Universidad Zaragoza
1542

Trabajo Fin de Grado en Ingeniería Industrial

Demostrador VSLAM con partes rígidas y deformables

Autor

Javier Morlana Ledesma

Directores

José María Martínez Montiel

Escuela de Ingeniería y Arquitectura

2017

TRABAJOS DE FIN DE GRADO / FIN DE MÁSTER

Escuela de
Ingeniería y Arquitectura
Universidad Zaragoza

DECLARACIÓN DE AUTORÍA Y ORIGINALIDAD

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

D./Dª. Javier Morlana Ledesma,

con nº de DNI 17760685Q en aplicación de lo dispuesto en el art. 14 (Derechos de autor) del Acuerdo de 11 de septiembre de 2014, del Consejo de Gobierno, por el que se aprueba el Reglamento de los TFG y TFM de la Universidad de Zaragoza,

Declaro que el presente Trabajo de Fin de (Grado/Máster) Grado, (Título del Trabajo) Demostrador VSLAM con partes rígidas y deformables,

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

Zaragoza, 1 de Septiembre de 2017

Fdo:

Resumen

Se ha implementado un sistema de VSLAM (Simultaneous Location and Mapping with Visual sensor) que procesa secuencias de vídeo que incluyen una escena con elementos tanto rígidos como deformables. El objetivo es un programa para procesar la secuencia de vídeo obteniendo un modelo 3D de la escena y la posición de la cámara respecto del modelo construido, todo ello en tiempo real a frecuencia de vídeo.

Para ello se ha construido una escena física para el demostrador que contiene un material textil de gran riqueza visual y deformable bajo la acción de una fuerza. El marco del demostrador es rígido y se encuentra estático en la escena.

Para construir el software se ha partido del sistema ORBSLAM, que es un sistema de VSLAM para escenas rígidas, extendiéndolo para que pueda procesar las escenas que contienen el demostrador.

La parte no rígida se inicializa mediante una segmentación entre los puntos rígidos y deformables del mapa 3D del demostrador en reposo. Con los puntos identificados como deformables se construye el modelo de deformación, que consiste en una malla regular triangular. Cada frame se procesa secuencialmente, buscando y emparejando los puntos no rígidos del mapa en cada nueva imagen. Si el demostrador sufre deformaciones, aparecerá un error de reproyección en estos emparejamientos. Una optimización no lineal reduce este error modificando la posición de los nodos de la malla, estimando así la deformación ocurrida.

El sistema ha sido validado experimentalmente y es capaz de estimar pequeñas deformaciones a frecuencia de vídeo.

La implementación se ha hecho en C++ y está disponible en un repositorio privado de GitHub.

Índice

Capítulo 1

Introducción y objetivos

1.1. Introducción

Los demostradores en vivo de los sistemas de SLAM visual con "cámara en la mano" han sido clave para el desarrollo y difusión de esta tecnología [1][2][3][4]. El objetivo último de la línea de trabajo en la que se enmarca este proyecto es construir un demostrador para un SLAM visual con cámara en mano para el caso de una escena que se deforma. Este proyecto es el primer paso.

El objetivo en sentido amplio es concebir e implementar un mapa que combina partes rígidas y deformables. La parte rígida sería procesada por el sistema ORBSLAM [4], capaz de procesar miles de puntos. La parte deformable con condiciones de contorno sería procesada secuencialmente asumiendo el modelado laplaciano con inextensibilidad propuesto en [5].

La demostración consiste en una escena que está segmentada en una región que puede sufrir deformaciones rígidas y otra que no. La región deformable está completamente rodeada por la región rígida. Inicialmente toda la escena se comporta como rígida mientras la cámara monocular en mano hace una exploración de la escena y estima un modelo 3D de la misma. En una segunda etapa, la región deformable comienza a sufrir deformaciones. El sistema tiene que poder mapear todas las regiones de la escena, incluida su deformación, a frecuencia de video.

El interés en solucionar problemas de SLAM en entornos no rígidos se debe a lo bien que podría adaptarse este sistema a ciertos procedimientos médicos donde gran parte de la escena es deformable como un endoscopio. El endoscopio dispone de una cámara mediante la que se observa la escena y en ocasiones incluye herramientas que permiten realizar pequeñas intervenciones, como quitar pequeños tumores o pólipos. El problema reside en la incapacidad del endoscopio de localizarse respecto de la escena, ya que únicamente puede proporcionar su posición global. Con la incorporación de un sistema SLAM no rígido se puede conseguir localizar el endoscopio respecto de la

escena y realizar estas intervenciones con precisión e incluso de forma automatizada. Esto no se puede conseguir actualmente sin utilizar procedimientos invasivos.

1.2. Estimación escena deformable a partir de una imagen

El problema a resolver a lo largo de este proyecto consiste en la estimación de una escena deformable a partir de las imágenes provenientes de una secuencia o una cámara monocular en vivo. El objetivo final del trabajo es la obtención de una malla capaz de reproducir los movimientos de un sólido no rígido en una escena que combina elementos rígidos y no rígidos.

La demo es capaz de procesar pequeñas deformaciones en vivo, lo cual de por sí ya es un gran avance, dado el estado del arte en sistemas de SLAM no rígido. Para ello se emplea el software ORBSLAM, que construye mapas de miles de puntos rígidos y sitúa la posición de la cámara respecto a ellos en cada imagen. Este software ha sido extendido para lograr lo propuesto en este proyecto.

En la escena tendremos puntos que se deformen y puntos que permanezcan fijos. Los puntos fijos, denominados rígidos, son empleados por ORBSLAM para localizar a frecuencia de vídeo la cámara. Los puntos que se deformen, no rígidos, sirven para estimar el modelo geométrico de la parte de la escena que sufre deformación.

Para simplificar el problema, se parte de un demostrador (figura 1.1). Este demostrador consta de una caja a la que se le ha practicado un agujero. A esta hendidura se ha adherido un material textil de gran riqueza visual para que el programa sea capaz de reconocer gran cantidad de puntos de interés. Este material textil será el único elemento deformable, siendo el resto de la escena rígida. Para inicializar el sistema se parte de una posición de reposo (sin deformación), en la que la tela permanece tensa y plana. El software tiene que estimar la escena rígida y el modelo de deformación para la parte no rígida, y después se pasa a la etapa la que la tela será deformada.

Para estimar la deformación, el programa necesita conocer la posición de la cámara respecto del mapa, así como emparejar los puntos no rígidos del mapa con sus observaciones sobre la imagen. Teniendo estos datos, la optimización no lineal es capaz de transformar el modelo deformable para que este estimen las deformaciones que se producen en el demostrador.

En resumen, los procedimientos necesarios para estimar el movimiento de la cámara y la deformación son:

Posición de la cámara y construcción del mapa rígido: es proporcionada a frecuencia de video por el ORBSLAM, este sistema también nos proporciona la

(a) Vista frontal (b) Vista de perfil

Figura 1.1: Demostrador. La cara plana donde está el círculo en la vista frontal se denomina marco

posición de los puntos de contorno del modelo deformable.

Modelo deformable: La forma de modelar la escena es través de una malla situada sobre un plano. Dicho plano es construido a través de los puntos detectados en el mapa y la imagen. La malla se ajusta a la zona deformable y se encuentra plana en situación de reposo. Cuando comienza la deformación del modelo físico, la malla se deforma con él. La malla regular escogida es triangular y los puntos se representan respecto a ella con coordenadas baricéntricas. Se detalla en el capítulo 3.

Segmentación de la zona no rígida: Para situar la malla en el mapa sobre la zona deformable se ha realizado una segmentación de los puntos obtenidos por ORBS-LAM a través de una homografía que relaciona cada fotograma de la secuencia con una imagen canónica referencia, por medio de una triangulación de Delaunay. De esta manera se tienen tres conjuntos de puntos: puntos rígidos, utilizados por ORBSLAM para localizarse; puntos no rígidos, que serán los que se muevan con la malla al deformar el modelo; y puntos de contorno, que son rígidos también y se emplean para el cálculo del plano. Se aborda en el capítulo 2.

Optimización: En la cámara identificamos unos puntos de medida. La posición de los puntos de medida sobre la imagen dependen de la deformación de la malla. Mediante optimización no lineal del error de reproyección resolvemos el problema inverso para determinar cual es la deformación de la malla que mejor explica las observaciones. El problema está subdeterminado por lo que deberemos añadir un regularizador para restringir la solución de acuerdo con la los aprioris disponibles. Asúmanos una malla laplaciana e inextensible [5]. Se detalla en el capítulo 4.

Capítulo 2

Segmentación de la parte deformable de la escena

En este capítulo se van a describir los procedimientos para realizar la segmentación entre los puntos rígidos y no rígidos.

El demostrador incluye el diseño de un elemento plano con parte rígida y parte deformable, que llamamos marco. La figura 1.1 muestra el marco construido donde la parte deformable de la escena es un circulo plano rodeado por un plano rígido. El plano rígido tiene una textura visual predefinida. La textura visual se explota para poder segmentar la escena en parte deformable y parte no deformable, este es paso previo para poder modelar la parte deformable de la escena mediante una malla y también para poder localizar la cámara a frecuencia de vídeo.

A continuación se detalla como mediante una homografía podemos identificar en cualquier imagen que contenga la parte frontal del marco y cual es la parte rígida de la parte deformable. También se obtienen los puntos de contorno a partir de los puntos anteriores, calculando una triangulación de Delaunay.

2.1. Homografía entre el fotograma y la imagen canónica del marco

Una homografía es una matriz H_{3x3} que establece una relación entre dos imágenes de un mismo objeto plano. Mediante esta matriz se puede establecer una correspondencia biunívoca entre dos imágenes que contengan la parte frontal del marco. Mediante un proceso de calibración definimos una imagen canónica del marco donde de forma manual segmentamos la parte deformable. Posteriormente, para cada imagen de la secuencia de vídeo, calcularemos una homografía que relaciona el fotograma del vídeo con la imagen canónica.

Para la obtención de la correspondecia son necesarios una serie de datos:

1. Puntos de interés (keypoints) detectados en el fotograma(frame) y en la imagen canónica. Los puntos a segmentar son los ORB que detecta el programa en la imagen en la que se inicia el procesamiento no rígido.

2. Emparejamientos (matches) entre los puntos del fotograma y la imagen canónica.

Una vez obtenidos los puntos y sus correspondencias, se ha de resolver un sistema de ecuaciones que dará lugar a la matriz H_{fc}. En principio solo se necesitan cuatro correspondencias, pero se usan más y se aplica RANSAC para mejorar la solución [6]. El resultado obtenido puede observarse en la Figura 2.1.

Las ecuaciones que permiten transformar las coordenadas del fotograma en las coordenadas de la imagen canónica (eq. 2.1) y viceversa (eq. 2.2), así como la relación entre ambas transformaciones (eq. 2.3) son:

$$X_c = H_{cf} X_f \tag{2.1}$$

$$X_f = H_{fc} X_c \tag{2.2}$$

$$H_{cf} = (H_{fc})^{-1} \tag{2.3}$$

Donde X_c y X_f representan las coordenadas homgéneas de un punto en la imagen canónica y en el fotograma, respectivamente. H_{cf} permite llevar un punto del fotograma a la imagen canónica, y H_{fc} permite lo contrario.

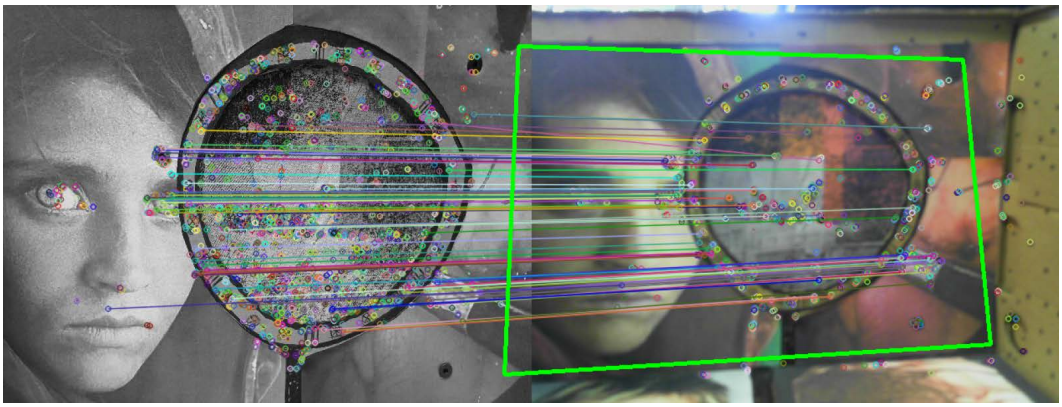

Figura 2.1: Keypoints y matches entre la imagen canónica y el fotograma de la secuencia. El rectángulo verde representa donde se ve la imagen canónica en la secuencia

2.2. Segmentación entre puntos rígidos y no rígidos

Tras obtener la correspondencia, se realiza la primera segmentación de puntos. La forma de funcionamiento de ORBSLAM es proyectar los puntos que tiene en el mapa

3D y buscarles emparejamientos con keypoints de la imagen. Si hay emparejamiento, se dibuja el keypoint. De manera que todos los keypoints que aparecen en la imagen se identifican con puntos del mapa. Debido a esta relación los keypoints se denominarán puntos de aquí en adelante.

Sólo se segmentan por tanto los puntos que han sido emparejados en el fotograma actual de la secuencia. Esta elección, en lugar de proyectar el mapa completo y realizar los cálculos que se exponen a continuación, se debe a que si se proyectase el mapa completo, podrían aparecer en la imagen puntos que se encuentran detrás del demostrador. De manera que se correría el riesgo de tomar como puntos no rígidos a puntos que no pertenecen a la zona deformable.

Para realizar la segmentación, se aplica la transformación H_{cf} a los puntos del fotograma de la secuencia de vídeo. Así se tiene donde están dichos puntos en la imagen canónica (figura 2.2a). Ahora se llevan a la imagen de calibración (figura 2.2b), que es idéntica a la canónica con la salvedad de que la zona deformable ha sido coloreada en rojo.

(a) Imagen canónica (b) Imagen de calibración

Figura 2.2: Imágenes para calcular la segmentación

Por tanto, si el punto transformado se encuentra sobre un pixel de color rojo de la imagen de calibración, el punto es no rígido. Los demás puntos se consideran rígidos (figura 2.3).

2.3. Identificación de los puntos de contorno

Partiendo de un conjunto de puntos segmentados como rígidos y no rígidos, se calcula una triangulación de Delaunay para calcular la conectividad de los puntos (figura 2.4). Los puntos de contorno serán aquellos puntos rígidos que estén conectados a través de una arista de Delaunay con un punto no rígido, y que serán empleados para generar el plano en el modelo de deformación (capítulo 3).

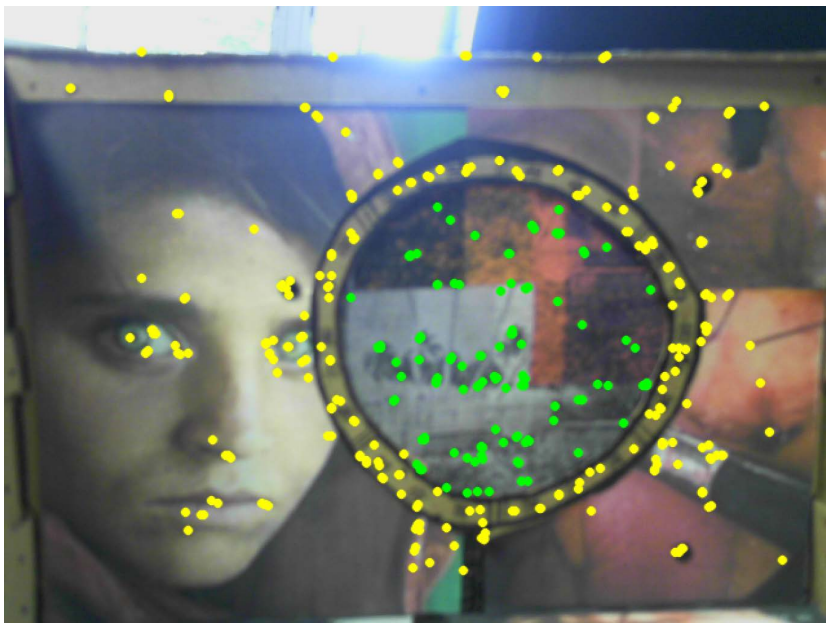

Figura 2.3: Primera segmentación. En verde: puntos no rígidos. En amarillo: puntos rígidos

Una triangulación hace referencia a la subdivisión del plano en triángulos, usando los puntos dados como vértices. Un mismo conjunto de puntos puede tener diferentes triangulaciones, pero la triangulación de Delaunay tiene unas propiedades que la óptima. Estas son:

- Tres puntos del conjunto de puntos son vértices de un mismo triángulo de la triangulación de Delaunay si y sólo si puede trazarse un círculo cuyo contorno contenga esos tres puntos y no contenga otros puntos del conjunto en su interior.

- Dos puntos del conjunto definen una arista de la triangulación si y sólo si es posible trazar un círculo cuyo contorno contenga a esos dos puntos pero en su interior no contenga ningún otro punto del conjunto.

Así que se puede decir que la triangulación de Delaunay se diferencia de otras de esta forma: una triangulación de un conjunto de puntos es de Delaunay si y sólo si la circunferencia circunscrita de cualquier triángulo de la triangulación no contiene puntos del conjunto [7].

Tras calcular la triangulación, se tienen todos los puntos conectados entre sí formando el mayor número de triángulos posibles sin que se crucen sus aristas, dando esto lugar a triángulos regulares formados con los puntos más próximos entre sí.

Existen tres tipos de conexiones (figura 2.4):

- Conexión entre puntos no rígidos (NR-NR), representada en verde. Son las conexiones entre puntos que están dentro del círculo de tela.

– Conexión entre puntos rígidos (R-R), representada en amarillo. Son las conexiones entre los puntos del marco del demostrador o de la escena exterior.

– Conexión entre puntos rígidos y no rígidos (R-NR), representada en fucsia. Son conexiones entre puntos del extremo del círculo de tela y puntos del marco. Los puntos del marco presentes en estas conexiones pasan a considerarse puntos de contorno, representados en rojo.

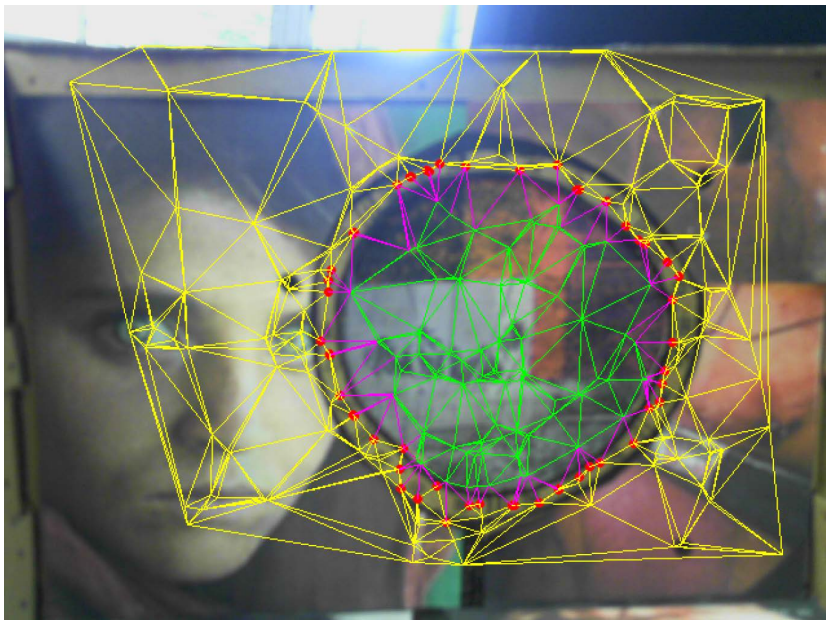

Figura 2.4: Triangulación de Delaunay usando los puntos del fotograma. Verde: conexión NR-NR. Amarillo: conexión R-R. Fucsia: conexión R-NR. Puntos en rojo: puntos de contorno

Capítulo 3

Modelo de deformación

Es este capítulo se detalla cómo generar la malla con la que se modelan las deformaciones de la zona no rígida. La malla está relacionada con los puntos de contorno y con los puntos los no rígidos.

El modelo deformable escogido es una malla regular triangular de 11x11 nodos, componiéndose de nodos rígidos y deformables, según dónde se sitúen sobre el marco del demostrador. Los nodos rígidos permanecen fijos durante todo el procesamiento secuencial, mientras que los nodos deformables se mueven tratando de explicar la deformación.

Al ser una malla regular, lo más probable es que los puntos no rígidos del mapa no coincidan con ninguno de los nodos de la malla. Por ello es necesario un mecanismo de interpolación que relacione los puntos no rígidos que hemos medido con los nodos de la malla. Esta relación se consigue a través de las coordenadas baricéntricas, que expresan las coordenadas del punto no rígido en función de los 3 vértices de la celda triangular en la que se encuentra.

Para obtener el modelo deformable se realizan una serie de procedimientos que se explican a continuación.

3.1. Construcción del plano a partir de los puntos de contorno

Dado que el marco del demostrador es plano, el modelo deformable debe estar situado en un plano formado por puntos detectados en dicho marco. Los puntos escogidos para tal efecto son los puntos de contorno, ya que éstos puntos van a estar fijos durante la deformación y no habrá opción de incluir un punto de la escena que no pertenezca al marco. Esto último podría ocurrir si cogiésemos todos los puntos rígidos para calcular el plano.

A la hora de construir el plano, hemos de tener en cuenta que los puntos del mapa

estimados por ORBSLAM no son totalmente exactos, y por tanto, no están situados en el mismo plano. Por ello se aplica el algoritmo RANSAC con 10 iteraciones, en el que se calculan diferentes hipótesis iniciales para el plano a partir de tríos de puntos de contorno.

Para cada trío de puntos se calcula el plano correspondiente empleando coordenadas homogéneas para los puntos y resolviendo el correspondiente sistema de ecuaciones homogéneo mediante un SVD. Después se calculan las distancias de cada uno de los demás puntos al plano calculado. La forma de comparar los planos entre sí es con la mediana de las distancias cuadráticas punto-plano. El plano cuya mediana sea mínima será utilizado para rechazar los puntos de contorno que no pertenecen al plano (espúreos).

La aceptación o rechazo de un punto viene dada por la desviación típica (s) para el plano empleado que se estima a partir de la mediana de los residuos cuadráticos (eq. 3.1). Si la distancia de un punto dividida por la desviación típica supera un cierto umbral, entonces se rechaza (eq. 3.2).

$$s = 1{,}4826(1 + \frac{5}{n-k})\sqrt{med(dist_i^2)} \tag{3.1}$$

$$\left|\frac{dist_i}{s}\right| > 2{,}5 \tag{3.2}$$

Donde n es el número total de puntos y k es el número de puntos usados para crear el modelo, en este caso, tres.

Tras descartar los espúreos, se calcula finalmente el plano con todos los puntos de contorno, resolviendo un nuevo SVD. Y procediendo de la misma manera, descartamos los puntos no rígidos considerados como espúreos.

3.2. Obtención de la malla deformable

Sobre el plano calculado se coloca una malla triangular de 11x11 nodos. Esta malla necesita estar sobre los puntos de contorno y los no rígidos para ser capaz de procesar las deformaciones. Además, se realizará la proyección ortogonal de los puntos sobre el plano para poder calcular las coordenadas baricéntricas de cada punto no rígido respecto de los nodos.

Para ello será necesario calcular el cambio de coordenadas de la referencia global a la referencia del plano, y viceversa. Trabajando en coordenadas homogéneas, el cambio

de referencia se puede representar mediante una matriz T_{4x4}:

$$T_{4x4} = \begin{bmatrix} R_{3x3} & t_{3x1} \\ 0_{1x3} & 1 \end{bmatrix} \tag{3.3}$$

Donde R_{3x3} es la matriz de rotación que define el giro entre los dos sistemas de coordenadas, y t_{3x1} es el vector de traslación que define el desplazamiento entre los orígenes de los dos sistemas de coordenadas.

Primeramente, se resuelve un SVD llenando la matriz A con las coordenadas del plano. Esto da lugar a tres puntos, que son tres puntos cualesquiera pertenecientes al plano: P_1, P_2 y P_3. A partir de ellos se define la referencia.

Con el primer punto se define el vector $\vec{t}$, traslación entre el origen de la referencia global y la del plano. El vector $P2\vec{P}1$ es el eje x del plano, y lo denotamos como $\vec{n}$. El vector $P3\vec{P}1$ es un eje temporal con el cual obtendremos un eje perpendicular al plano, $\vec{a}$, mediante el producto cruzado con $\vec{n}$. Finalmente, haciendo otro producto cruzado entre $\vec{n}$ y $\vec{a}$, obtenemos el eje y del plano, denotado como $\vec{o}$.

Con estos 4 vectores ya se puede contruir la matriz $T_{P\leftarrow W}$, que define el cambio de coordenadas de la referencia global al plano.

$$T_{P\leftarrow W} = \begin{bmatrix} \begin{bmatrix} n_1 & o_1 & a_1 \\ n_2 & o_2 & a_2 \\ n_3 & o_3 & a_3 \end{bmatrix} & \begin{bmatrix} t_1 \\ t_2 \\ t_3 \end{bmatrix} \\ \begin{matrix} 0 \end{matrix} & \begin{matrix} 1 \end{matrix} \end{bmatrix} \tag{3.4}$$

Y la matriz $T_{W\leftarrow P}$, que define el cambio de la referencia del plano a la global:

$$T_{W\leftarrow P} = \begin{bmatrix} \begin{bmatrix} \\ R^T \\ \\ \end{bmatrix} & \begin{bmatrix} | \\ -R^T t \\ | \end{bmatrix} \\ \begin{matrix} 0 \end{matrix} & \begin{matrix} 1 \end{matrix} \end{bmatrix} \tag{3.5}$$

La malla se sitúa de forma que comience en la x e y mínimas de todos los puntos de contorno y se extienda abarcándolos a todos ellos. Para ello se proyectan ortogonalmente los puntos de contorno en el plano, primero aplicando el cambio de coordenadas y después fijando la coordenada $z = 0$.

$$X_{P,proy} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} T_{P\leftarrow W} X_W \tag{3.6}$$

Siendo $X_{P,proy}$ las coordenadas de los puntos de contorno en la referencia del plano tras proyectarse ortogonalmente, y X_W las coordenadas en la referencia global. Con $X_{P,proy}$ se comparan todas las coordenadas, guardándose x_{min} e y_{min}.

Desde las mínimas coordenadas se definen los nodos de la malla en coordenadas homogéneas, y luego se llevan a la referencia global (eq. 3.7). Los nodos se separan mediante una distancia constante d y forman celdas triangulares, teniendo cada fila 20 celdas.

$$N_W = T_{W \leftarrow P} N_P \tag{3.7}$$

N_W y N_P es la posición de los nodos de la malla en coordenadas homogéneas para la referencia global y la del plano, respectivamente.

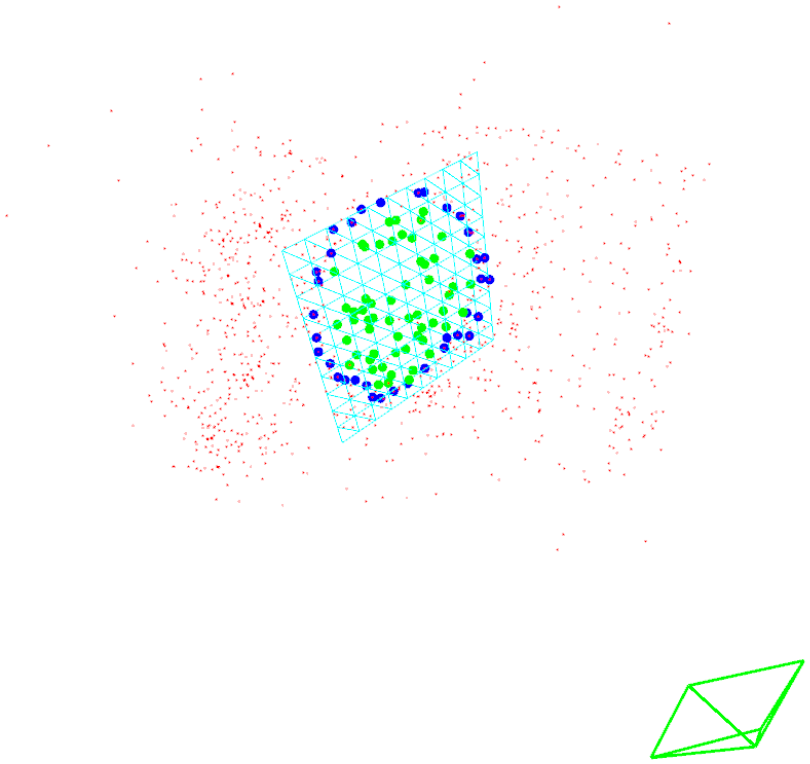

Figura 3.1: Malla construida sobre la escena y vista por la cámara. Azul claro: malla. Verde: puntos no rígidos. Azul: puntos de contorno. Rojo: puntos rígidos

3.3. Relación puntos no rígidos con la malla

Con los puntos ya segmentados y el modelo construido, se relacionan los puntos no rígidos con los nodos de la malla.

Cada celda de la malla se guarda mediante los tres índices de los nodos por los que está formada. Cada punto no rígido está asociado a una celda.

Por otra parte, cada punto está relacionado con sus vecinos mediante arcos. A la hora de optimizar la malla se debe conocer cuáles son los vecinos de cada punto para

formular el modelo de deformación.

3.3.1. Coordenadas baricéntricas

Las coordenadas baricéntricas representan las coordenadas de un punto perteneciente a un triángulo a través de las coordenadas de sus vértices. Dado un punto X_i y los vértices N_{i_1}, N_{i_2} y N_{i_3}, su representación a través de las coordenadas baricéntricas α_i, β_i y γ_i se expresa como:

$$X_i = \alpha_i N_{i_1} + \beta_i N_{i_2} + \gamma_i N_{i_3} \tag{3.8}$$

Cada punto no rígido se sitúa dentro de una celda triangular y se expresa como combinación de lineal de sus vértices con las coordenadas baricéntricas. Si un nodo se mueve en la optimización, todos los puntos no rígidos que estén asociados a ese nodo se moverán con él.

Capítulo 4

Procesamiento secuencial

En este capítulo se describen los procedimientos que realiza el algoritmo en cada frame a lo largo de la secuencia.

Se comienza con una inicialización en la que se reconoce la escena mientras se crean puntos del mapa. Cuando el mapa dispone de suficientes puntos, se realiza la segmentación y se construye el modelo deformable.

Tras terminar la inicialización, se realiza para cada imagen la predicción y el emparejamiento de los puntos no rígidos, así como la optimización no lineal del modelo deformable.

4.1. Inicialización

4.1.1. Mapa rígido

Durante los primeras imágenes de la secuencia, ORBSLAM crea keyframes y puntos del mapa 3D a través de la triangulación entre keypoints emparejados en diferentes frames. A su vez, calcula la posición de la cámara respecto al mapa en cada frame (figura 4.1).

4.1.2. Mapa no rígido

Cuando se considera que el mapa tiene puntos suficientes para localizarse a lo largo de la secuencia y realizar la estimación de las deformaciones, se pasa al modo de localización. Dentro de este modo, ORBSLAM ya no trata de crear nuevos puntos ni keyframes, su única función es la de localizarse. Esto resulta muy útil dado ya que el mapa se congela, evitando la aparición de nuevos puntos en la zona deformable, pero a su vez sigue calculando la pose de la cámara en cada frame. Dejar el mapa fijado también reduce notablemente la carga computacional del sistema ORBSLAM.

Sobre el mismo frame en el que se ha realizado el cambio de modo se ejecuta la segmentación de los puntos de dicha imagen. En el frame escogido debe aparecer la

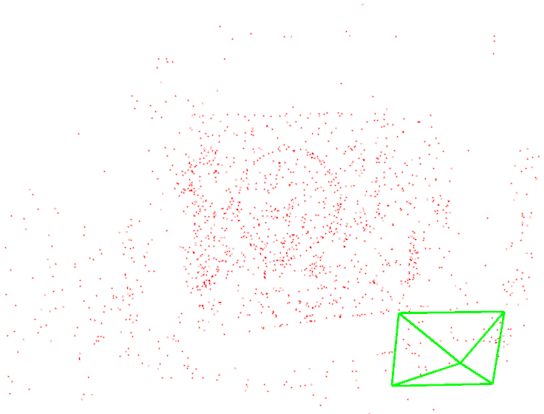

Figura 4.1: Construcción del mapa rígido. En rojo: puntos del mapa. En verde: posición de la cámara

zona a deformar, sino es imposible realizar la segmentación y los cálculos posteriores. Los puntos que se obtienen de esta segmentación se guardan en el mapa no rígido y se eliminan del mapa rígido para que ORBSLAM no trate de localizarlos, ya que se van a mover durante la deformación. A partir de estos puntos se calcula el modelo deformable explicado en el capítulo 3.

Además, es necesario guardar una subimagen (patch) centrada en cada punto no rígido segmentado de la imagen. Este patch será utilizado para hacer un emparejamiento por correlación en la sección 4.2.

En los 20 frames consecutivos se realiza nuevamente un reconocimiento de los puntos no rígidos, ya que en un único frame ORBSLAM no detecta todos los puntos que tiene en su mapa. Dichos puntos no son necesarios para el cálculo no rígido, simplemente se eliminan por el mismo motivo que antes.

La malla calculada se representa sobre la escena 3D, al igual que los puntos de contorno y los nodos deformables. Sobre el frame actual se representan también los nodos de la malla, así como las predicciones de los puntos del mapa.

4.2. Emparejamiento secuencial mediante búsqueda activa

El procesamiento secuencial de cada imagen incluye una etapa de emparejamiento, que consiste en identificar donde son detectados los puntos del mapa deformable en la nueva imagen. Se emplea emparejamiento por correlación en la imagen y búsqueda activa de los puntos del mapa. La búsqueda activa incluye la estimación de la región en píxeles de la imagen donde es probable que aparezca la imagen de un punto del mapa deformable.

4.2.1. Predicción del punto en la imagen

ORBSLAM nos proporciona la pose de la cámara en cada frame T_{cw} durante toda la secuencia, y también disponemos de los parámetros de calibración de la cámara, K. Con estos datos basta con construir la matriz de proyección $P = KT_{cw}$ y aplicarla a todos los nodos de la malla (eq. 4.1).

Una vez se tienen los nodos proyectados, se aplican las coordenadas baricéntricas a los 3 nodos correspondientes y se obtiene la proyección del punto no rígido en la imagen (eq. 4.2). De aquí en adelante denominaremos a esta proyección como predicción.

$$n_{imagen} = PN_{mapa} \tag{4.1}$$

$$x_{imagen} = \alpha N_1 + \beta N_2 + \gamma N_3 \tag{4.2}$$

Donde x_{imagen} es la predicción del punto no rígido en la imagen, mientras que n_1, n_2 y n_3 son las coordenadas de las proyecciones en la imagen de los nodos de la celda a la que pertenece el punto.

4.2.2. Emparejamiento

El emparejamiento por correlación consiste en buscar un área determinada en una imagen que coincida con una imagen modelo previamente definida. El primer emparejamiento que se intentó fue con características ORB, pero los emparejamientos eran poco estables cuando se producía deformación.

Para cada punto no rígido en el mapa se un patch que permite identificar el punto el la imagen. Para emparejar, se busca por correlación en una región donde es probable que el punto sea observado. Para cada pixel de la región de búsqueda se evalúa la correlación normalizada con el patch del punto buscado. El punto emparejado se sitúa

donde la corrrelación normalizada es máxima. Al punto emparejado se le denomina observación, x_i.

Se emplea la implementación de OpenCV de la búsqueda por correlación.

4.3. Optimización no lineal

Una vez que se han obtenido el modelo de deformación y los emparejamientos, ya se está en disposición de realizar la estimación del movimiento a través de una optimización no lineal. Dicha optimización se resuelve mediante la librería g2o [8].

En dicha optimización se trata de minimizar el error, representado por la siguiente expresión:

$$\begin{aligned} e &= \frac{1}{N_e}\lambda_R \sum_{i=0}^{N_e} \left(P \begin{bmatrix} N_{i1} & N_{i2} & N_{i3} \end{bmatrix} \begin{bmatrix} \alpha_i \\ \beta_i \\ \gamma_i \end{bmatrix} - x_i \right)^2 \\ &+ \frac{1}{d^2}\lambda_L \sum_{i=0}^{N_n} (\delta_{i,1} - \delta_{i,0})^2 \\ &+ \frac{1}{d^2}\lambda_I \sum_{i=0}^{N_n} (d(N_i, N_j) - d_0(N_i, N_j))^2 \end{aligned} \tag{4.3}$$

Como se observa, la función del error se compone de varios términos que se exponen a continuación:

4.3.1. Reproyección

El primer término de la optimización corresponde al error de reproyección. Este es el error producido al obtener las predicciones mediante una proyección y emparejarlas con las observaciones del frame. Su expresión es:

$$e_{reproyección} = \frac{1}{N_e}\lambda_R \sum_{i=0}^{N_e} \left(P \begin{bmatrix} N_{i1} & N_{i2} & N_{i3} \end{bmatrix} \begin{bmatrix} \alpha_i \\ \beta_i \\ \gamma_i \end{bmatrix} - x_i \right)^2 \tag{4.4}$$

Donde N_e es el número de emparejamientos, λ_R es el peso otorgado a este error, P es la matriz de proyección del mundo a la imagen, N_i son las coordenadas de los nodos de la celda, α, β, γ las coordenadas baricéntricas y x_i la observación con la que se le ha emparejado.

En el estado de reposo este error se supone nulo, dado que los puntos no se han movido de su lugar inicial. Al comenzar la deformación, los puntos del demostrador se desplazan de su posición original (figura 4.2). Esto provoca que la predicción y

la observación no estén exactamente en la misma posición, dando lugar al error de reproyección. La optimización trata de mover la malla de forma que las predicciones se sitúen lo más cerca posible de las observaciones con las que se han emparejado.

Como no siempre se empareja el mismo número de puntos, se utiliza el promedio para normalizar esta componente del error, evitando que este error tenga mayor ponderación cuando hay más emparejamientos.

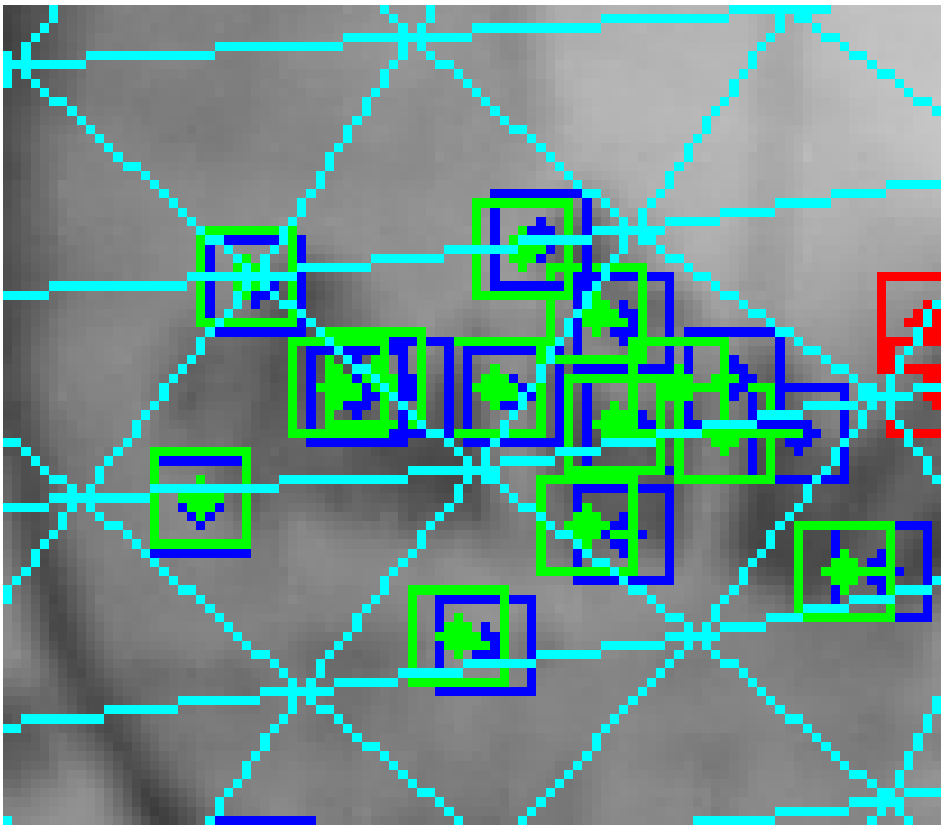

Figura 4.2: Al producirse deformación, las observaciones (azul oscuro) se alejan de las predicciones con las que se emparejan (verde) dando lugar al error de reproyección

4.3.2. Laplaciano

Al usar una cámara perspectiva, existen infinitas soluciones capaces de explicar la deformación producida. Es decir, el problema está subdeterminado. Por ello se añaden regularizadores que permiten obtener la solución.

El primero de ellos es el regularizador laplaciano. Este regularizador trata de mantener la curvatura presente en la malla antes y después de la optimización. Se calculan unas coordenadas laplacianas de cada nodo respecto de sus vecinos, y se trata de que se mantengan constantes tras optimizar. La expresión del regularizador es:

$$e_{laplaciano} = \frac{1}{d^2}\lambda_L \sum_{i=0}^{N_n}(\delta_{i,1} - \delta_{i,0}) \tag{4.5}$$

Donde N_n es el número de nodos presentes en la malla, λ_L es el peso otorgado al error y $\delta_{i,0}$, $\delta_{i,1}$ son las coordenadas diferenciales de un nodo de la malla laplaciana antes y después de la optimización, respectivamente. d es la distancia inicial entre dos nodos contiguos en la malla en reposo. Su función eliminar el factor de escala geométrica.El ORBSLAM, al ser monocular, trabaja sin unidades y calcula una escena de tamaño diferente cada vez que se inicia el sistema, de este modo, este cambio de tamaño no afecta a la ponderación.

Las coordenadas diferenciales de un nodo se calculan como sigue [9]:

$$\delta_i = N_i - \frac{1}{d_i} \sum_{j \epsilon N(i)} N_j \tag{4.6}$$

Siendo N_i las coordenadas del nodo que se estudia, d_i el número de vecinos inmediatos ($N(i)$) que tiene, y N_j las coordenadas de cada nodo vecino.

De esta manera se consigue que los nodos de la malla se muevan correladamente unos con otros. Si un nodo se mueve, sus vecinos se desplazarán condicionados por él.

4.3.3. Inextensibilidad

El regularizador de inextensibilidad trata de mantener constante la distancia entre dos nodos que se encuentran unidos en la malla. Se limita la extensión de de la malla, que aunque pudiesen explicar la deformación observada, serían incorrectas.

$$e_{inextensibilidad} = \frac{1}{d^2} \lambda_I \sum_{i=0}^{N_n} \left(d(N_i, N_j) - d_0(N_i, N_j) \right)^2 \tag{4.7}$$

Donde N_n es el número de nodos presentes en la malla, λ_I es el peso otorgado al error y $d_0(v_i, v_j)$, $d(v_i, v_j)$ es la distancia entre dos nodos vecinos antes y después de la optimización, respectivamente. d tiene la misma función que en el regularizador laplaciano.

Capítulo 5

Validación experimental

Para validar experimentalmente el proyecto se han realizado diversas secuencias en vivo. Se observa cualitativamente que aunque el movimiento estimado no es exacto, el modelo es capaz de explicar bastante bien pequeñas deformaciones producidas en el demostrador. A su vez, se ha realizado un análisis de tiempos que demuestra que el sistema puede funcionar a frecuencia de video de 30Hz.

5.1. Estimación de la deformación

El modelo consigue demostrar cualitivamente la deformación producida. Para ello han sido necesarias algunas modificaciones de los parámetros de ORBSLAM, así como un ajuste en los pesos de cada error a la hora de optimizar.

El modo de funcionamiento estándar de ORBSLAM utiliza 8 niveles de escala para detectar puntos de interés. El programa pasa la imagen por sucesivos filtros y pasa el detector para que dispare con cada escala. De esta manera se obtienen puntos en diferentes escalas, haciendo más robusto el sistema e invariante a la escala.

En este caso, al aplicar varias escalas, se producían duplicaciones de puntos de interés en la zona no rígida, ya que aparecían muchos puntos en una pequeña región. Esto producía emparejamientos incorrectos y errores en la optimización. Por ello ahora los puntos detectados por ORBSLAM se encuentran en la escala más fina. Así los puntos escogidos son muy buenos y no se duplican, a costa de perder invarianza a la escala y peor inicialización.

Además de cambiar la escala, para los puntos no rígidos se ha realizado el emparejamiento a través de correlación, como se ha explicado en la sección 4.2. Esta forma de emparejar, aunque es más cara computacionalmente, empareja en todos los frames casi todos los puntos no rígidos obtenidos. Se muestra una comparativa entre ambos modos de emparejamiento en la tabla 5.1.

La correlación permite un emparejamiento más estable que el ORB, que pierde los

Método	Puntos	Emparejamientos (%)
ORB	30-80	$\simeq 50$
Correlación	30-80	> 90

Tabla 5.1: Comparación entre los métodos de emparejamiento

emparejamientos en cuanto se deforma en la zona en la que estaba emparejando. El número de puntos con la escala fina es unos 4 veces menos que los que se detectaban con todas las escalas.

Los experimentos funcionan correctamente para los pesos de error que aparecen en la tabla 5.2. Además, se ha implementado un kernel robusto de Huber que da menos ponderación al error cuando este supera los 0.5 pixeles. Así si un punto se ha emparejado mal o resulta de una deformación demasiado grande, su error se tiene en cuenta pero en menor medida.

Error	Peso
Reproyección (λ_R)	10
Laplaciano (λ_L)	100
Inextensibilidad (λ_I)	10

Tabla 5.2: Asignación de pesos para el error

El sistema implementado funciona correctamente para pequeñas deformaciones producidas en el demostrador. En la figura 5.1 podemos observar la malla en la imagen y en la escena 3D cuando está en la posición de reposo. En la figura 5.2 se ha aplicado una fuerza por detrás del demostrador, deformando la tela hacia delante. Tanto en la imagen como en la escena 3D se puede apreciar el movimiento de la malla para modelar esta deformación.

El sistema trabaja mejor si tiene cierto ángulo respecto del marco, ya que así se observa mejor la deformación. También trabaja mejor con la cámara estática, aunque puede hacerlo en movimiento también.

Para deformaciones mayores es posible que el sistema se pierda ya que el error de reproyección será mucho mayor y la localización es peor al reducir las escalas.

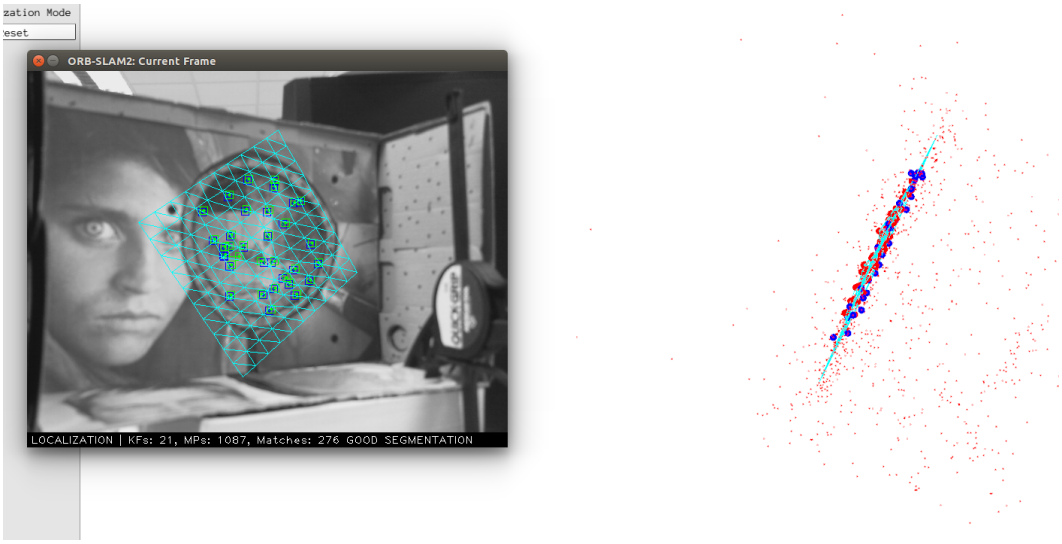

Figura 5.1: Posición de reposo: la malla está perfectamente plana

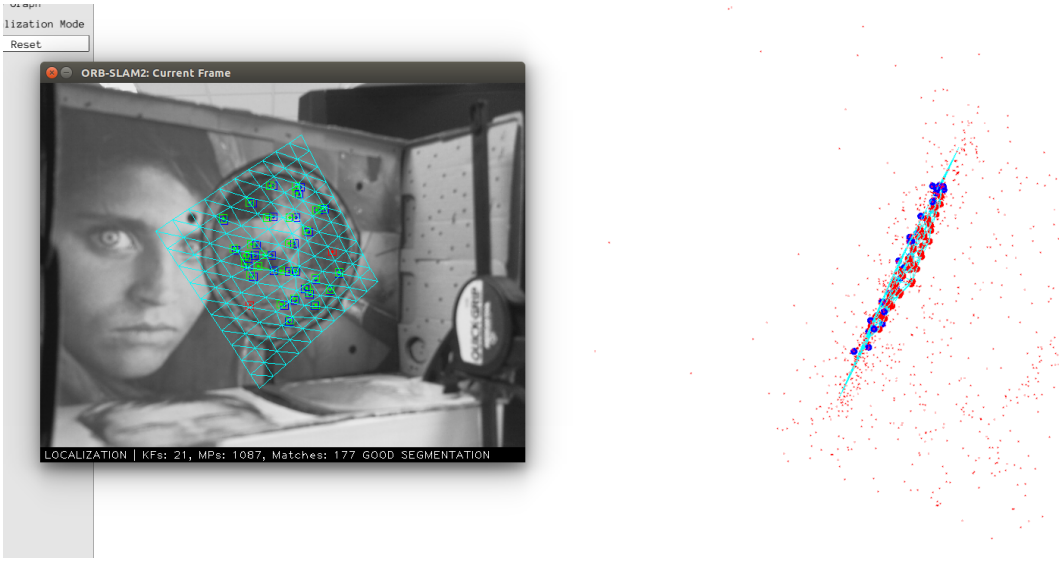

Figura 5.2: Deformación en el centro de la malla: puede observarse tanto en la imagen como en el mapa de puntos que el modelo se deforma

5.2. Análisis de tiempos

Para que el sistema sea capaz de trabajar a frecuencia de vídeo necesitamos que la suma del tiempo de todos los cálculos que se realizan cada frame sea inferior a 30 ms, que es el tiempo que pasa entre un frame y el anterior.

El cálculo de la segmentación y la eliminación no influyen en este objetivo, ya que se realizan una única vez al inicio del procesamiento no rígido, pero se han calculado por completitud (table 5.3).

Los procesos que se ejecutan cada frame de la secuencia son: predicción de puntos en la imagen, emparejamiento de estos con observaciones de la misma y optimización no lineal del error. Esto correspondería el procesamiento no rígido, al cual hay que

Proceso realizado	Tiempo (ms)
Segmentación	370
Eliminación	120

Tabla 5.3: Tiempos de cálculo para la segmentación y la eliminación de puntos no rígidos

añadirle el propio tiempo que emplea ORBSLAM en tomar la imagen, emparejar los puntos y hallar la posición de la cámara. Estos resultados se observan en la siguiente tabla:

Proceso realizado	Tiempo (ms)
Predicción	0.104
Emparejamiento	2.591
Optimización	7.407
Procesamiento no rígido	10.103
Procesamiento total	21.213
Frec. de vídeo	30

Tabla 5.4: Tiempos de cálculo para el procesamiento secuencial

Se puede observar que la suma total del tiempo de procesamiento es mucho menor que la frecuencia de vídeo.

Capítulo 6

Conclusiones y líneas futuras

6.1. Conclusiones

Este trabajo supone una de las primeras aproximaciones a la manipulación de mapas que combinan elementos rígidos y no rígidos. El sistema ha sido implementado sobre el software de SLAM en vivo ORBSLAM [4], dotándole de la capacidad para procesar un sólido deformable que es capaz de reconocer al pasarle una imagen de referencia de dicho objeto. De este software se ha obtenido la posición de la cámara cada frame y el mapa rígido inicial.

Se ha conseguido que el sistema sea capaz de modelar pequeñas deformaciones en tiempo real emparejando de forma estable prácticamente todos los puntos reconocidos como no rígidos. El sistema es capaz de estimar la deformación incluso con la cámara en movimiento, aunque es más preciso una vez que está fija.

A pesar de lo conseguido, hay ciertas áreas de mejoras posibles:

- ORBSLAM es un software muy preciso, pero aún así los puntos que se obtienen no son totalmente exactos. Esto produce cierto ruido que hace que los emparejamientos no sean perfectos en la posición de reposo.
- La posición de la cámara presenta cierta vibración incluso en reposo (jitter), posiblemente porque se han eliminado muchos puntos y no se utiliza el sistema de procesamiento de imagen multiescala. Se podría mejorar si se emplea multiescala para la localización de la cámara y escala fina para la parte no rígida.
- El sistema puede hacerse más robusto y extenderse a deformaciones mayores.
- La malla puede hacerse más fina consiguiendo así mayor precisión.

6.2. Líneas futuras de investigación

El campo del SLAM no rígido es un terreno en el que queda mucho por investigar. Algunas líneas de investigación podrían ser:

- Aplicaciones médicas en las que los puntos de contorno sean diferenciables y se pase una imagen de referencia del área deformable.
- Segmentación automática de elementos deformables.
- Realidad aumentada en elementos deformables.
- Reconomiento de expresiones faciales a partir de imágenes.

Capítulo 7

Bibliografía

[1] Andrew J Davison. Real-time simultaneous localisation and mapping with a single camera. In *IEEE Int. Conf. Computer Vision*, page 1403. IEEE, 2003.

[2] Georg Klein and David Murray. Parallel tracking and mapping for small ar workspaces. In *Mixed and Augmented Reality, 2007. ISMAR 2007. 6th IEEE and ACM International Symposium on*, pages 225–234. IEEE, 2007.

[3] Javier Civera, Andrew J Davison, and JM Martinez Montiel. Inverse depth parametrization for monocular slam. *IEEE transactions on robotics*, 24(5):932–945, 2008.

[4] Raul Mur-Artal, Jose Maria Martinez Montiel, and Juan D Tardos. Orb-slam: a versatile and accurate monocular slam system. *IEEE Transactions on Robotics*, 31(5):1147–1163, 2015.

[5] Dat Tien Ngo, Jonas Östlund, and Pascal Fua. Template-based monocular 3d shape recovery using laplacian meshes. *IEEE transactions on pattern analysis and machine intelligence*, 38(1):172–187, 2016.

[6] Adrian Kaehler and Gary Bradski. *Learning OpenCV 3: Computer Vision in C++ with the OpenCV Library*. O'Reilly, 2016.

[7] Universidad Politécnica de Madrid. Triangulación de delaunay. `http://www.dma.fi.upm.es/recursos/aplicaciones/geometria_computacional_y_grafos/web/triangulaciones/delaunay.html`, 2017.

[8] Rainer Kümmerle, Giorgio Grisetti, Hauke Strasdat, Kurt Konolige, and Wolfram Burgard. g 2 o: A general framework for graph optimization. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 3607–3613. IEEE, 2011.

[9] Olga Sorkine. Laplacian mesh processing. In *Eurographics (STARs)*, pages 53–70, 2005.

Lista de Figuras

Lista de Tablas