo (Conclusiones y Trabajos Futuro) en la sección de futuras líneas de investigación, la obtención de estos tiempos de cómputo plantea la posibilidad de incluir el algoritmo de triangulación en la propia ejecución del PTAM.

El tiempo empleado es de en torno a 1 segundo, pero este tiempo se realiza llevando a cabo un procesado completo de toda la secuencia. Si el análisis se realizara iterativamente durante la ejecución el tiempo de cómputo se repartirá por cada frame haciendo posible su incursión durante el tracking y el mapping.

Conclusiones y Trabajo Futuro

7.1. Conclusiones

En este proyecto se presenta un algoritmo que permite obtener mapas estimados de cavidades abdominales libres de espurios a partir de una secuencia real exploratoria de una intervención quirúrgica. Permitiendo obtener reconstrucciones fotorealistas de la escena a partir de la información proporcionada por el algoritmo de SLAM llamado PTAM.

Las principales aportaciones de este trabajo son:

1. Sintonización del algoritmo de detección y eliminación de espurios en tiempo real.
2. Algoritmo de detección y eliminación de espurios a posteriori que asegure un mapa libre de espurios mediante el procesamiento del mapa obtenido por el algoritmo PTAM.
3. Algoritmo de reconstrucción 3D fotorealista a partir de una secuencia de KFs y el mapa 3D estimado de la escena.

Todas ellas relacionadas ya que para obtener un modelo fotorealista de la escena es preciso un mapa 3D libre de espurios y, como consecuencia, que tanto el algoritmo de detección y eliminación de outliers implementado sobre el algoritmo en tiempo real como el algoritmo de eliminación funcionen correctamente para obtener un mapa libre de espurios.

Una correcta sintonización del algoritmo original de detección y eliminación de espurios permite obtener mapas estimados de la escena con menor tasa de puntos espurios en tiempo real y a frecuencia de vídeo. Ser más restrictivos a la hora de aceptar puntos inliers (sintonización de la puntuación acumulada) así como reducir el threshold en la detección de puntos espurios (sintonización del umbral de marcado de espurios) mejora la calidad del mapa. Este aspecto a su vez mejora la estimación de la trayectoria en menor medida ya que la función de influencia robusta maneja menor tasa de espurios durante el Bundle Adjustment tanto del mapping como del tracking mejorando la optimización de la esta función.

Haber obtenido resultados favorables para todas las secuencias sobre las que ha sido posible aplicar el algoritmo PTAM permite concluir que el algoritmo implementado puede generalizarse para cualquier tipo de secuencia con endoscopio. Resultaría interesante verificar si es posible emplearlo en otro tipo de escenarios con características diferentes y en los que la escena también sea rígida.

PTAM realiza una eliminación a tiempo real en función del frame en el que se encuentra por lo que existe siempre la posibilidad de que un punto sea creado y no vuelva a ser observado en ningún otro frame. Si esto ocurre el algoritmo de detección y eliminación no lo considera en su análisis y el punto nunca es eliminado del mapa. Por esta razón se aconseja el empleo del algoritmo de eliminación de espurios a posteriori que permita detectar este tipo de puntos y eliminarlos del mapa ya que pueden perturbar la escena mediante un procesamiento de todo el mapa estimado. Su implementación

asegura la obtención de mapas finales libres de espurios tras la ejecución del algoritmo PTAM.

A partir de un mapa libre de espurios, se ha podido comprobar que existe la posibilidad de crear un modelo 3D fotorealista de la escena observada por el cirujano durante la intervención. Como punto adicional decir que los tiempos de cómputo empleados para obtener las reconstrucciones 3D, son en torno al segundo para mapas del orden del millar de puntos y duración media de la secuencias analizadas de 30 segundos por lo que haciendo una comparación entre los tiempos empleados para la reconstrucción y la ejecución, se puede concluir que el algoritmo reconstrucción es capaz de trabajar en tiempo real al obtenerse unos tiempo equicomparables a los manejados en el procesado de la secuencia.

Este hecho posibilitaría la inclusión del código implementado para la reconstrucción 3D en el algoritmo PTAM de manera que se obtenga el modelo fotorealista al mismo tiempo que se obtiene el mapa estimado de la escena.

7.2. Líneas de investigación futuras

De este proyecto se extraen algunas posibles líneas de investigación que tendrían como objetivo a corto o medio plazo el desarrollo de una aplicación de utilidad médico-quirúrgica:

1. Obtención de una triangulación 3D más precisa y homogénea. Uno de los problemas que se tienen con el método empleado para la reconstrucción 3D es la superposición de triángulos cuando los puntos a incluir no se encuentran en el interior de la primera triangulación Delaunay. El problema viene de no obtener los emparejamientos de todos los puntos que son vistos sobre el KF. Una posibilidad que se plantea sería la de incluir inicialmente sobre cada KF todos los puntos del mapa que son vistos y a partir de ahí llevar a cabo una triangulación progresiva que fuera incluyéndolos en función de la triangulación inicial.
2. Comprobar que la sintonización propuesta para la detección y eliminación de espurios se puede generalizar a cualquier ambiente de trabajo. En este proyecto, únicamente se ha testado su funcionamiento sobre secuencias de cirugía de hernia ventral en humanos donde la escena es rígida, por lo que sería conveniente comprobar si el modelo que aquí se plantea puede aplicarse a otro tipo de escenarios y así verificar que el modelo que se plantea es útil y generalizable a todo tipo de ambientes.
3. Inclusión de elementos de realidad aumentada y virtual sobre la escena. Una posibilidad sería la inclusión de medidas entre puntos de la cavidad. Esto permitiría conocer con precisión, por ejemplo el tamaño del defecto a cubrir, sin necesidad de realizar una intervención más invasiva.

Actualmente los métodos empleados conllevan un tiempo en la toma de medidas muy elevado y no son muy precisos debido a la dificultad en el acceso a la zona así como la carencia de percepción de profundidad que se tiene con el endoscopio. Sería posible hacer una aproximación a la forma del defecto (aproximación elíptica) que permitiera conocer el tamaño del mismo. Su visualización sobre la reconstrucción 3D permitiría conocer su relieve y su forma real. Una aplicación podría ser para conocer el tamaño del parche que debería utilizarse para ocluir el defecto reduciendo costes en material que surgirían de aproximar el tamaño con el procedimiento actual [26, 27].

4. Posibilidad de reconocimiento de los utensilios quirúrgicos (pinzas y agujas) durante la secuencia a tiempo real. Esto resultaría especialmente útil para la reconstrucción fotorealista ya que sería posible eliminar aquellos puntos que pertenezcan a las herramientas quirúrgicas, permitiendo una representación más fidedigna de la cavidad abdominal.

5. Implementación a frecuencia de vídeo del algoritmo de reconstrucción de escenas 3D. Los tiempos para la creación de mallas y su posterior reconstrucción fotorealista son equicomparables a los tiempo empleados para el Tracking y el Mapping por lo que resultaría interesante como posible alternativa, incluir el algoritmo de generación de modelos 3D fotorealistas sobre el algoritmo PTAM a frecuencia de vídeo para conocer in situ la forma de la cavidad y sus medidas al mismo tiempo que se realiza el procedimiento exploratorio de la zona a intervenir.
6. Aplicación de métodos variacionales [28] sobre este tipo de escenas rígidas. Los mapas generados por el algoritmo PTAM son del orden del millar de puntos. Con este tipo de algoritmos se permitirían obtener mapas del orden del millón del puntos, consiguiendo mapas más densos que aproximasen la estructura de la escena, en este caso concreto la pared abdominal durante la cirugía de hernia ventral.

Bibliografía

- [1] Klein, G., Murray, D.: Parallel Tracking and Mapping for Small AR Workspaces. In: Symposium on Mixed and Augmented Reality (ISMAR). (2007).
- [2] F. Devernay and D. Faugeras. Straight lines have to be straight. Machine Vision and Applications, 13(1):14-24, 2001.
- [3] P. Huber. Robust Statistics. Wiley, 1981.
- [4] B Triggs, P McLauchlan, R Hartley, A Fitzgibbon: Bundle adjustment, A modern synthesis. Vision Algorithms: Theory and Practice, 2000 . Springer.
- [5] Zoltan Csaba Marton and Radu Bogdan Rusu and Michael Beetz: On Fast Surface Reconstruction Methods for Large and Noisy Datasets, Proceedings of the IEEE International Conference on Robotics and Automation (ICRA 2009), Mayo 2009.
- [6] Michael Kazhdan, Matthew Bolitho, Hugues Hoppe, "Poisson surface reconstruction", SGP '06 Proceedings of the fourth Eurographics symposium on Geometry processing
- [7] Civera, J., Davison, A.J., Magallón, J.A., Montiel, J.M.M.: Drift-free real-time sequential mosaicing. Int. Journal of Computer Vision 51(2) (2009) 128–137
- [8] O. Grasa, J. Civera, A. Güemes, V. Muñoz, and M. J.M.M., "EKF monocular slam 3d modeling, measuring and augmented reality from endoscope image sequences," in 5th Workshop on Augmented Environments for Medical Imaging including Augmented Reality in Computer-Aided Surgery.(MICCAI 2009), 2009.
- [9] Hartley R.I y Zisserman A. 'Multiple View Geometry', pages 325:338, Cambridge University Press. Second Edition. 2004.
- [10] J. Civera, A. J. Davison, and J.M.M. Montiel. Inverse depth parametrization for monocular slam. 24(5), 2008
- [11] Rousseeuw P.J. (1987). Robust Regression and Outlier Detection. Wiley Sons.
- [12] The VTK website: <http://www.vtk.org>
- [13] W. Schroeder, K. Martin and B. Lorensen. Visualization Toolkit: An Object-Oriented Approach to 3D Graphics, 4th Edition, 2006.
- [14] E. Rosten and T. Drummond. Machine learning for high-speed corner detection. In Proc. 9th European Conference on Computer Vision (ECCV'06), Graz, May 2006.
- [15] Foley, van Dam, van Dam, Feiner, Hughes. Computer Graphics: Principles and Practice in C (2nd Edition). 1995.

- [16] N. Bray. Notes on Mesh Smoothing. 2004.
- [17] Desbrun, M., Meyer, M., Schroder, P., and Barr, A. H. 1999. Implicit Fairing of Irregular Meshes Using Diffusion and Curvature Flow. In *Proceedings of SIGGRAPH 99*, 317–324.
- [18] Andrew Nealen , Takeo Igarashi , Olga Sorkine , Marc Alexa, Laplacian mesh optimization, *Proceedings of the 4th international conference on Computer graphics and interactive techniques in Australasia and Southeast Asia*, November 29-December 02, 2006, Kuala Lumpur, Malaysia [doi>10.1145/1174429.1174494]
- [19] Fisher, J., Visualizing the connection among convex hull, voronoi diagram and delaunay triangulation, in: *37th Midwest Instruction and Computing Symposium*, 2004
- [20] De Berg, van Kreveld, Overmars, Schwarzkopf. *Computational Geometry: Algorithms and Applications*. 2nd edition, Springer-Verlag. ISBN 3540656200.
- [21] A. M. Andrew, "Another Efficient Algorithm for Convex Hulls in Two Dimensions", *Info. Proc. Letters* 9, 216-219 (1979)
- [22] D. Koppel, C.-I. Chen, Y.-F. Wang, L. H., J. Gu, A. Poirson, and R. Wolters, "Toward automated model building from video in computer assisted diagnoses in colonoscopy," in *Proceedings of the SPIE Medical Imaging Conference*, 2007.
- [23] Stoyanov, D., Darzi, A., Yang, G.Z.: A practical approach towards accurate dense 3d depth recovery for robotic laparoscopic surgery. *Computer Aided Surgery* (2005) 199–208
- [24] Mourgues, F., Devernay, F., Coste-Mani'ère, E.: 3D reconstruction of the operating field for image overlay in 3D endoscopic surgery. In: *IEEE/ACM Symp. Augmented. Reality*. (2001) 191–192
- [25] Luis Manuel Arteaga. *Triangulaciones de nubes de puntos en Java*. 2002.
- [26] I. Gil, J. Marín, J.M. Martínez, E. Bernal, S. Casado, Óscar G. Grasa, J. Quintana. *Augmented Reality and 3D Measurement for Monocular Laparoscopic Abdominal Wall Hernia Repair*. 2011 European Society for Surgical Research (ESSR2011), Aachen (Germany), 2011.
- [27] Óscar G. Grasa, Javier Civera, J. M. M. Montiel. *EKF Monocular SLAM with Relocalization for Laparoscopic Sequences*. 2011 IEEE International Conference on Robotics and Automation (ICRA2011), Shanghai (China), 2011.
- [28] Richard A. Newcombe, Steven Lovegrove and Andrew J. Davison. DTAM: Dense Tracking and Mapping in Real-Time. *ICCV 2011* *(The associated live demonstration won ICCV Best Demo)*
- [29] H. Stewénius, C. Engels, and D. Nistér. Recent developments on direct relative orientation. *ISPRS Journal of Photogrammetry and Remote Sensing*, 60:284–294, June 2006.
- [30] Sameer Agarwal, Yasutaka Furukawa, Noah Snavely, Brian Curless, Steven M. Seitz and Richard Szeliski *Communications of the ACM*, Vol. 54, No. 14, Pages 105-112, October 2011.
- [31] Yasutaka Furukawa and Jean Ponce. *Accurate, Dense, and Robust Multi-View Stereopsis*. *PA-MI*, August 2010.

Apéndice A

Resultados y Reconstrucciones

El propósito de este anexo es detallar los resultados obtenidos para las distintas secuencias laparoscópicas del estudio. Cada una de las secciones del apéndice ha sido nombrada con la fecha en la que la toma de imágenes se realizó. De tal forma que se pueda verificar que todas ellas han sido extraídas de intervenciones reales.

Todas ellas tienen la misma estructura en las que se puede ver:

1. Una selección de frames característicos de la secuencia laparoscópica.
2. Tabla resumen con los porcentajes de de puntos eliminados con el algoritmo final de filtrado de espurios para cada uno de los sucesos planteados en la sintonización del algoritmo de detección y eliminación de espurios en tiempo real.
3. Tiempos de cómputo del algoritmo de eliminación a posteriori y de tiempos de reconstrucción 3D de la escena.
4. Gráfica resumen de los porcentajes de error que se obtienen al finalizar la ejecución para las distintas mejoras propuestas en comparación con la situación ideal de uso del algoritmo de eliminación a posteriori.
5. Vistas de la reconstrucción 3D obtenida tras la aplicación del algoritmo de triangulación.

A.1. Secuencia 2011-02-28

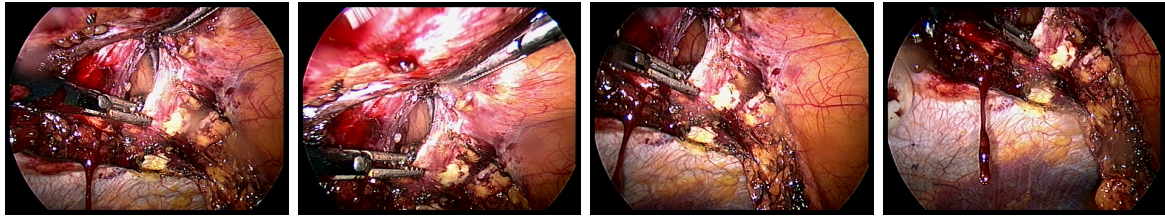


Figura A.1: Frames característicos seleccionados de la secuencia (a) de la fig. 6.1.

	% Error > 2px	% Mediana	Mapa Final(ptos)
PTAM Original	13 %	6 %	6098
PTAM $\sigma = 2$	6 %	5 %	6300
Inlier > 80 %	5 %	6 %	5562
Visto > 30 %	11 %	7 %	2755

Tabla A.1: Porcentaje de puntos eliminados aplicando las etapas de eliminación según error de reproyección y error de mínima mediana. Secuencia (a) fig. 6.1.

	Algoritmo Eliminación Posteriori		Algoritmo Triangulación
	% Error > 2px	% Error Mínima Mediana	Modelo 3D
Tiempo cómputo (seg)	7.1	1.3	9.8

Tabla A.2: Tiempo de cómputo de las diferentes etapas de eliminación de espurios del algoritmo a posteriori y de obtención del modelo 3D de la escena. Secuencia (a) fig. 6.1.

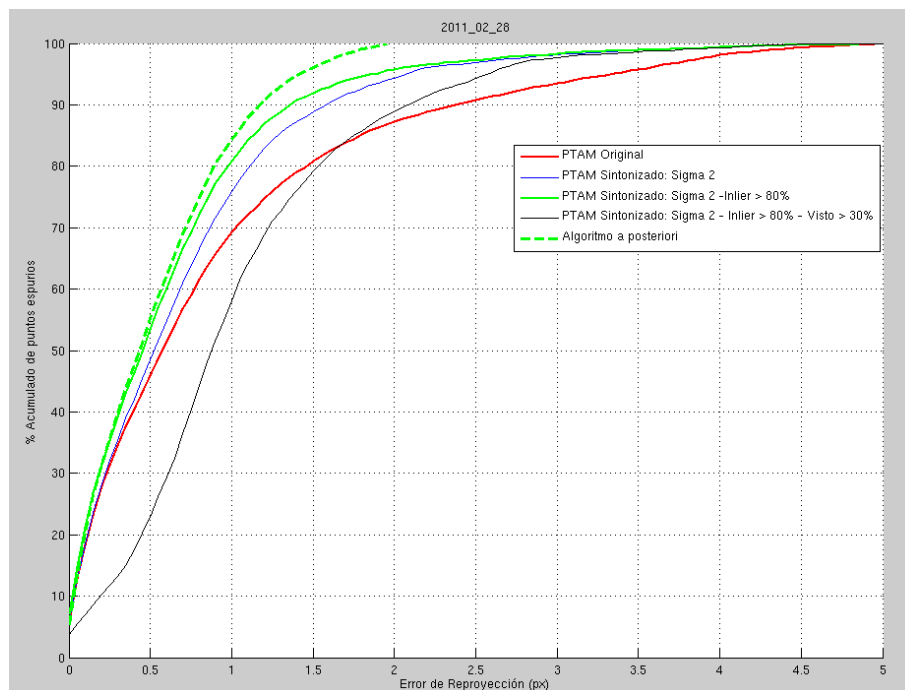


Figura A.2: Porcentaje acumulado del Error de Reproyección para la secuencia de (a) de la fig. 6.1.

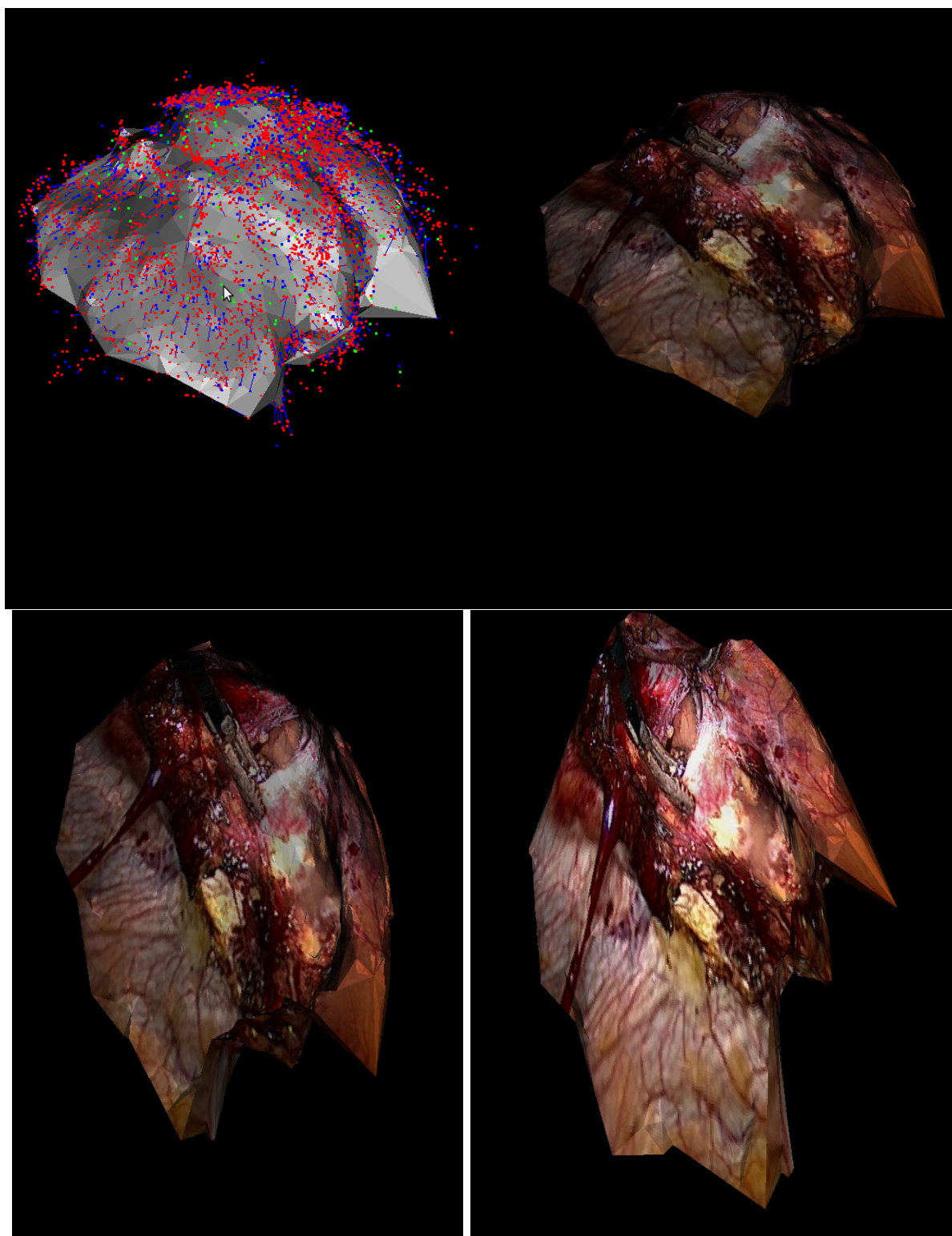


Figura A.3: Vistas de la reconstrucción final de la secuencia (a) de la fig. 6.1

A.2. Secuencia 2011-04-04

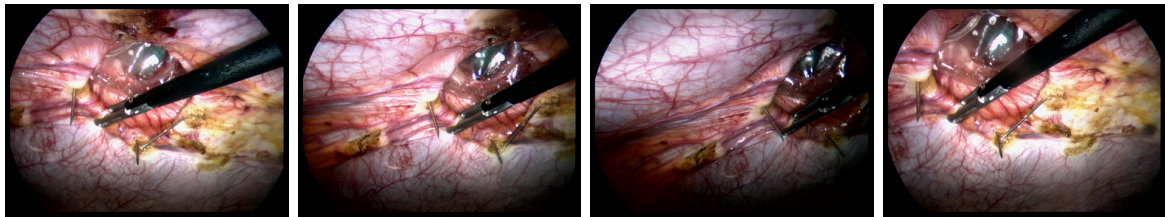


Figura A.4: Frames característicos seleccionados de la secuencia (b) de la fig. 6.1.

	% Error > 2px	% Mediana	Mapa Final(ptos)
PTAM Original	23 %	6 %	2951
PTAM $\sigma = 2$	16 %	5 %	2677
Inlier > 80 %	8 %	7 %	2311
Visto > 30 %	8 %	6 %	2033

Tabla A.3: Porcentaje de puntos eliminados aplicando las etapas de eliminación según error de reproyección y error de mínima mediana. Secuencia (b) fig. 6.1.

	Algoritmo Eliminación Posteriori		Algoritmo Triangulación
	% Error > 2px	% Error Mínima Mediana	Modelo 3D
Tiempo cómputo (seg)	1.5	0.33	1.19

Tabla A.4: Tiempo de cómputo de las diferentes etapas de eliminación de espurios del algoritmo a posteriori y de obtención del modelo 3D de la escena. Secuencia (b) fig. 6.1.

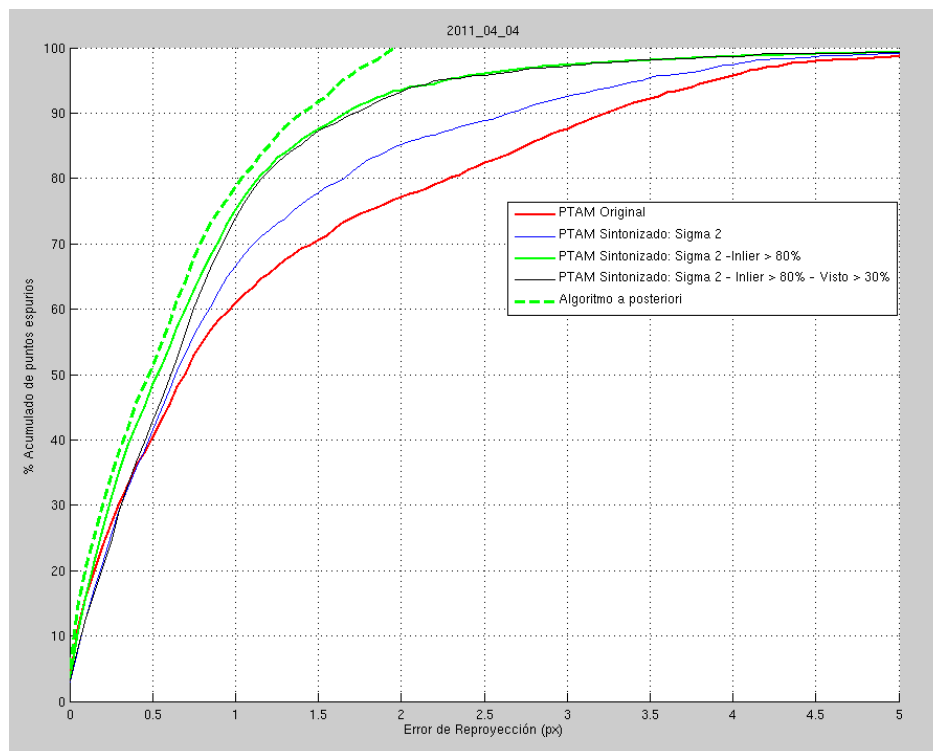


Figura A.5: Porcentaje acumulado del Error de Reproyección para la secuencia de (b) de la fig. 6.1.

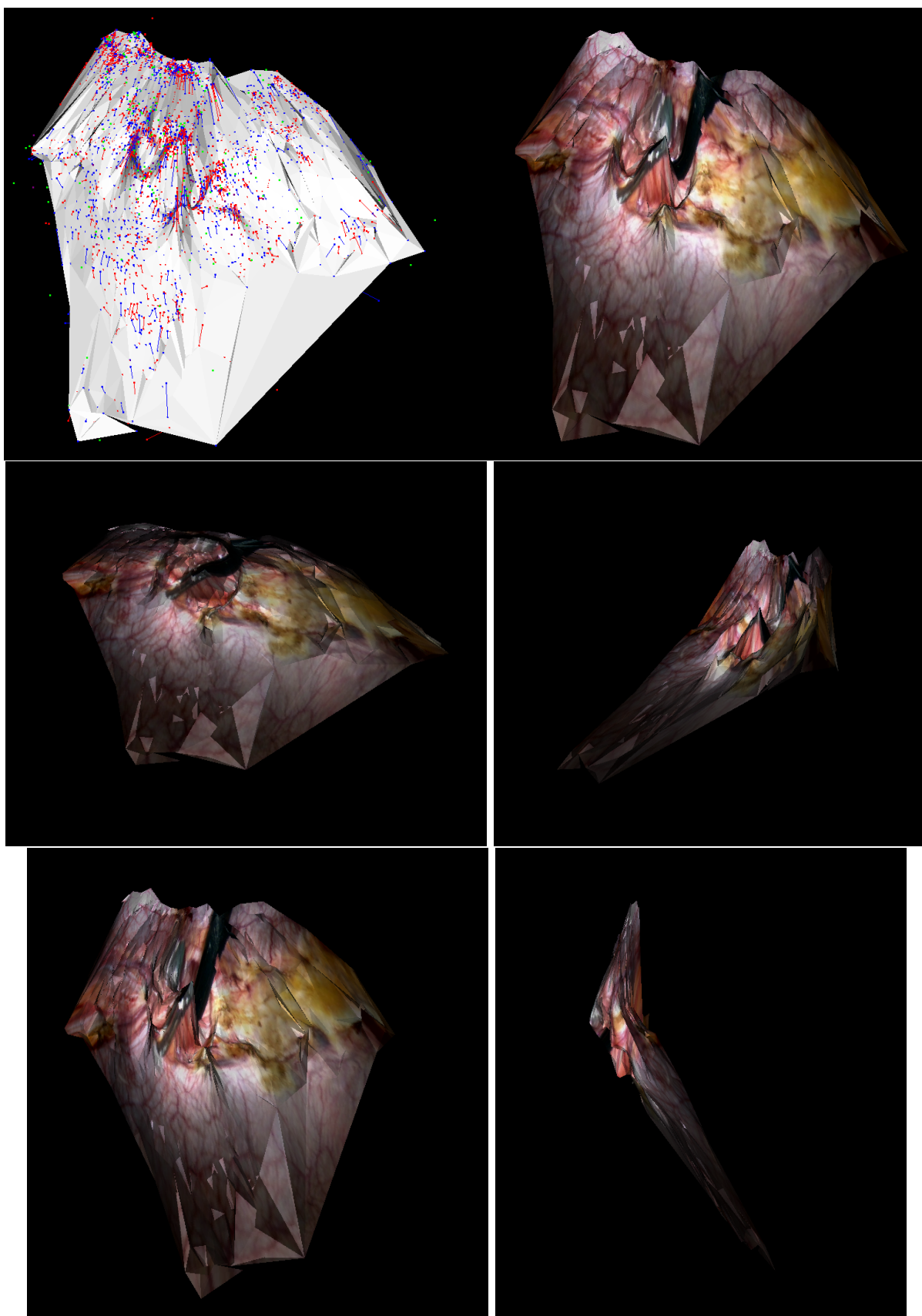


Figura A.6: Vistas de la reconstrucción final de la secuencia (b) de la fig. 6.1

A.3. Secuencia 2011-09-27

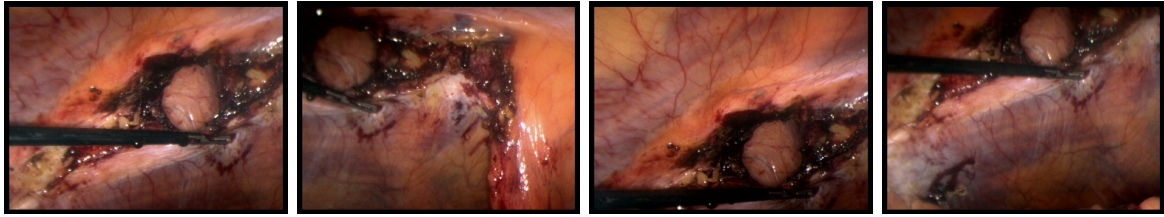


Figura A.7: Frames característicos seleccionados de la secuencia (c) de la fig. 6.1.

	% Error > 2px	% Mediana	Mapa Final(ptos)
PTAM Original	9 %	3 %	1350
PTAM $\sigma = 2$	5 %	3 %	1333
Inlier > 80 %	3 %	3 %	1107
Visto > 30 %	3 %	3 %	1037

Tabla A.5: Porcentaje de puntos eliminados aplicando las etapas de eliminación según error de reproyección y error de mínima mediana. Secuencia (c) fig. 6.1.

	Algoritmo Eliminación Posteriori		Algoritmo Triangulación
	% Error > 2px	% Error Mínima Mediana	Modelo 3D
Tiempo cómputo (seg)	0.29	0.09	0.34

Tabla A.6: Tiempo de cómputo de las diferentes etapas de eliminación de espurios del algoritmo a posteriori y de obtención del modelo 3D de la escena. Secuencia (c) fig. 6.1.

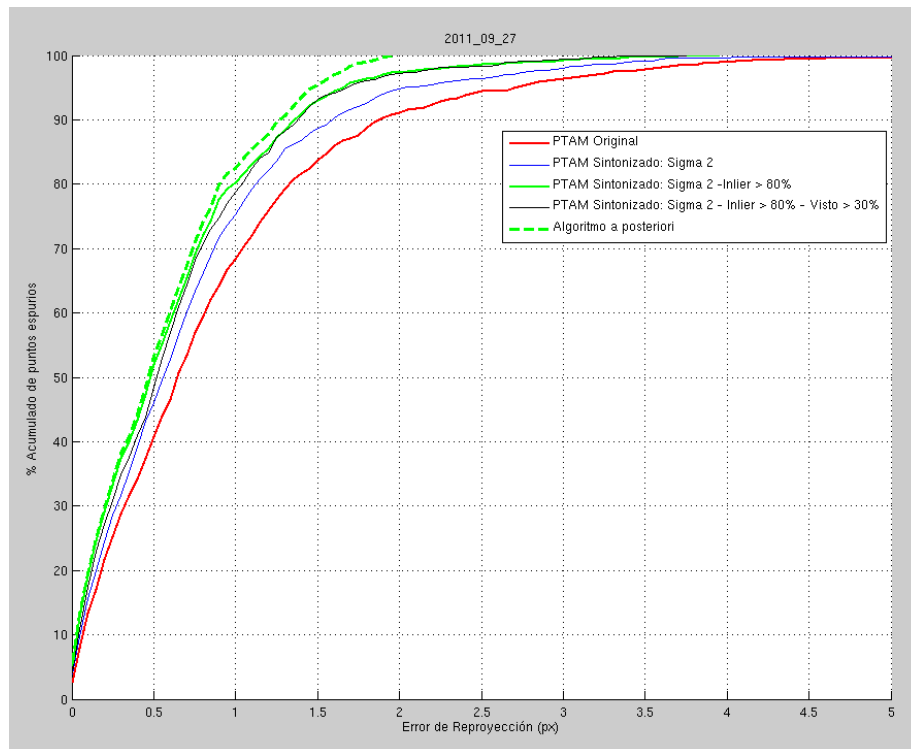


Figura A.8: Porcentaje acumulado del Error de Reproyección para la secuencia de (c) de la fig. 6.1.

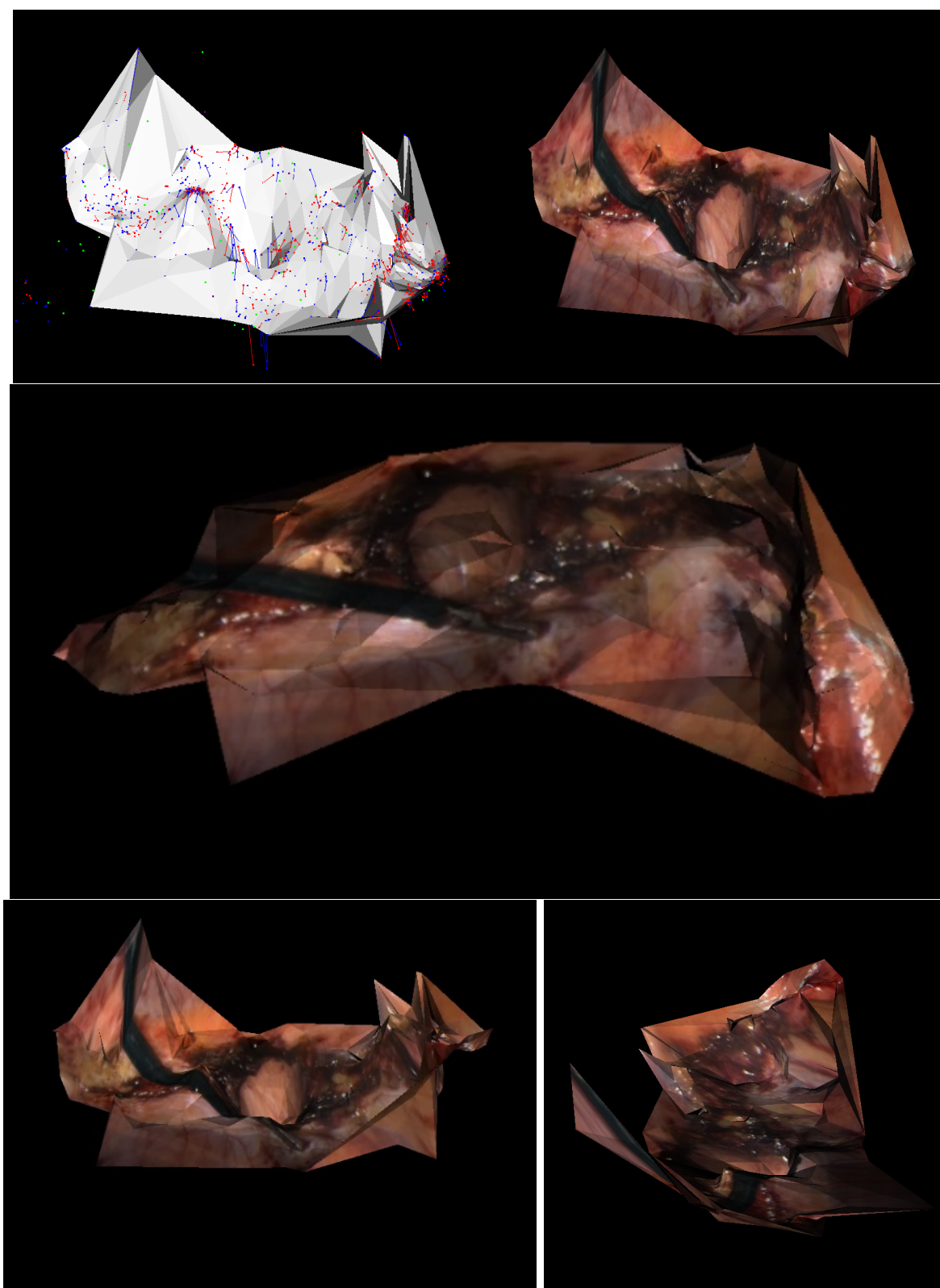


Figura A.9: Vistas de la reconstrucción final de la secuencia (c) de la fig. 6.1

A.4. Secuencia 2011-10-03

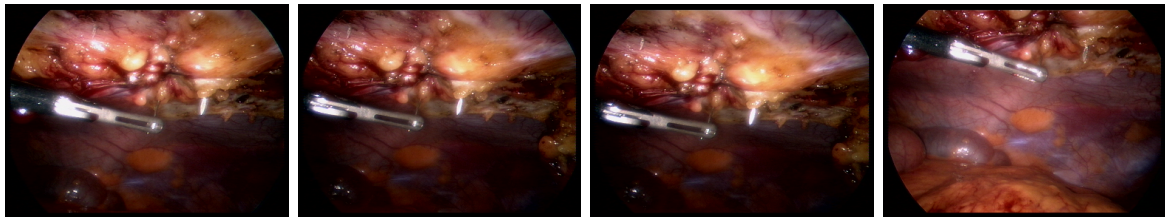


Figura A.10: Frames característicos seleccionados de la secuencia (d) de la fig. 6.1.

	% Error>2px	% Mediana	Mapa Final(ptos)
PTAM Original	25 %	13 %	871
PTAM $\sigma = 2$	12 %	12 %	875
Inlier > 80 %	8 %	14 %	745
Visto > 30 %	7 %	12 %	665

Tabla A.7: Porcentaje de puntos eliminados aplicando las etapas de eliminación según error de reproyección y error de mínima mediana. Secuencia (d) fig. 6.1.

	Algoritmo Eliminación Posteriori		Algoritmo Triangulación
	% Error>2px	% Error Mínima Mediana	Modelo 3D
Tiempo cómputo (seg)	0.22	0.06	0.27

Tabla A.8: Tiempo de cómputo de las diferentes etapas de eliminación de espurios del algoritmo a posteriori y de obtención del modelo 3D de la escena. Secuencia (d) fig. 6.1.

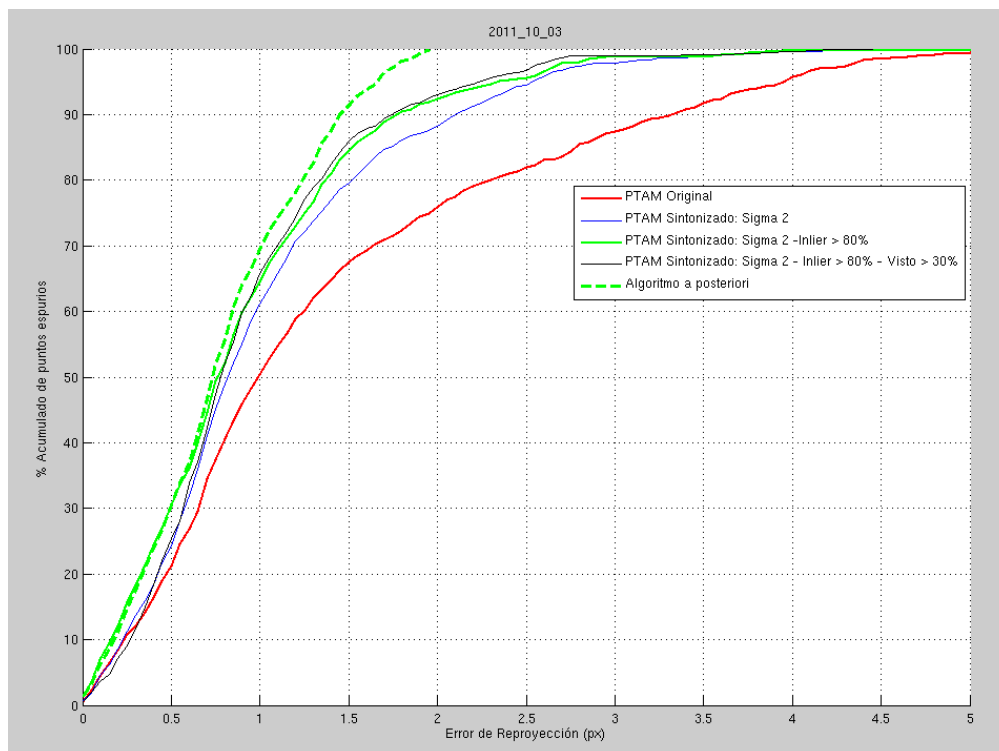


Figura A.11: Porcentaje acumulado del Error de Reproyección para la secuencia de (d) de la fig. 6.1.

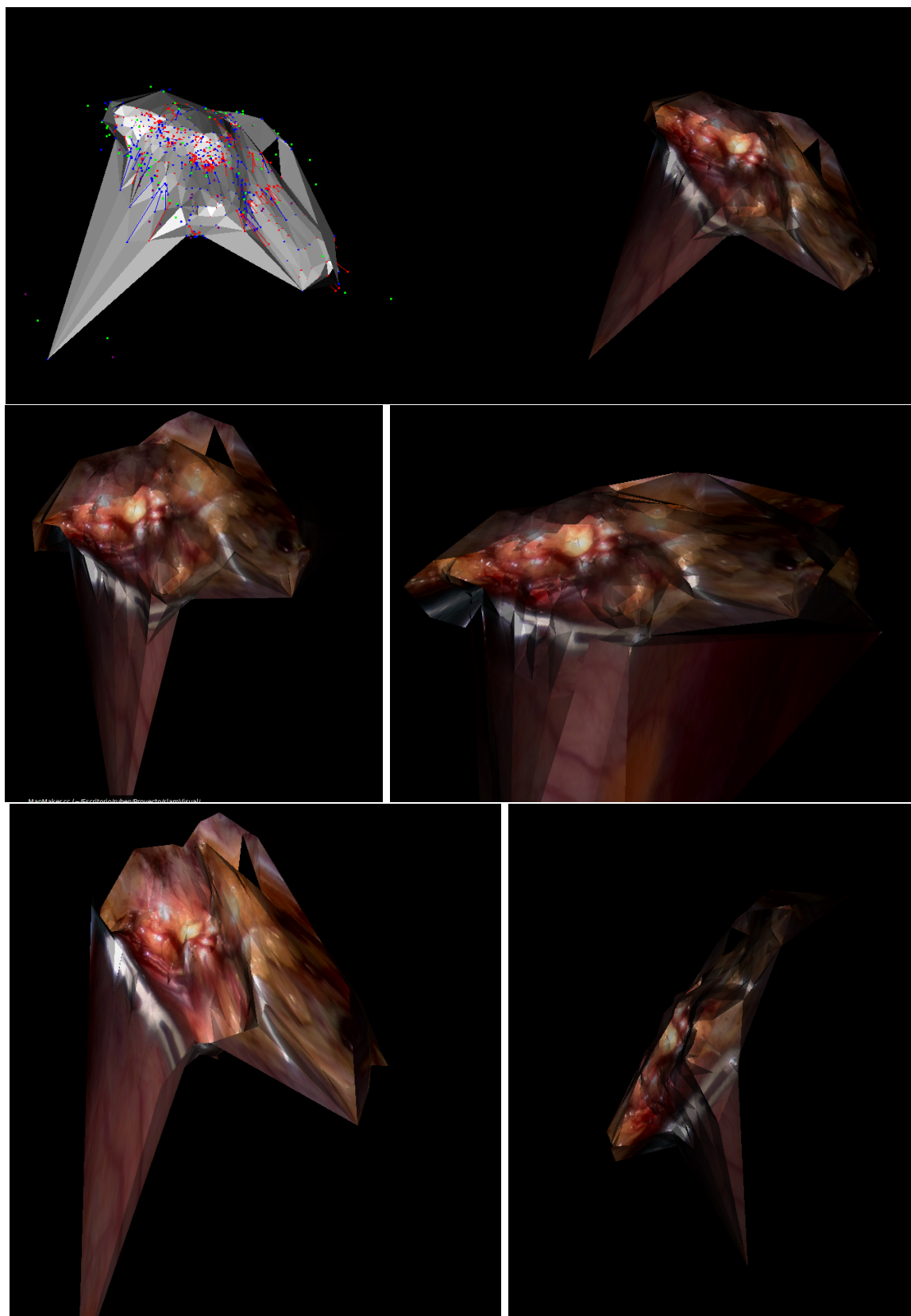


Figura A.12: Vistas de la reconstrucción final de la secuencia (d) de la fig. 6.1

A.5. Secuencia 2011-10-17

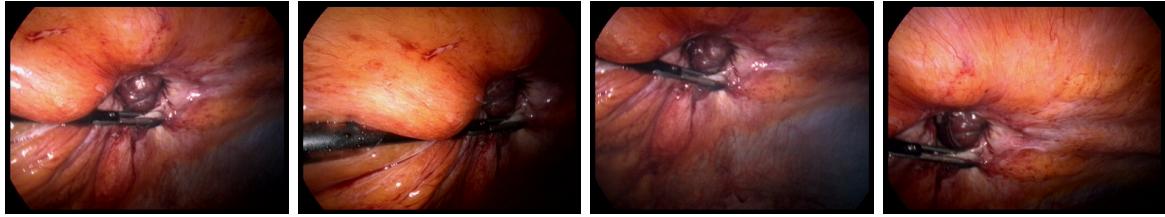


Figura A.13: Frames característicos seleccionados de la secuencia (e) de la fig. 6.1.

	% Error > 2px	% Mediana	Mapa Final(ptos)
PTAM Original	7 %	16 %	388
PTAM $\sigma = 2$	5 %	18 %	426
Inlier > 80 %	5 %	8 %	431
Visto > 30 %	5 %	20 %	144

Tabla A.9: Porcentaje de puntos eliminados aplicando las etapas de eliminación según error de reproyección y error de mínima mediana. Secuencia (e) fig. 6.1.

	Algoritmo Eliminación Posteriori		Algoritmo Triangulación
	% Error > 2px	% Error Mínima Mediana	Modelo 3D
Tiempo cómputo (seg)	0.03	0.03	0.05

Tabla A.10: Tiempo de cómputo de las diferentes etapas de eliminación de espurios del algoritmo a posteriori y de obtención del modelo 3D de la escena. Secuencia (e) fig. 6.1.

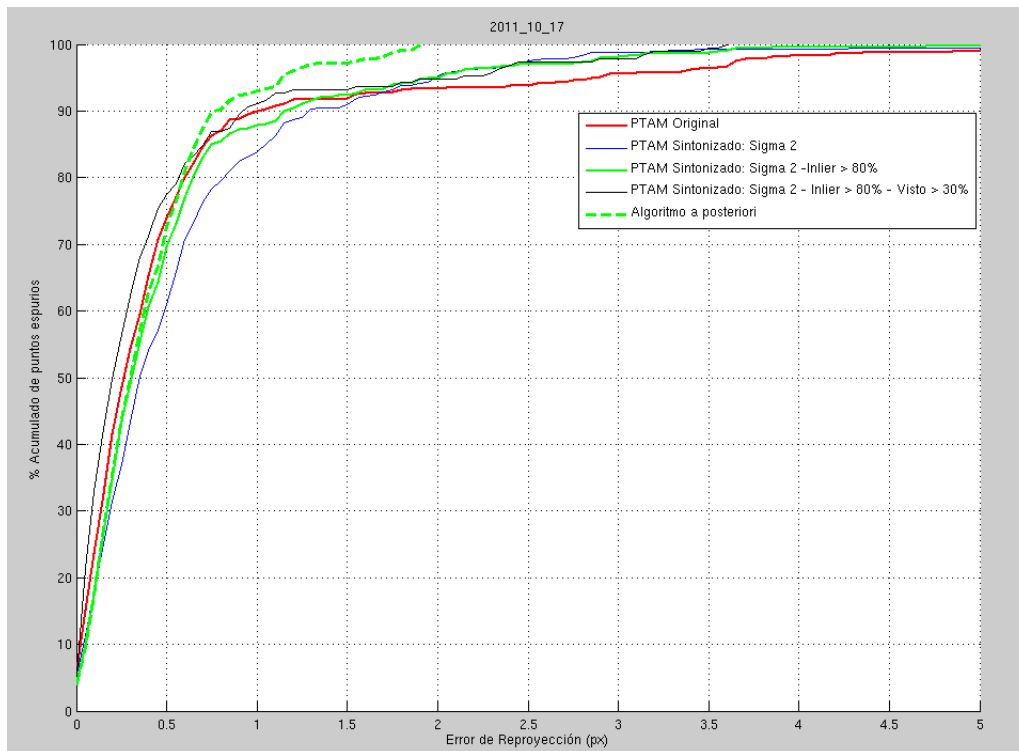


Figura A.14: Porcentaje acumulado del Error de Reproyección para la secuencia de (e) de la fig. 6.1.

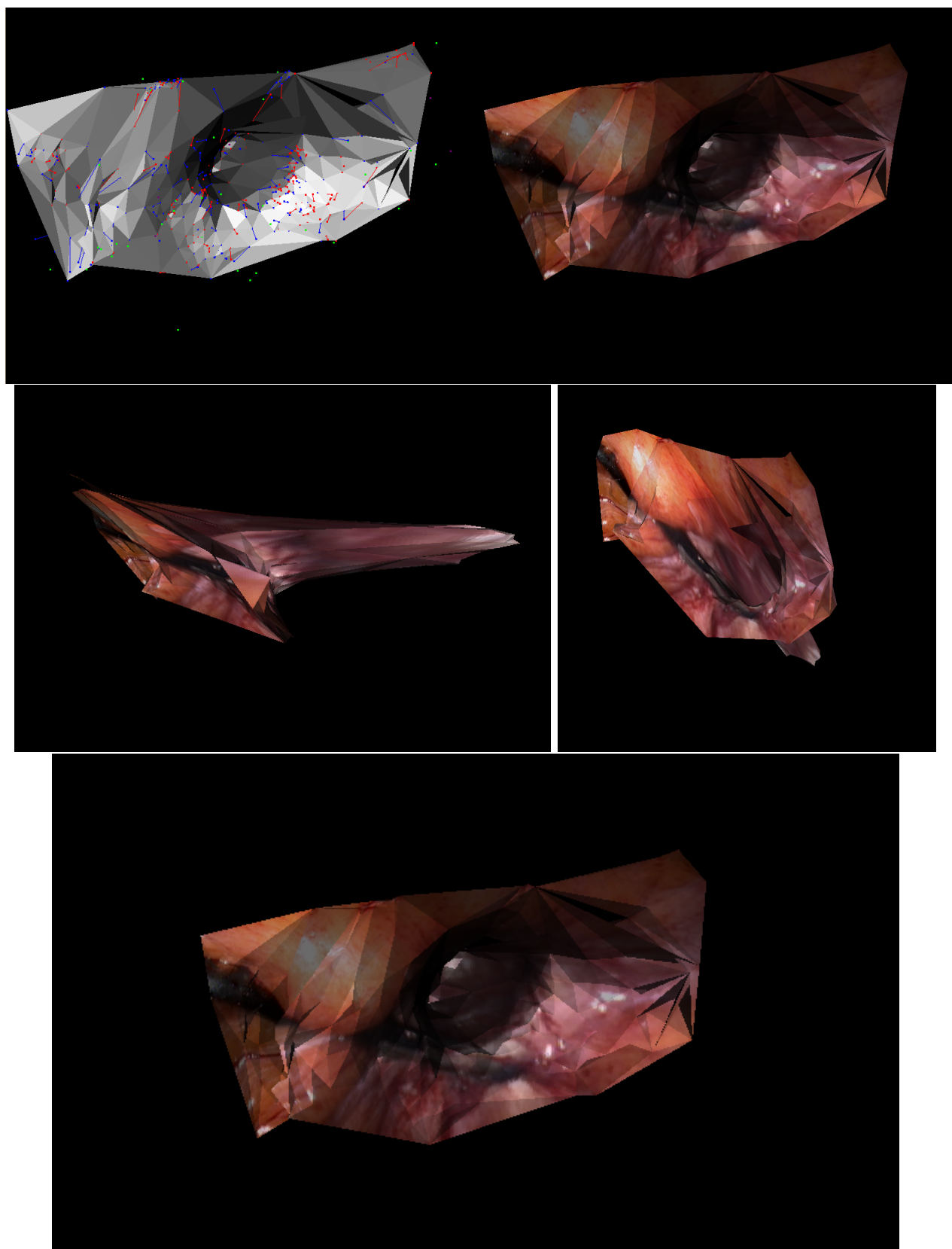


Figura A.15: Vistas de la reconstrucción final de la secuencia (e) de la fig. 6.1

A.6. Secuencia 2011-10-18

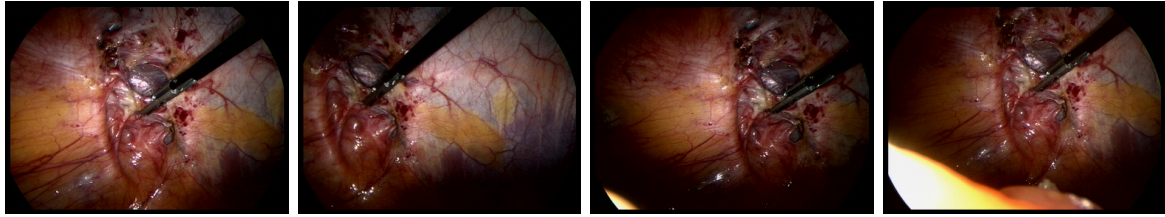


Figura A.16: Frames característicos seleccionados de la secuencia (f) de la fig. 6.1.

	% Error > 2px	% Mediana	Mapa Final(ptos)
PTAM Original	31 %	2 %	1503
PTAM $\sigma = 2$	19 %	1 %	1354
Inlier > 80 %	14 %	2 %	1166
Visto > 30 %	16 %	1 %	936

Tabla A.11: Porcentaje de puntos eliminados aplicando las etapas de eliminación según error de reproyección y error de mínima mediana. Secuencia (f) fig. 6.1.

	Algoritmo Eliminación Posteriori		Algoritmo Triangulación
	% Error > 2px	% Error Mínima Mediana	Modelo 3D
Tiempo cómputo (seg)	0.98	0.13	0.32

Tabla A.12: Tiempo de cómputo de las diferentes etapas de eliminación de espurios del algoritmo a posteriori y de obtención del modelo 3D de la escena. Secuencia (f) fig. 6.1.

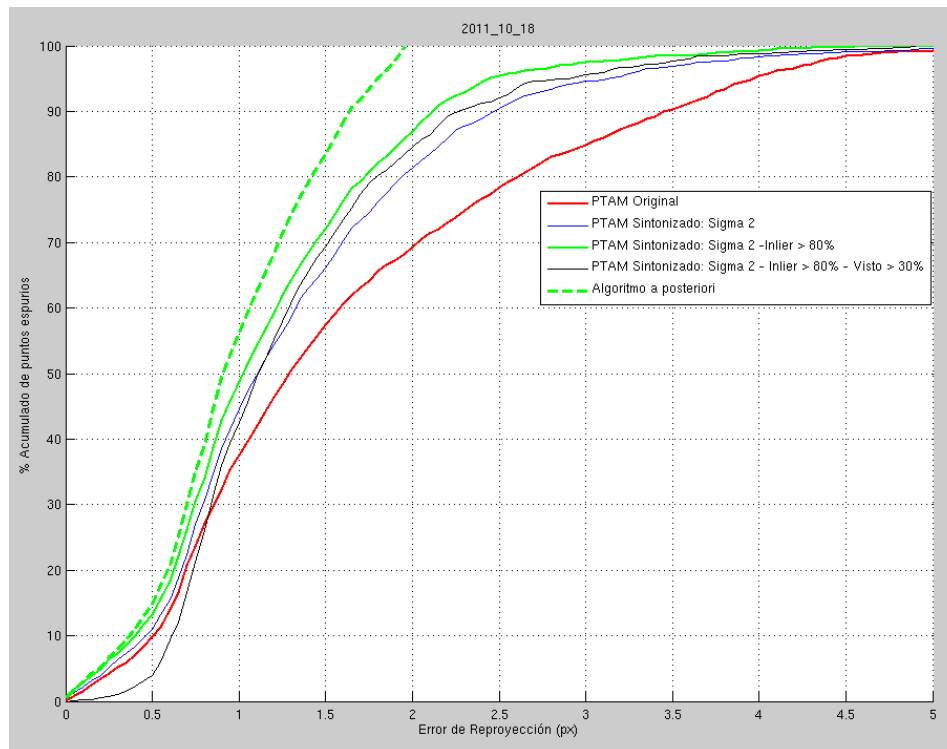


Figura A.17: Porcentaje acumulado del Error de Reproyección para la secuencia de (f) de la fig. 6.1.

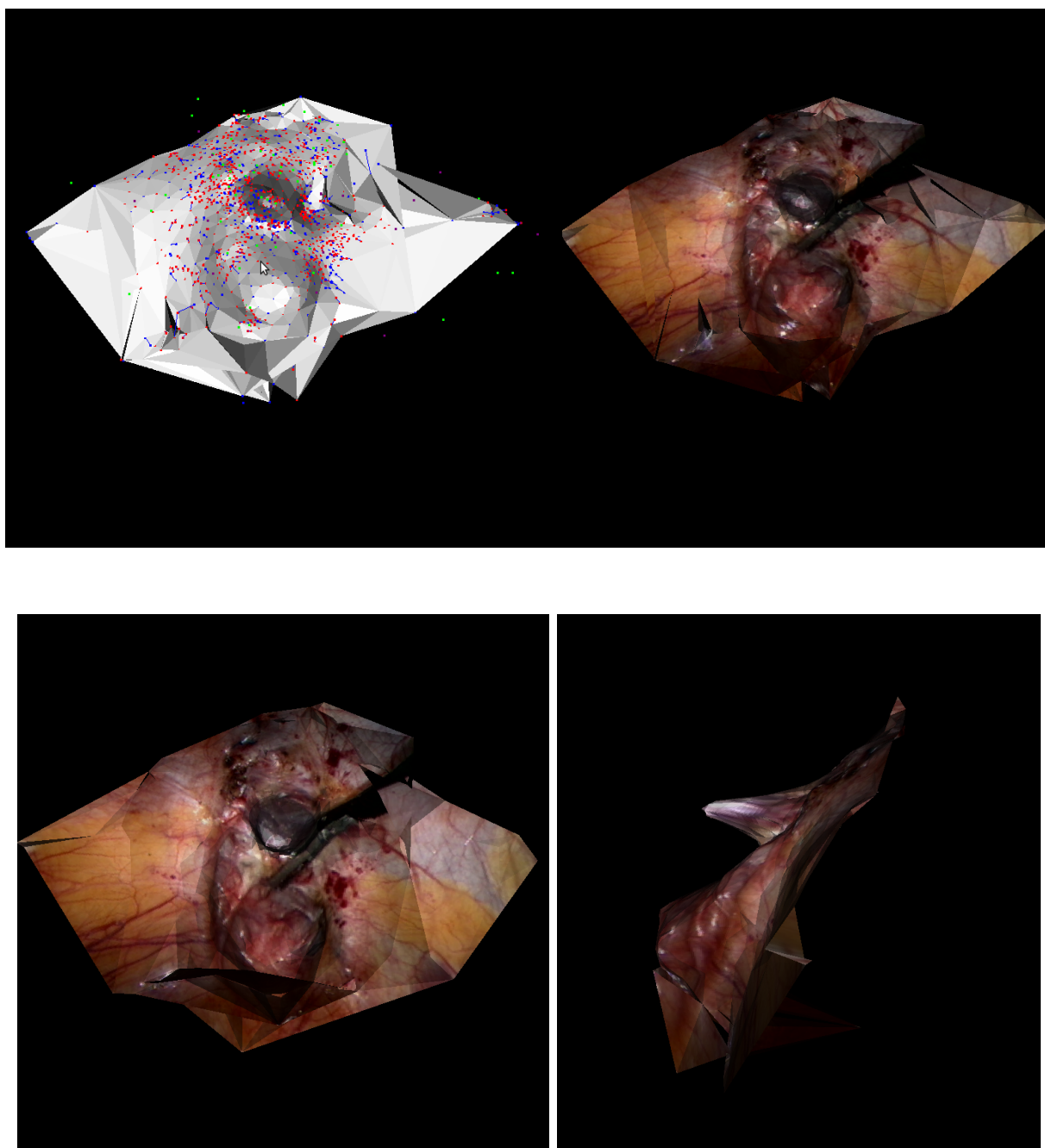


Figura A.18: Vistas de la reconstrucción final de la secuencia (f) de la fig. 6.1

A.7. Secuencia 2011-11-29

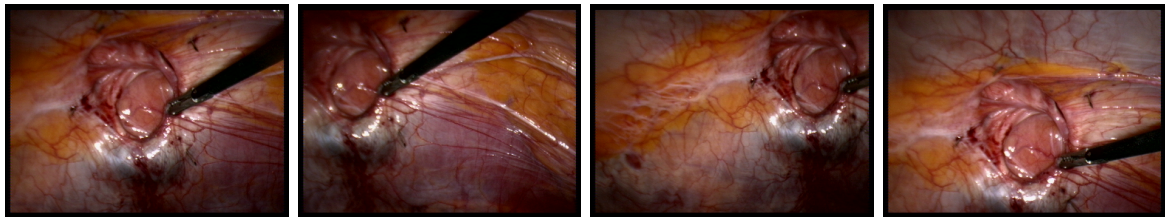


Figura A.19: Frames característicos seleccionados de la secuencia (g) de la fig. 6.1

	% Error > 2px	% Mediana	Mapa Final(ptos)
PTAM Original	19 %	1 %	428
PTAM $\sigma = 2$	18 %	2 %	472
Inlier > 80 %	7 %	1 %	291
Visto > 30 %	8 %	1 %	262

Tabla A.13: Porcentaje de puntos eliminados aplicando las etapas de eliminación según error de reproyección y error de mínima mediana. Secuencia (g) fig. 6.1.

	Algoritmo Eliminación Posteriori		Algoritmo Triangulación
	% Error > 2px	% Error Mínima Mediana	Modelo 3D
Tiempo cómputo (seg)	0.08	0.01	0.08

Tabla A.14: Tiempo de cómputo de las diferentes etapas de eliminación de espurios del algoritmo a posteriori y de obtención del modelo 3D de la escena. Secuencia (g) fig. 6.1.

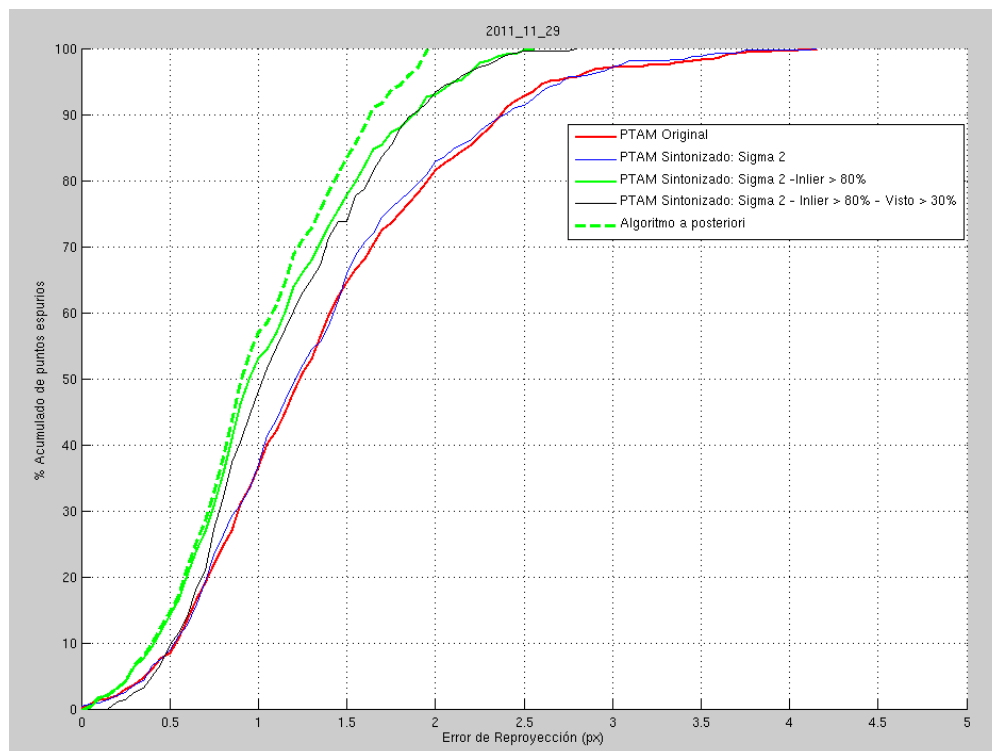


Figura A.20: Porcentaje acumulado del Error de Reproyección para la secuencia de (g) de la fig. 6.1.

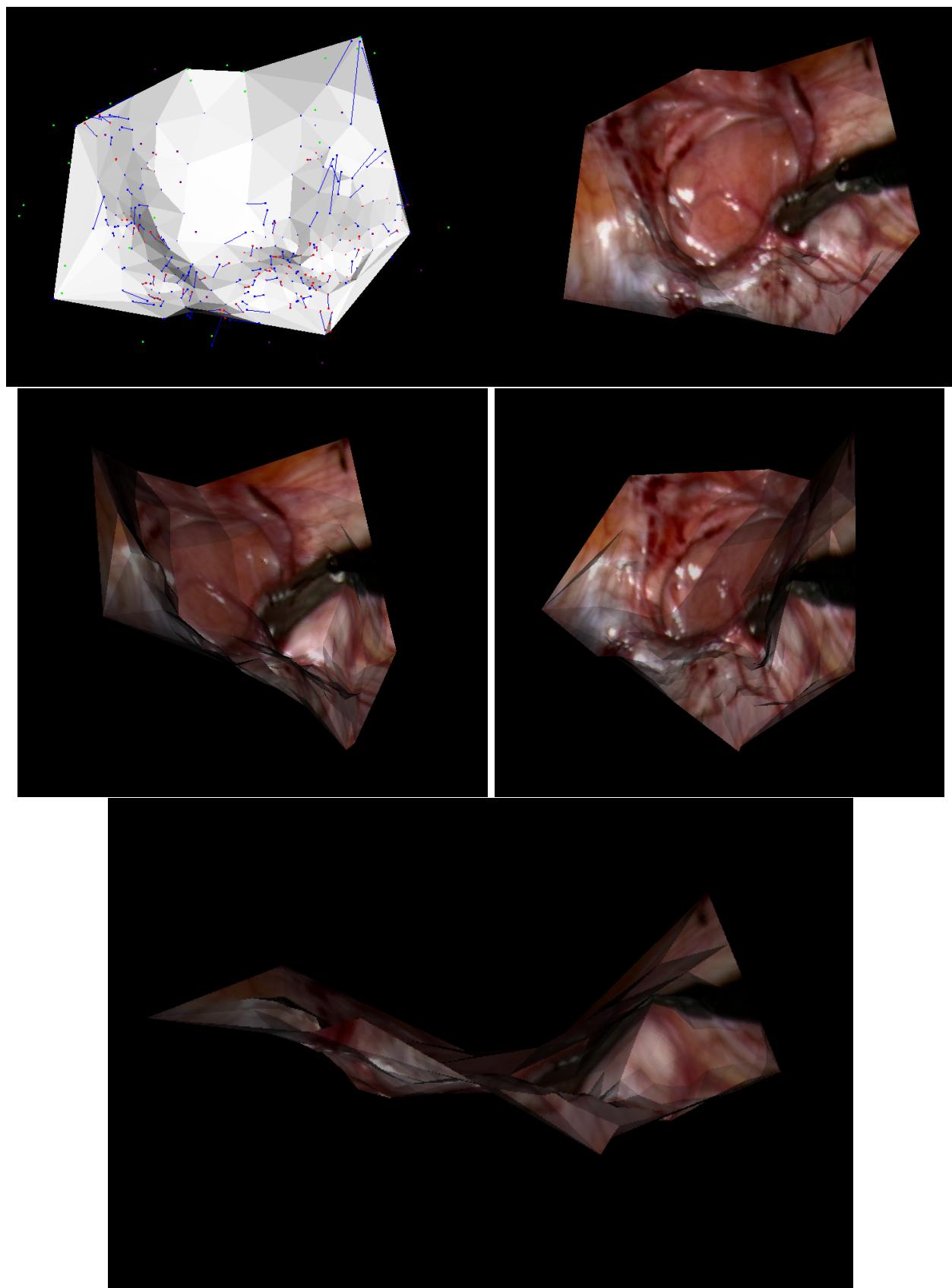


Figura A.21: Vistas de la reconstrucción final de la secuencia (g) de la fig. 6.1

A.8. Secuencia 2012-01-23

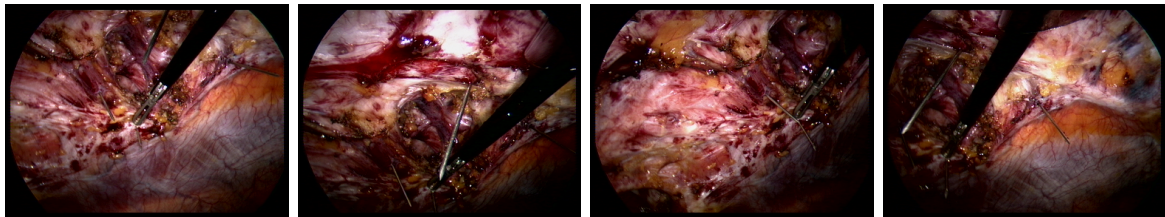


Figura A.22: Frames característicos seleccionados de la secuencia (h) de la fig. 6.1.

	% Error > 2px	% Mediana	Mapa Final(ptos)
PTAM Original	29 %	3 %	3919
PTAM $\sigma = 2$	14 %	3 %	4523
Inlier > 80 %	7 %	3 %	3897
Visto > 30 %	7 %	2 %	3348

Tabla A.15: Porcentaje de puntos eliminados aplicando las etapas de eliminación según error de reproyección y error de mínima mediana. Secuencia (h) fig. 6.1.

	Algoritmo Eliminación Posteriori		Algoritmo Triangulación
	% Error > 2px	% Error Mínima Mediana	Modelo 3D
Tiempo cómputo (seg)	2.2	0.51	2.51

Tabla A.16: Tiempo de cómputo de las diferentes etapas de eliminación de espurios del algoritmo a posteriori y de obtención del modelo 3D de la escena. Secuencia (h) fig. 6.1.

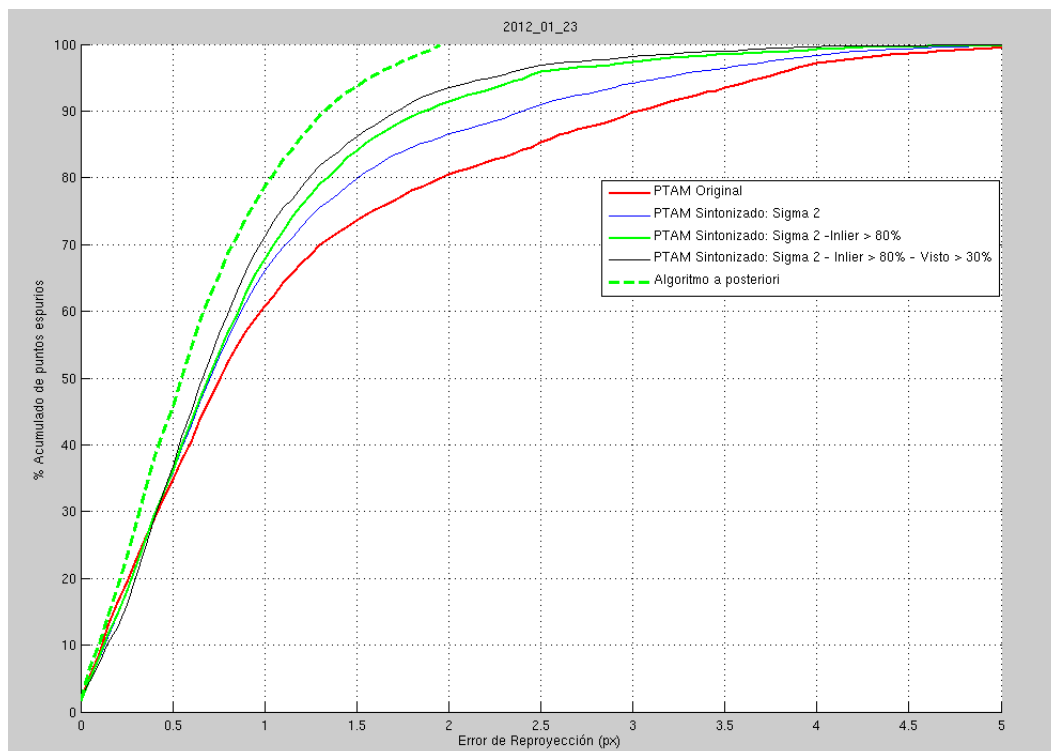


Figura A.23: Porcentaje acumulado del Error de Reproyección para la secuencia de (h) de la fig. 6.1.

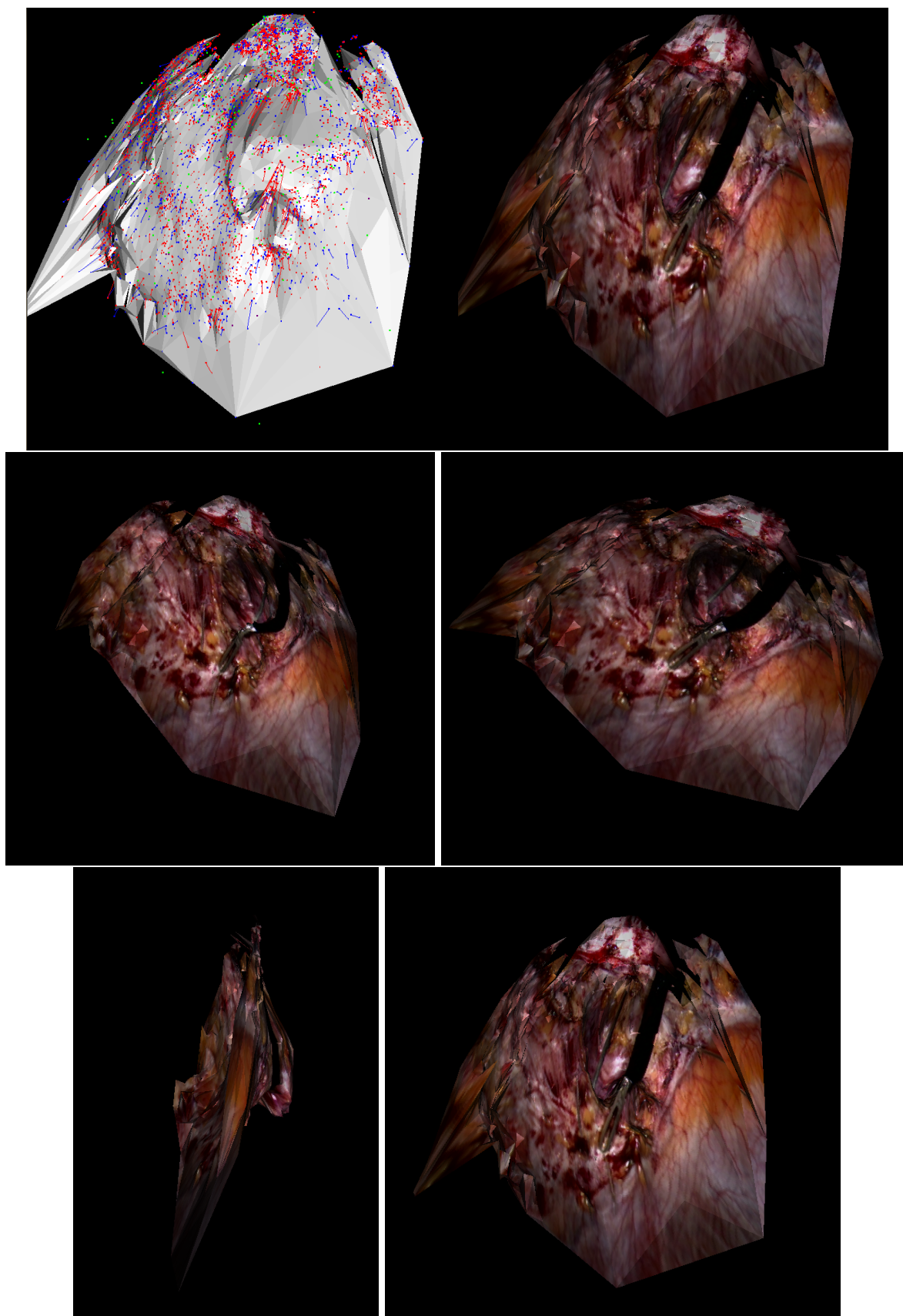


Figura A.24: Vistas de la reconstrucción final de la secuencia (h) de la fig. 6.1

A.9. Secuencia 2012-02-06

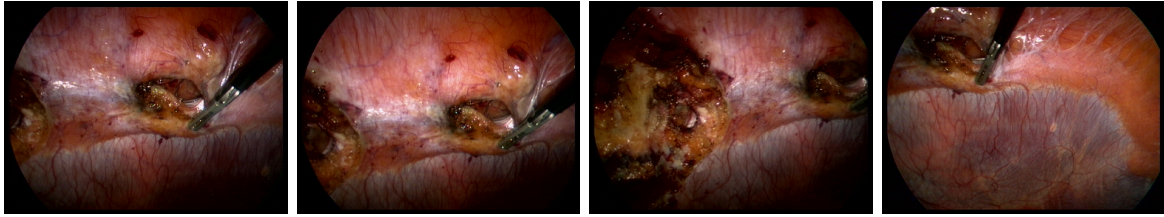


Figura A.25: Frames característicos seleccionados de la secuencia (i) de la fig. 6.1.

	% Error > 2px	% Mediana	Mapa Final(ptos)
PTAM Original	28 %	6 %	1433
PTAM $\sigma = 2$	14 %	6 %	1442
Inlier > 80 %	7 %	7 %	1431
Visto > 30 %	8 %	6 %	1264

Tabla A.17: Porcentaje de puntos eliminados aplicando las etapas de eliminación según error de reproyección y error de mínima mediana. Secuencia (i) fig. 6.1.

	Algoritmo Eliminación Posteriori		Algoritmo Triangulación
	% Error > 2px	% Error Mínima Mediana	Modelo 3D
Tiempo cómputo (seg)	0.56	0.15	0.60

Tabla A.18: Tiempo de cómputo de las diferentes etapas de eliminación de espurios del algoritmo a posteriori y de obtención del modelo 3D de la escena. Secuencia (i) fig. 6.1.

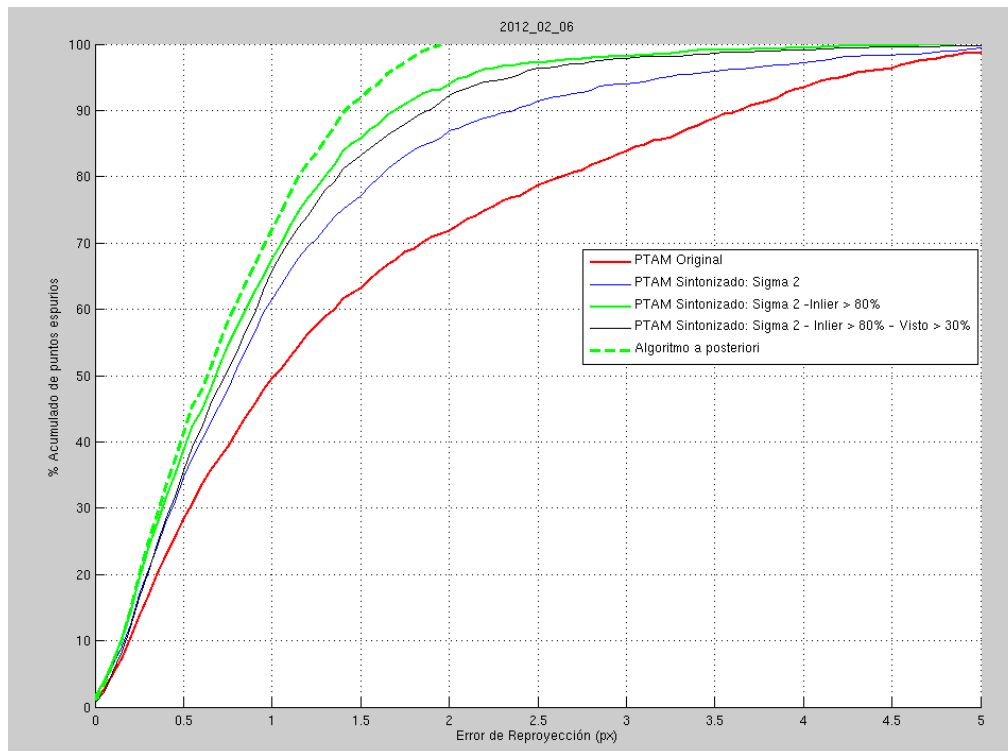


Figura A.26: Porcentaje acumulado del Error de Reproyección para la secuencia de (i) de la fig. 6.1.

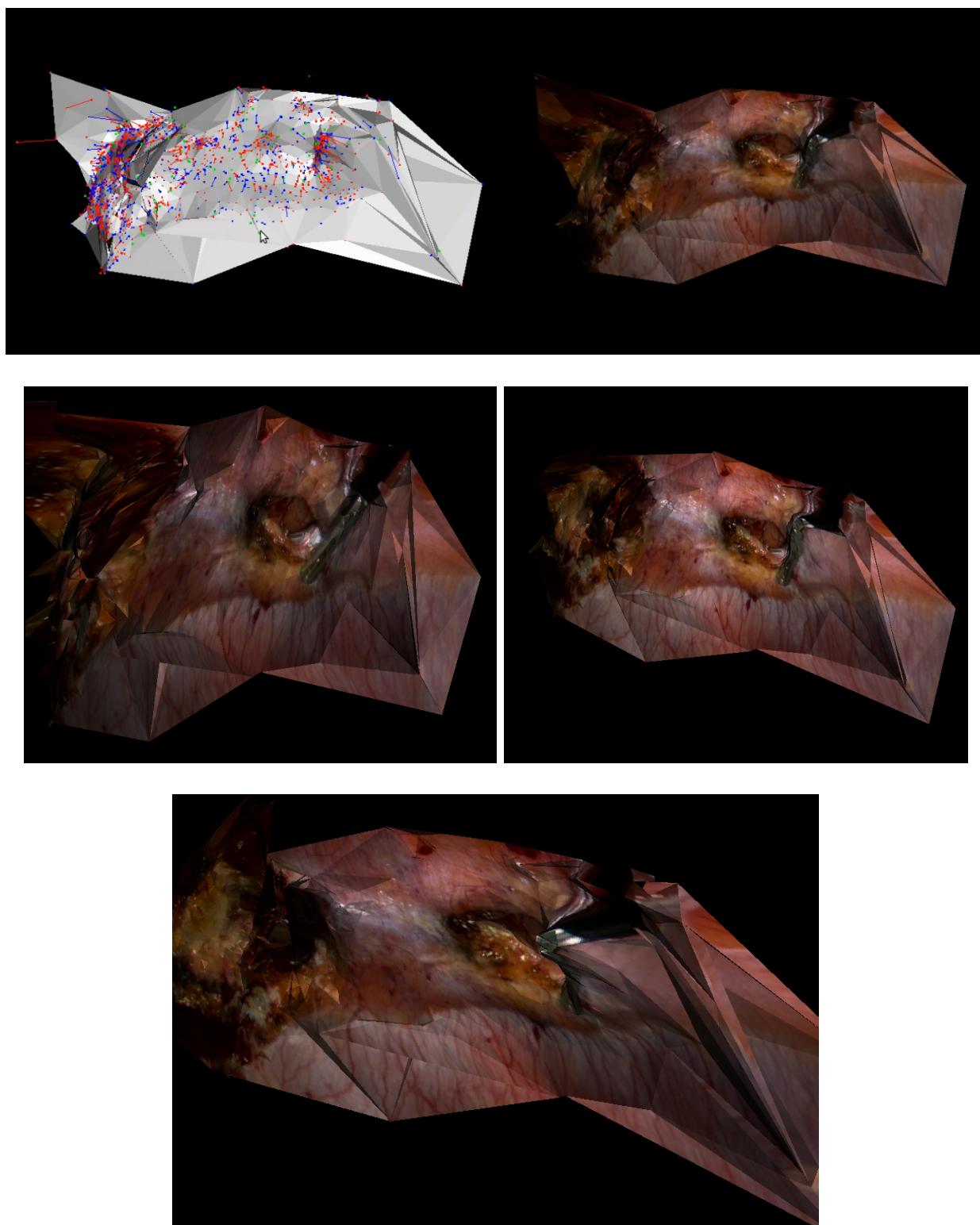


Figura A.27: Vistas de la reconstrucción final de la secuencia (i) de la fig. 6.1

A.10. Secuencia 2012-03-06

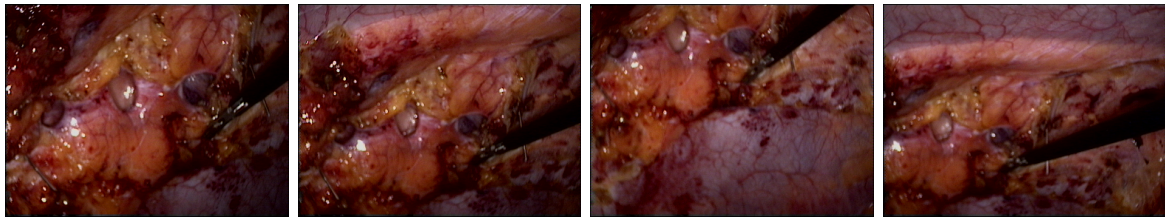


Figura A.28: Frames característicos seleccionados de la secuencia (j) de la fig. 6.1.

	% Error > 2px	% Mediana	Mapa Final(ptos)
PTAM Original	29 %	8 %	1069
PTAM $\sigma = 2$	15 %	6 %	1054
Inlier > 80 %	5 %	9 %	803
Visto > 30 %	5 %	9 %	704

Tabla A.19: Porcentaje de puntos eliminados aplicando las etapas de eliminación según error de reproyección y error de mínima mediana. Secuencia (j) fig. 6.1.

	Algoritmo Eliminación Posteriori		Algoritmo Triangulación
	% Error > 2px	% Error Mínima Mediana	Modelo 3D
Tiempo cómputo (seg)	0.23	0.09	0.32

Tabla A.20: Tiempo de cómputo de las diferentes etapas de eliminación de espurios del algoritmo a posteriori y de obtención del modelo 3D de la escena. Secuencia (j) fig. 6.1.

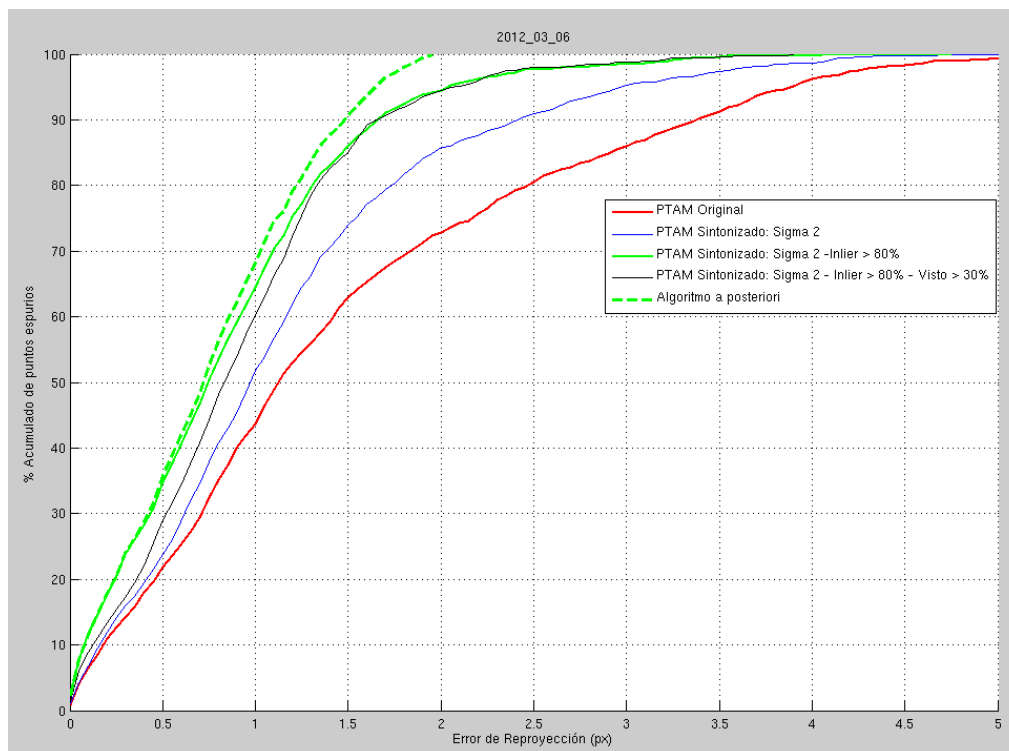


Figura A.29: Porcentaje acumulado del Error de Reproyección para la secuencia de (j) de la fig. 6.1.

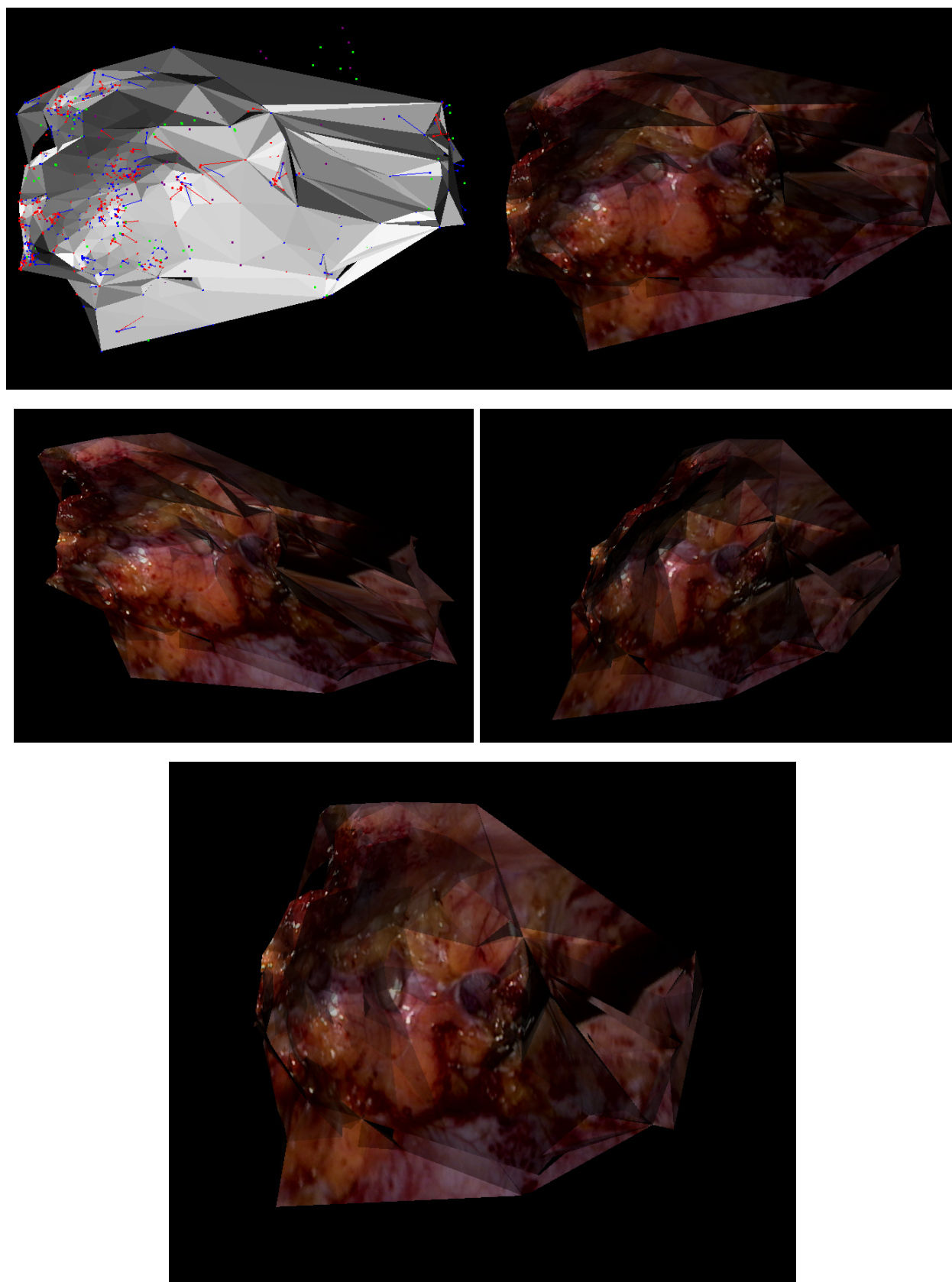


Figura A.30: Vistas de la reconstrucción final de la secuencia (j) de la fig. 6.1

A.11. Secuencia 2012-05-21(1)

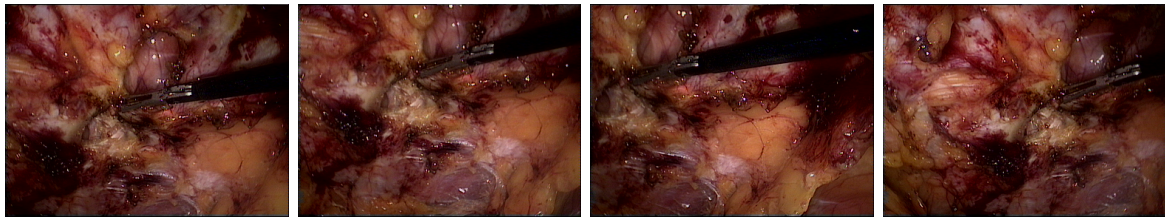


Figura A.31: Frames característicos seleccionados de la secuencia (k) de la fig. 6.1.

	% Error > 2px	% Mediana	Mapa Final(ptos)
PTAM Original	14 %	7 %	1293
PTAM $\sigma = 2$	13 %	5 %	1055
Inlier > 80 %	3 %	5 %	784
Visto > 30 %	4 %	4 %	689

Tabla A.21: Porcentaje de puntos eliminados aplicando las etapas de eliminación según error de reproyección y error de mínima mediana. Secuencia (k) fig. 6.1.

	Algoritmo Eliminación Posteriori		Algoritmo Triangulación
	% Error > 2px	% Error Mínima Mediana	Modelo 3D
Tiempo cómputo (seg)	0.49	0.06	0.20

Tabla A.22: Tiempo de cómputo de las diferentes etapas de eliminación de espurios del algoritmo a posteriori y de obtención del modelo 3D de la escena. Secuencia (k) fig. 6.1.

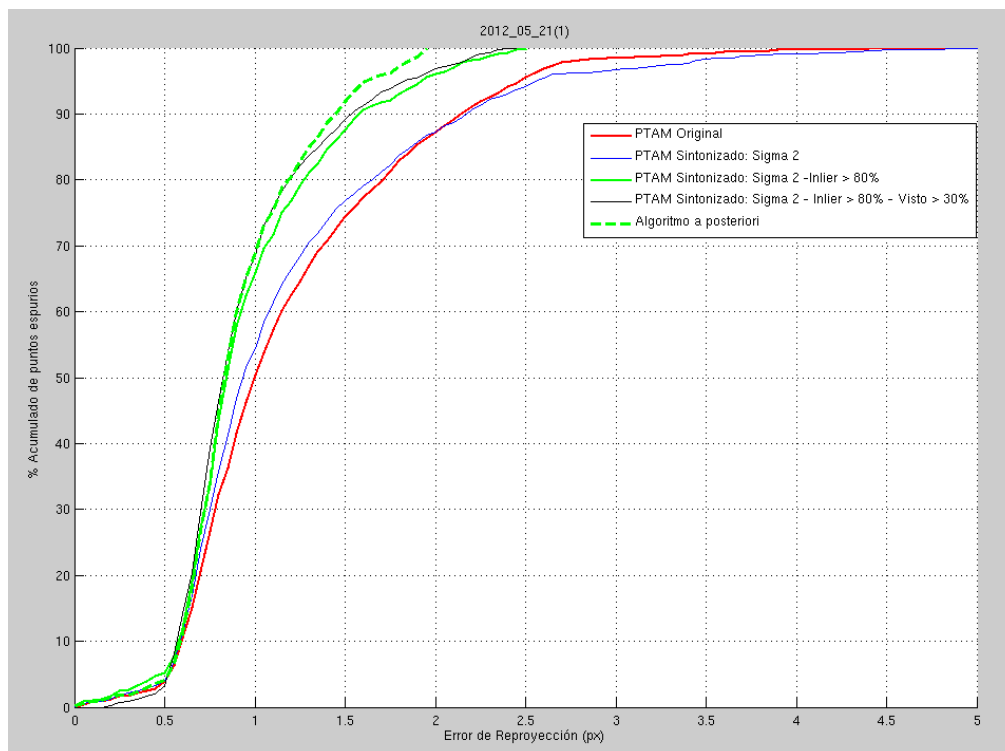


Figura A.32: Porcentaje acumulado del Error de Reproyección para la secuencia de (k) de la fig. 6.1.

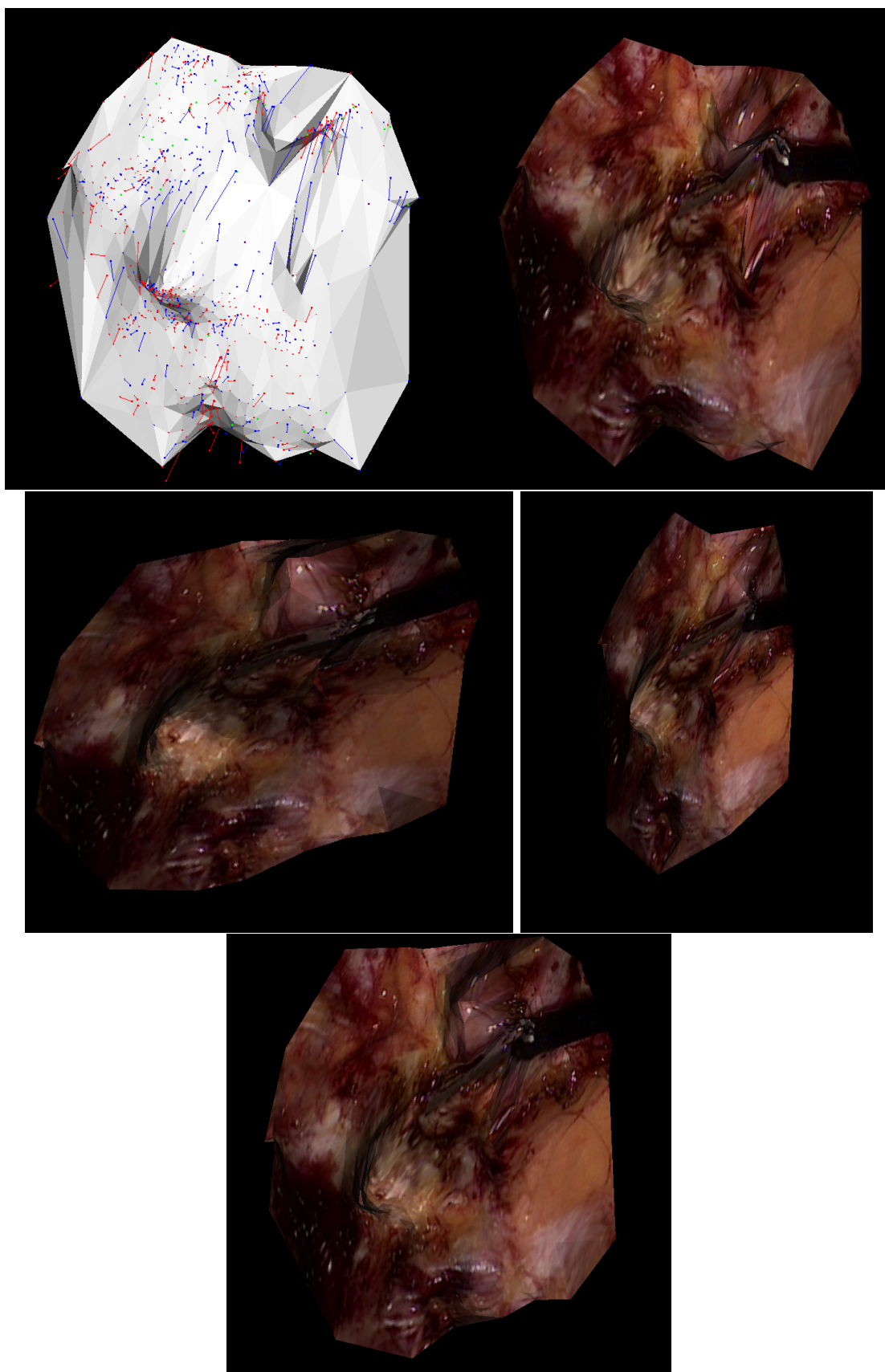


Figura A.33: Vistas de la reconstrucción final de la secuencia (k) de la fig. 6.1

A.12. Secuencia 2012-05-21(2)

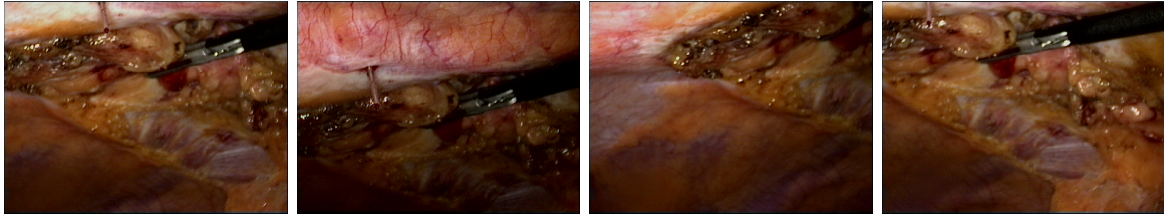


Figura A.34: Frames característicos seleccionados de la secuencia (l) de la fig. 6.1.

	% Error > 2px	% Mediana	Mapa Final(ptos)
PTAM Original	11 %	10 %	581
PTAM $\sigma = 2$	7 %	9.47955 %	452
Inlier > 80 %	0 %	9 %	523
Visto > 30 %	4 %	7 %	198

Tabla A.23: Porcentaje de puntos eliminados aplicando las etapas de eliminación según error de reproyección y error de mínima mediana. Secuencia (l) fig. 6.1.

	Algoritmo Eliminación Posteriori		Algoritmo Triangulación
	% Error > 2px	% Error Mínima Mediana	Modelo 3D
Tiempo cómputo (seg)	0.5	0.02	0.21

Tabla A.24: Tiempo de cómputo de las diferentes etapas de eliminación de espurios del algoritmo a posteriori y de obtención del modelo 3D de la escena. Secuencia (l) fig. 6.1.

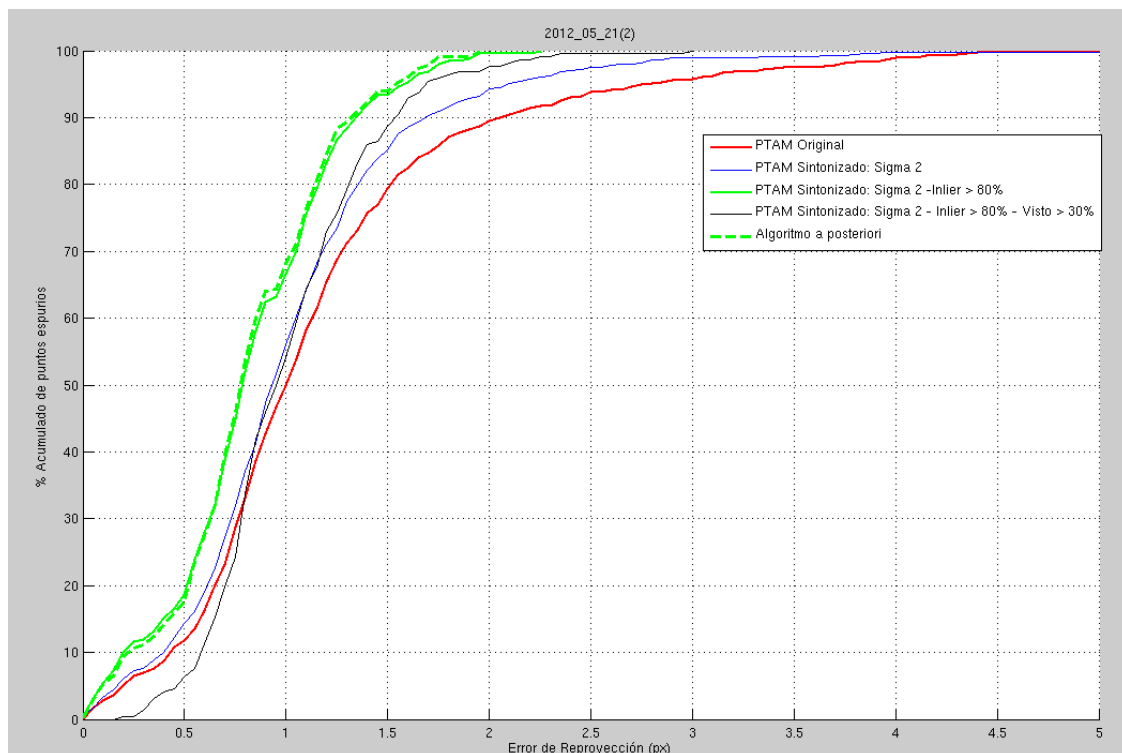


Figura A.35: Porcentaje acumulado del Error de Reproyección para la secuencia de (l) de la fig. 6.1.

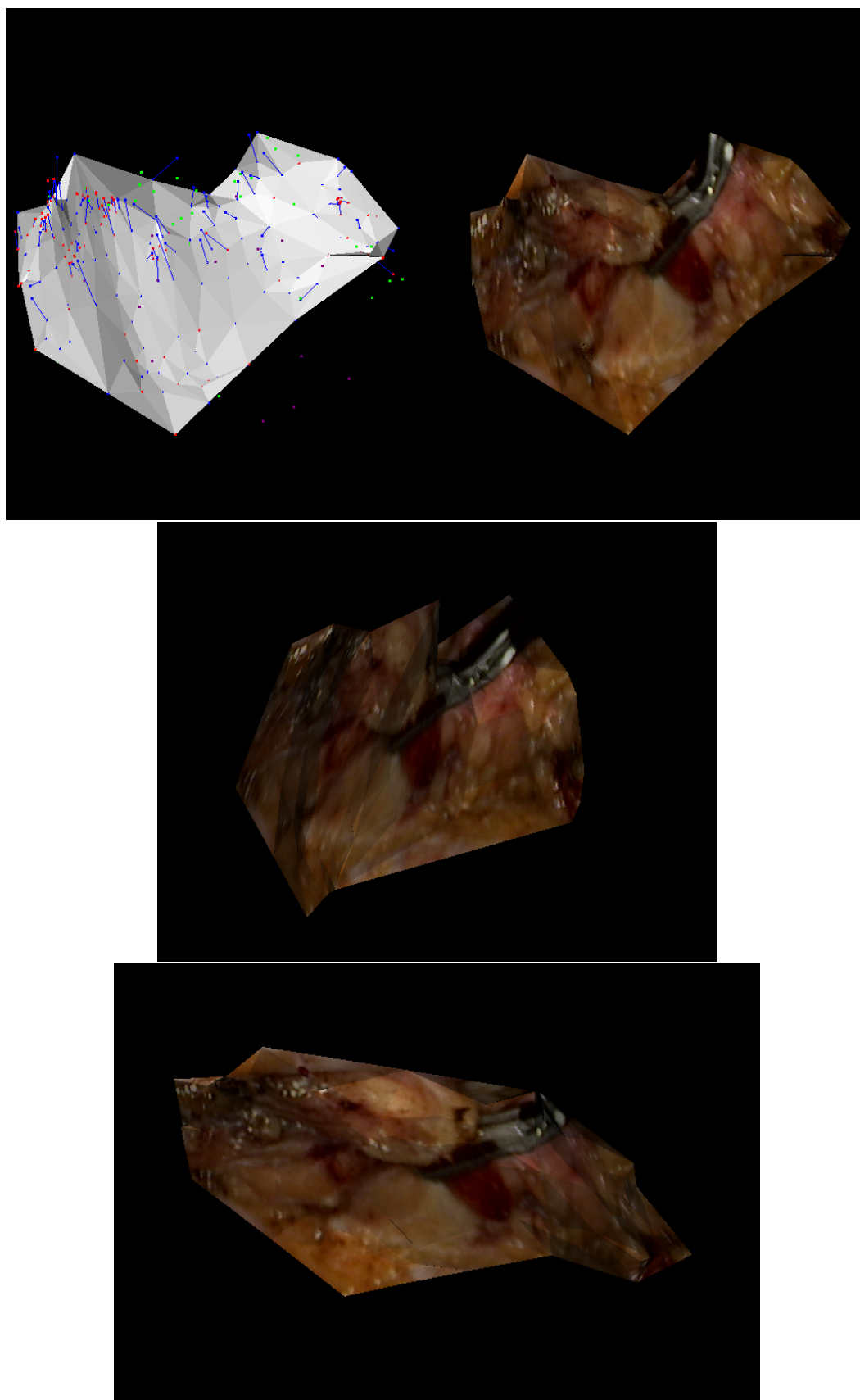


Figura A.36: Vistas de la reconstrucción final de la secuencia (l) de la fig. 6.1

A.13. Secuencia 2012-06-04

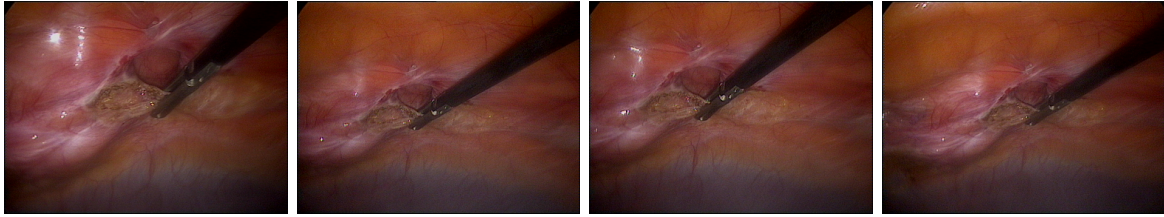


Figura A.37: Frames característicos seleccionados de la secuencia (m) de la fig. 6.1.

	% Error > 2px	% Mediana	Mapa Final(ptos)
PTAM Original	39 %	15 %	43
PTAM $\sigma = 2$	7 %	24 %	50
Inlier > 80 %	6 %	65 %	16
Visto > 30 %	10 %	45 %	14

Tabla A.25: Porcentaje de puntos eliminados aplicando las etapas de eliminación según error de reproyección y error de mínima mediana. Secuencia (m) fig. 6.1.

	Algoritmo Eliminación Posteriori		Algoritmo Triangulación
	% Error > 2px	% Error Mínima Mediana	Modelo 3D
Tiempo cómputo (seg)	0	0	0

Tabla A.26: Tiempo de cómputo de las diferentes etapas de eliminación de espurios del algoritmo a posteriori y de obtención del modelo 3D de la escena. Secuencia (m) fig. 6.1.

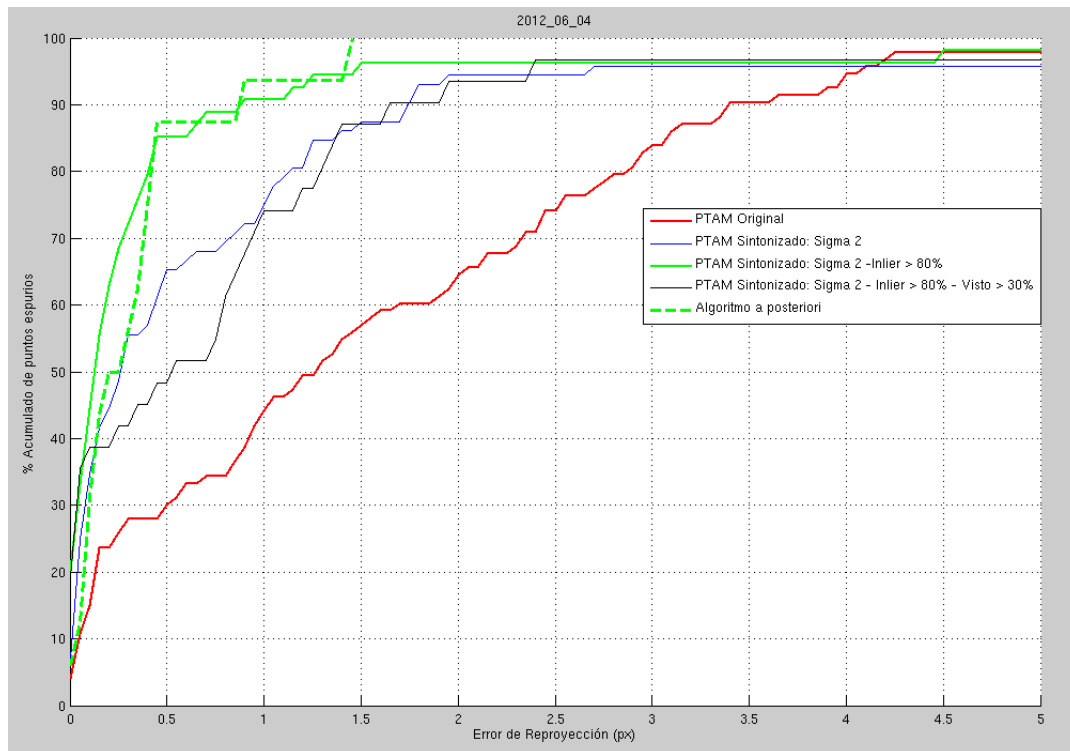


Figura A.38: Porcentaje acumulado del Error de Reproyección para la secuencia de (m) de la fig. 6.1.

A.14. Secuencia 2012-07-02(1)

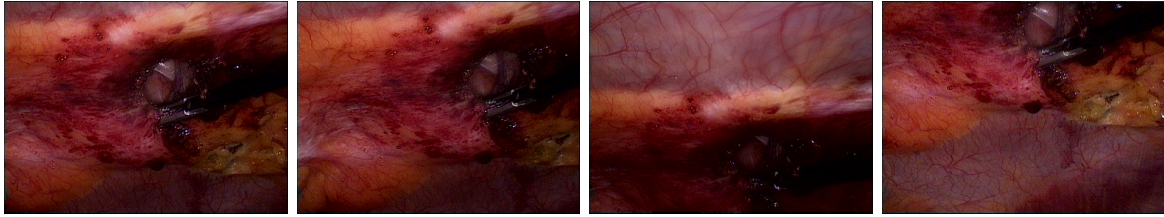


Figura A.39: Frames característicos seleccionados de la secuencia (n) de la fig. 6.1.

	% Error > 2px	% Mediana	Mapa Final(ptos)
PTAM Original	29 %	4 %	1199
PTAM $\sigma = 2$	28 %	4 %	1099
Inlier > 80 %	17 %	5 %	907
Visto > 30 %	13 %	6 %	992

Tabla A.27: Porcentaje de puntos eliminados aplicando las etapas de eliminación según error de reproyección y error de mínima mediana. Secuencia (n) fig. 6.1.

	Algoritmo Eliminación Posteriori		Algoritmo Triangulación
	% Error > 2px	% Error Mínima Mediana	Modelo 3D
Tiempo cómputo (seg)	0.41	0.10	0.26

Tabla A.28: Tiempo de cómputo de las diferentes etapas de eliminación de espurios del algoritmo a posteriori y de obtención del modelo 3D de la escena. Secuencia (n) fig. 6.1.

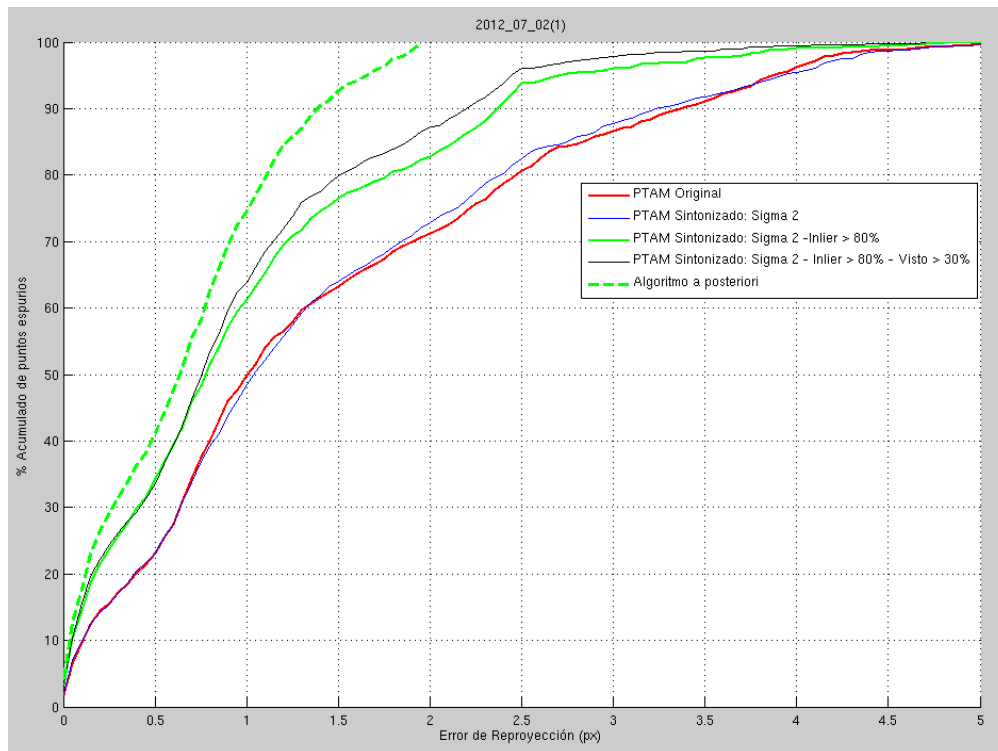


Figura A.40: Porcentaje acumulado del Error de Reproyección para la secuencia de (n) de la fig. 6.1.

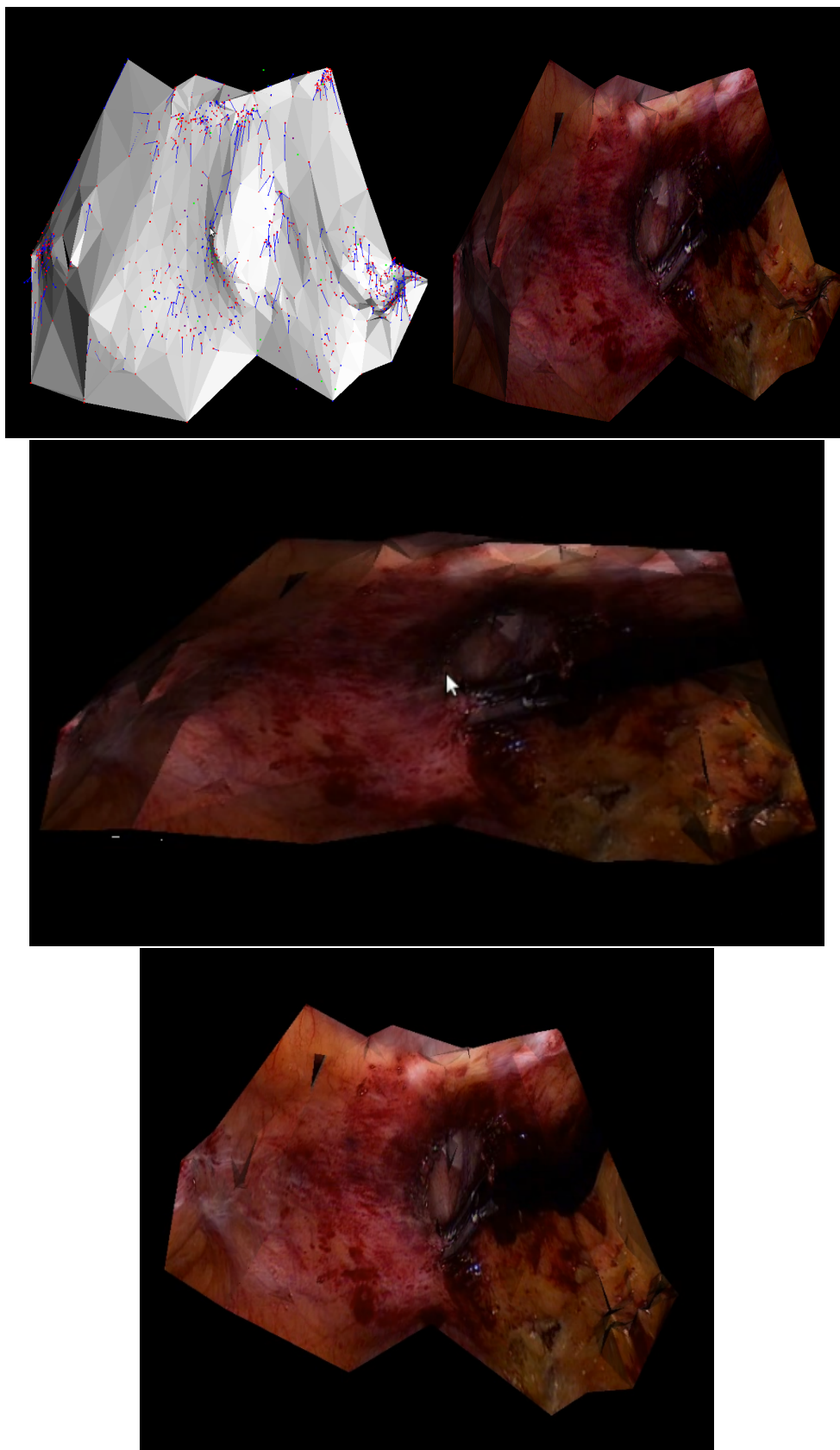
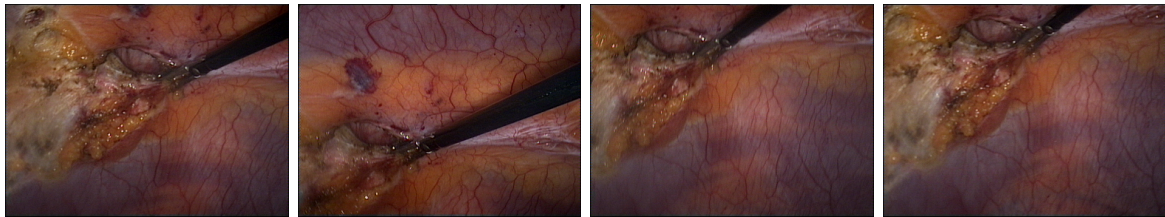


Figura A.41: Vistas de la reconstrucción final de la secuencia (n) de la fig. 6.1

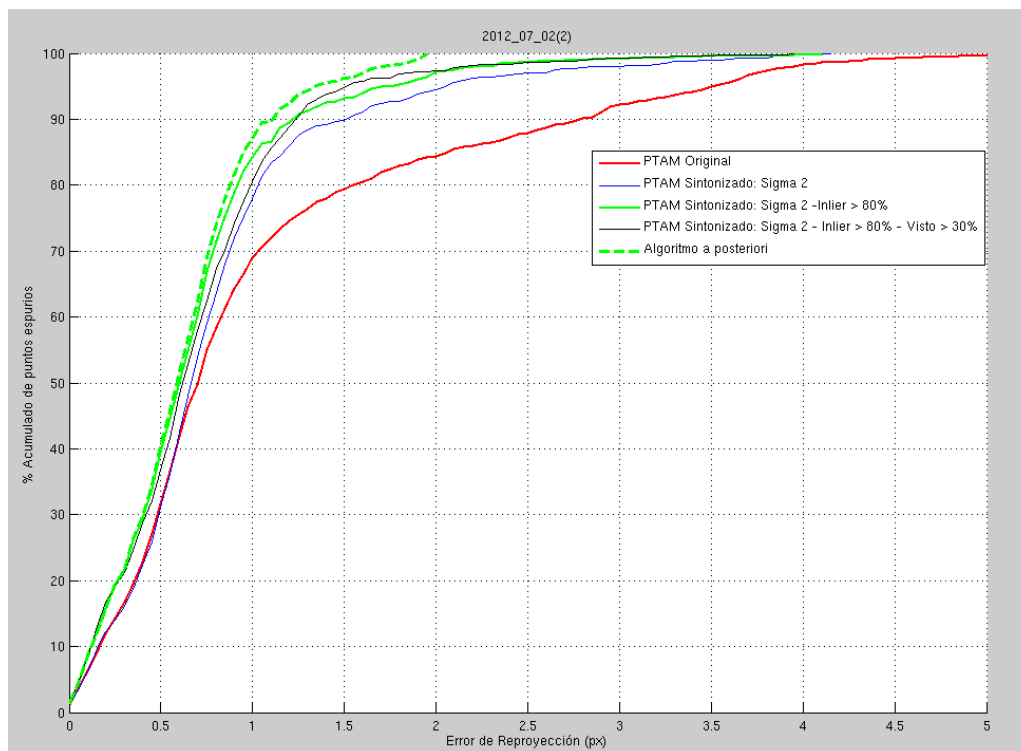
A.15. Secuencia 2012-07-02(2)

Figura A.42: Frames característicos seleccionados de la secuencia (\tilde{n}) de la fig. 6.1.

	% Error > 2px	% Mediana	Mapa Final(ptos)
PTAM Original	16 %	3 %	861
PTAM $\sigma = 2$	6 %	2 %	846
Inlier > 80 %	3 %	3 %	606
Visto > 30 %	3 %	2 %	551

Tabla A.29: Porcentaje de puntos eliminados aplicando las etapas de eliminación según error de reproyección y error de mínima mediana. Secuencia (\tilde{n}) fig. 6.1.

	Algoritmo Eliminación Posteriori		Algoritmo Triangulación
	% Error > 2px	% Error Mínima Mediana	Modelo 3D
Tiempo cómputo (seg)	0.11	0.04	0.17

Tabla A.30: Tiempo de cómputo de las diferentes etapas de eliminación de espurios del algoritmo a posteriori y de obtención del modelo 3D de la escena. Secuencia (\tilde{n}) fig. 6.1.Figura A.43: Porcentaje acumulado del Error de Reproyección para la secuencia de (\tilde{n}) de la fig. 6.1.

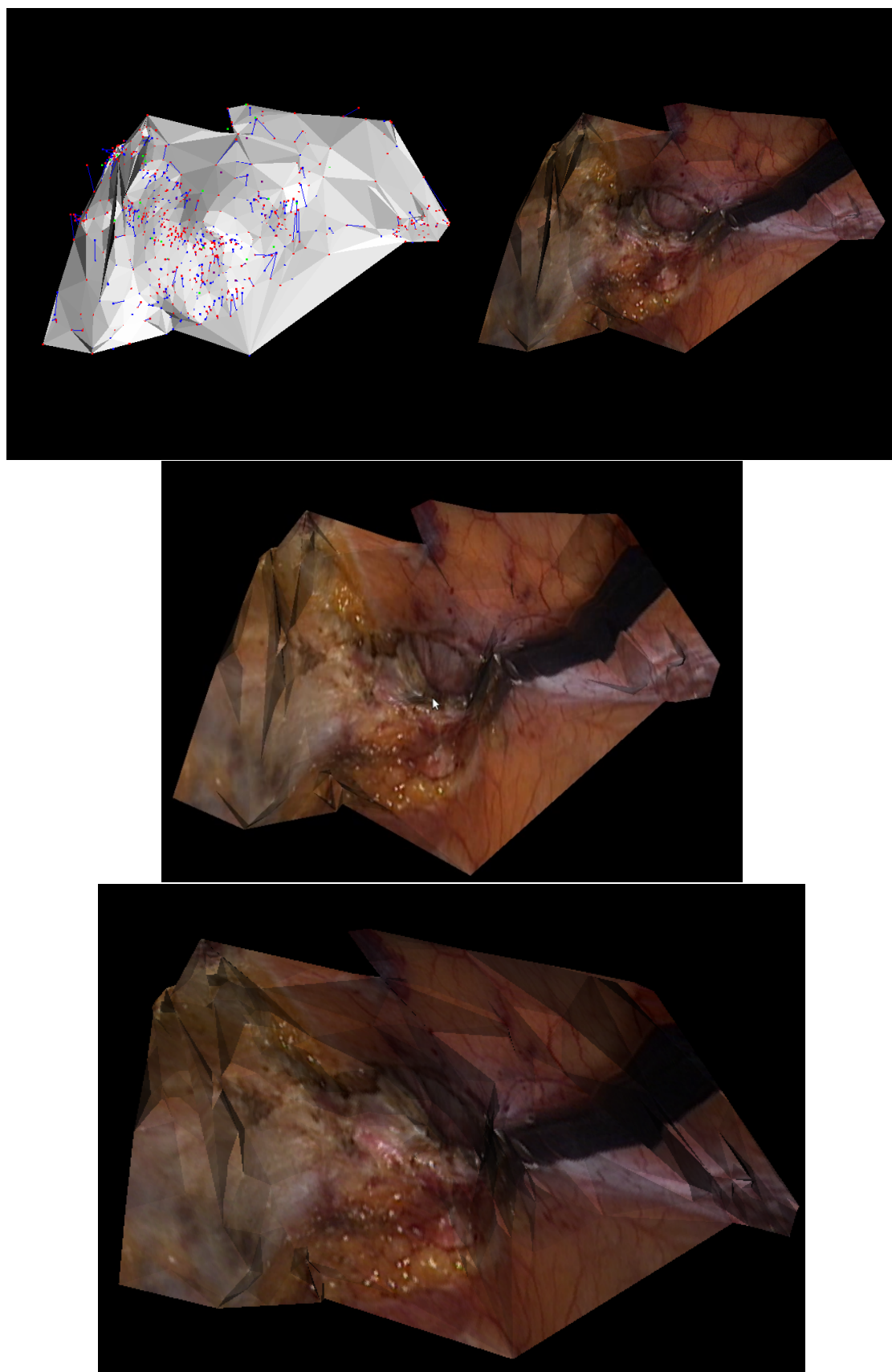


Figura A.44: Vistas de la reconstrucción final de la secuencia (\tilde{n}) de la fig. 6.1

Índice de figuras

1.1. Modelo fotorealista obtenido aplicando el software 123D Catch.	2
2.1. Selección de frames de una operación real de reparación de hernia ventral.	7
2.2. Reconstrucción 3D de la cavidad abdominal a partir del mapa estimado por PTAM.	8
3.1. (a) Búsqueda de los distintos KFs que ven el punto. (b) Cálculo del ángulo entre los rayos de los KFs.	11
3.2. Eliminación de puntos outliers en función de la profundidad media del KF	13
4.1. Frames consecutivos en los que se puede apreciar cómo no todo el mapa estimado es proyectado sobre ellos.	17
5.1. Reconstrucción 3D de la cavidad abdominal a partir de una secuencia real de cirugía	19
5.2. Método de búsqueda empleado para la triangulación Greedy. (a) silueta del contorno que debería dar forma a la cavidad abdominal. (b) Búsqueda de los k-vecinos que se encuentran dentro del radio de búsqueda. (c) Selección de los puntos que cumplen la condición de aceptación y que serán triangulados. En verde, la línea que debería seguir el mallado correcto.	20
5.3. Reconstrucción fotorealista obtenida para la secuencia (f) de la fig. 6.1. La imagen de la izq. muestra el resultado tras aplicar una triangulación DELAUNAY sobre el plano XY. La de la derecha es el resultado final obtenido tras la aplicación del algoritmo desarrollado de triangulación.	21
5.4. Triangulación Delaunay del mapa estimado sobre el primer KF de la secuencia 6.1(b).	22
5.5. Método de triangulación de puntos interiores a la Triangulación Delaunay inicial. Si un punto del mapa se encuentra en el interior de un triángulo ya creado, este se reparticiona eliminando el triángulo original y generando 3 nuevos triángulos al conjunto.	23
6.1. Secuencias reales de operaciones con endoscopio proporcionadas por el Hospital clínico Lozano Blesa para verificar los resultados.	25
6.2. Porcentaje acumulado del Error de Reproyección para la secuencia de (b) de la fig. 6.1 según el criterio de eliminación de error de reproyección a posteriori.	26
6.3. Mapa final tras aplicar la etapa de eliminación según ángulo de paralaje. En rojo, los puntos eliminados en la etapa de ángulo de paralaje.	27
6.4. Mapa final tras aplicar la etapa de eliminación según ángulo de paralaje y error de mínima mediana. En rojo, puntos eliminados en la etapa de ángulo de paralaje. En azul, puntos eliminados en la etapa de error de mínima mediana.	28

6.5. Mapa final tras aplicar la etapa de eliminación según ángulo de paralaje. En rojo, los puntos eliminados en la etapa de ángulo de paralaje.	28
6.6. Porcentaje acumulado del Error de Reproyección para la secuencia de (b) de la fig. 6.1. . . .	30
6.7. Mallado obtenido tras aplicar la condición de reobservabilidad. En blanco aparecen los puntos considerados inliers tras la aplicación del algoritmo a posteriori y que generan el mallado 3D. En rojo los puntos eliminados durante la ejecución según la condición de reobservabilidad.	31
6.8. Porcentajes de puntos espurios con error de reproyección mayor a 2 píxeles para las distintas secuencias.	32
6.9. Reconstrucción 3D de la secuencia (b) de la fig. 6.1	33
6.10. Vistas de la reconstrucción final de la secuencia (b) de la fig. 6.1	34
A.1. Frames característicos seleccionados de la secuencia (a) de la fig. 6.1.	44
A.2. Porcentaje acumulado del Error de Reproyección para la secuencia de (a) de la fig. 6.1. . . .	44
A.3. Vistas de la reconstrucción final de la secuencia (a) de la fig. 6.1	45
A.4. Frames característicos seleccionados de la secuencia (b) de la fig. 6.1.	46
A.5. Porcentaje acumulado del Error de Reproyección para la secuencia de (b) de la fig. 6.1. . . .	46
A.6. Vistas de la reconstrucción final de la secuencia (b) de la fig. 6.1	47
A.7. Frames característicos seleccionados de la secuencia (c) de la fig. 6.1.	48
A.8. Porcentaje acumulado del Error de Reproyección para la secuencia de (c) de la fig. 6.1. . . .	48
A.9. Vistas de la reconstrucción final de la secuencia (c) de la fig. 6.1	49
A.10. Frames característicos seleccionados de la secuencia (d) de la fig. 6.1.	50
A.11. Porcentaje acumulado del Error de Reproyección para la secuencia de (d) de la fig. 6.1. . . .	50
A.12. Vistas de la reconstrucción final de la secuencia (d) de la fig. 6.1	51
A.13. Frames característicos seleccionados de la secuencia (e) de la fig. 6.1.	52
A.14. Porcentaje acumulado del Error de Reproyección para la secuencia de (e) de la fig. 6.1. . . .	52
A.15. Vistas de la reconstrucción final de la secuencia (e) de la fig. 6.1	53
A.16. Frames característicos seleccionados de la secuencia (f) de la fig. 6.1.	54
A.17. Porcentaje acumulado del Error de Reproyección para la secuencia de (f) de la fig. 6.1. . . .	54
A.18. Vistas de la reconstrucción final de la secuencia (f) de la fig. 6.1	55
A.19. Frames característicos seleccionados de la secuencia (g) de la fig. 6.1	56
A.20. Porcentaje acumulado del Error de Reproyección para la secuencia de (g) de la fig. 6.1. . . .	56
A.21. Vistas de la reconstrucción final de la secuencia (g) de la fig. 6.1	57
A.22. Frames característicos seleccionados de la secuencia (h) de la fig. 6.1.	58
A.23. Porcentaje acumulado del Error de Reproyección para la secuencia de (h) de la fig. 6.1. . . .	58
A.24. Vistas de la reconstrucción final de la secuencia (h) de la fig. 6.1	59
A.25. Frames característicos seleccionados de la secuencia (i) de la fig. 6.1.	60
A.26. Porcentaje acumulado del Error de Reproyección para la secuencia de (i) de la fig. 6.1. . . .	60
A.27. Vistas de la reconstrucción final de la secuencia (i) de la fig. 6.1	61

A.28.Frames característicos seleccionados de la secuencia (j) de la fig. 6.1.	62
A.29.Porcentaje acumulado del Error de Reproyección para la secuencia de (j) de la fig. 6.1. . . .	62
A.30.Vistas de la reconstrucción final de la secuencia (j) de la fig. 6.1	63
A.31.Frames característicos seleccionados de la secuencia (k) de la fig. 6.1.	64
A.32.Porcentaje acumulado del Error de Reproyección para la secuencia de (k) de la fig. 6.1. . . .	64
A.33.Vistas de la reconstrucción final de la secuencia (k) de la fig. 6.1	65
A.34.Frames característicos seleccionados de la secuencia (l) de la fig. 6.1.	66
A.35.Porcentaje acumulado del Error de Reproyección para la secuencia de (l) de la fig. 6.1. . . .	66
A.36.Vistas de la reconstrucción final de la secuencia (l) de la fig. 6.1	67
A.37.Frames característicos seleccionados de la secuencia (m) de la fig. 6.1.	68
A.38.Porcentaje acumulado del Error de Reproyección para la secuencia de (m) de la fig. 6.1. . . .	68
A.39.Frames característicos seleccionados de la secuencia (n) de la fig. 6.1.	69
A.40.Porcentaje acumulado del Error de Reproyección para la secuencia de (n) de la fig. 6.1. . . .	69
A.41.Vistas de la reconstrucción final de la secuencia (n) de la fig. 6.1	70
A.42.Frames característicos seleccionados de la secuencia (ñ) de la fig. 6.1.	71
A.43.Porcentaje acumulado del Error de Reproyección para la secuencia de (ñ) de la fig. 6.1. . . .	71
A.44.Vistas de la reconstrucción final de la secuencia (ñ) de la fig. 6.1	72

Indice de tablas

6.1. Tamaño del mapa final estimado tras la aplicación del algoritmo de eliminación a posteriori aplicando los tres criterios de eliminación: error de reproyección, ángulo de paralaje y error de mínima mediana en profundidad.	30
6.2. Porcentajes de puntos espurios tras aplicar las condiciones de error de reproyección y profundidad del punto. Se incluye también el tamaño del mapa obtenido al final de la simulación.	31
6.3. Tiempo de cómputo de las diferentes etapas de eliminación de espurios del algoritmo a posteriori.	35
6.4. Tiempo de cómputo empleado en la generación del modelo 3D de la escena.	35
A.1. Porcentaje de puntos eliminados aplicando las etapas de eliminación según error de reproyección y error de mínima mediana. Secuencia (a) fig. 6.1.	44
A.2. Tiempo de cómputo de las diferentes etapas de eliminación de espurios del algoritmo a posteriori y de obtención del modelo 3D de la escena. Secuencia (a) fig. 6.1.	44
A.3. Porcentaje de puntos eliminados aplicando las etapas de eliminación según error de reproyección y error de mínima mediana. Secuencia (b) fig. 6.1.	46
A.4. Tiempo de cómputo de las diferentes etapas de eliminación de espurios del algoritmo a posteriori y de obtención del modelo 3D de la escena. Secuencia (b) fig. 6.1.	46
A.5. Porcentaje de puntos eliminados aplicando las etapas de eliminación según error de reproyección y error de mínima mediana. Secuencia (c) fig. 6.1.	48
A.6. Tiempo de cómputo de las diferentes etapas de eliminación de espurios del algoritmo a posteriori y de obtención del modelo 3D de la escena. Secuencia (c) fig. 6.1.	48
A.7. Porcentaje de puntos eliminados aplicando las etapas de eliminación según error de reproyección y error de mínima mediana. Secuencia (d) fig. 6.1.	50
A.8. Tiempo de cómputo de las diferentes etapas de eliminación de espurios del algoritmo a posteriori y de obtención del modelo 3D de la escena. Secuencia (d) fig. 6.1.	50
A.9. Porcentaje de puntos eliminados aplicando las etapas de eliminación según error de reproyección y error de mínima mediana. Secuencia (e) fig. 6.1.	52
A.10. Tiempo de cómputo de las diferentes etapas de eliminación de espurios del algoritmo a posteriori y de obtención del modelo 3D de la escena. Secuencia (e) fig. 6.1.	52
A.11. Porcentaje de puntos eliminados aplicando las etapas de eliminación según error de reproyección y error de mínima mediana. Secuencia (f) fig. 6.1.	54

A.12. Tiempo de cómputo de las diferentes etapas de eliminación de espurios del algoritmo a posteriori y de obtención del modelo 3D de la escena. Secuencia (f) fig. 6.1.	54
A.13. Porcentaje de puntos eliminados aplicando las etapas de eliminación según error de reproyección y error de mínima mediana. Secuencia (g) fig. 6.1.	56
A.14. Tiempo de cómputo de las diferentes etapas de eliminación de espurios del algoritmo a posteriori y de obtención del modelo 3D de la escena. Secuencia (g) fig. 6.1.	56
A.15. Porcentaje de puntos eliminados aplicando las etapas de eliminación según error de reproyección y error de mínima mediana. Secuencia (h) fig. 6.1.	58
A.16. Tiempo de cómputo de las diferentes etapas de eliminación de espurios del algoritmo a posteriori y de obtención del modelo 3D de la escena. Secuencia (h) fig. 6.1.	58
A.17. Porcentaje de puntos eliminados aplicando las etapas de eliminación según error de reproyección y error de mínima mediana. Secuencia (i) fig. 6.1.	60
A.18. Tiempo de cómputo de las diferentes etapas de eliminación de espurios del algoritmo a posteriori y de obtención del modelo 3D de la escena. Secuencia (i) fig. 6.1.	60
A.19. Porcentaje de puntos eliminados aplicando las etapas de eliminación según error de reproyección y error de mínima mediana. Secuencia (j) fig. 6.1.	62
A.20. Tiempo de cómputo de las diferentes etapas de eliminación de espurios del algoritmo a posteriori y de obtención del modelo 3D de la escena. Secuencia (j) fig. 6.1.	62
A.21. Porcentaje de puntos eliminados aplicando las etapas de eliminación según error de reproyección y error de mínima mediana. Secuencia (k) fig. 6.1.	64
A.22. Tiempo de cómputo de las diferentes etapas de eliminación de espurios del algoritmo a posteriori y de obtención del modelo 3D de la escena. Secuencia (k) fig. 6.1.	64
A.23. Porcentaje de puntos eliminados aplicando las etapas de eliminación según error de reproyección y error de mínima mediana. Secuencia (l) fig. 6.1.	66
A.24. Tiempo de cómputo de las diferentes etapas de eliminación de espurios del algoritmo a posteriori y de obtención del modelo 3D de la escena. Secuencia (l) fig. 6.1.	66
A.25. Porcentaje de puntos eliminados aplicando las etapas de eliminación según error de reproyección y error de mínima mediana. Secuencia (m) fig. 6.1.	68
A.26. Tiempo de cómputo de las diferentes etapas de eliminación de espurios del algoritmo a posteriori y de obtención del modelo 3D de la escena. Secuencia (m) fig. 6.1.	68
A.27. Porcentaje de puntos eliminados aplicando las etapas de eliminación según error de reproyección y error de mínima mediana. Secuencia (n) fig. 6.1.	69
A.28. Tiempo de cómputo de las diferentes etapas de eliminación de espurios del algoritmo a posteriori y de obtención del modelo 3D de la escena. Secuencia (n) fig. 6.1.	69
A.29. Porcentaje de puntos eliminados aplicando las etapas de eliminación según error de reproyección y error de mínima mediana. Secuencia (ñ) fig. 6.1.	71
A.30. Tiempo de cómputo de las diferentes etapas de eliminación de espurios del algoritmo a posteriori y de obtención del modelo 3D de la escena. Secuencia (ñ) fig. 6.1.	71