



Universidad
Zaragoza

Trabajo Fin de Grado en Ingeniería Industrial

Relocalización deformable con restricciones laplacianas

Autor

LUIS MIGUEL PÉREZ MORENTE

Directores

JOSÉ MARÍA MARTÍNEZ MONTIEL

Escuela de Ingeniería y Arquitectura
2018



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./D^a. Luis Miguel Pérez Morente

con nº de DNI 17768460V 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)

Ingeniería de tecnologías industriales, (Título del Trabajo)

Relocalización deformable con restricciones laplacianas

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

Zaragoza, 22 de noviembre de 2018

Fdo: Luis Miguel Pérez Morente

Resumen

Sea un mapa de una escena. El problema de la relocalización, también conocido como “robot secuestrado”, consiste en la localización de una cámara respecto del mapa tomando como data una única imagen tomada por la cámara.

El objetivo es hacer un sistema de relocalización para un sistema VSLAM (Simultaneous Location and Mapping with Visual sensor). Nuestro sistema de VSLAM, llamado ORBSDTAM, procesa una secuencia de vídeo de una escena y obtiene la forma indeformada de la escena. También es resultado del ORBSDTAM una base de datos de keyframes con su correspondiente plantilla en reposo. Los keyframes son un subconjunto reducido de imágenes del vídeo, elegido con el criterio de cubrir la escena observada, habiendo solapamiento entre los distintos keyframes. Además, para cada keyframe, se dispone de una estimación de la superficie de la escena en su posición de reposo.

Nuestro programa toma una imagen de la escena y la base de datos de keyframes de la escena, a partir de ellos estima la posición de la cámara que observó la imagen respecto de la plantilla de uno de los keyframes. Además también estima la deformación que ha sufrido la plantilla cuando ha sido observada por la cámara. Asumimos que el ORBSDTAM nos da cual es el keyframe más similar a la imagen, y también nos da las correspondencias entre el keyframe y la imagen.

Se ha concebido y desarrollado un software para estimar la posición entre la cámara y la plantilla a partir de las correspondencias, siguiendo la propuesta de Ngo et al. [1]. A partir de las correspondencias y la plantilla en reposo se construye un sistema lineal de ecuaciones que combina el error de medida en las imágenes con un regularizador que penaliza la deformación de la plantilla, finalmente el sistema de ecuaciones se resuelve mediante SVD. El sistema ha sido validado experimentalmente con una secuencia de imágenes reales de una plantilla deformable, mostrando es capaz de relocalizar la cámara en un escenario con una deformación significativa para más del 90 % de los frames, con un tiempo de procesamiento por frame inferior a 300 ms.

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

Índice

1. Introducción y objetivos	3
1.1. Objetivos	3
1.2. Relocalización de la cámara a partir de una imagen	5
2. Relocalización mediante plantilla deformable	7
2.1. Observación de la plantilla por la cámara	7
2.2. Estimación de la plantilla deformada a partir de una imagen	8
2.3. Estimación de la plantilla deformada con regularizador laplaciano	9
2.3.1. Matriz de regularización	10
2.3.2. Sistema laplaciano	11
3. Validación experimental	13
3.1. Estimación de la deformación	14
3.2. Análisis de tiempos	15
4. Conclusiones	16
4.1. Conclusiones	16
5. Bibliografía	17
Lista de Figuras	18
Lista de Tablas	19

Capítulo 1

Introducción y objetivos

1.1. Objetivos

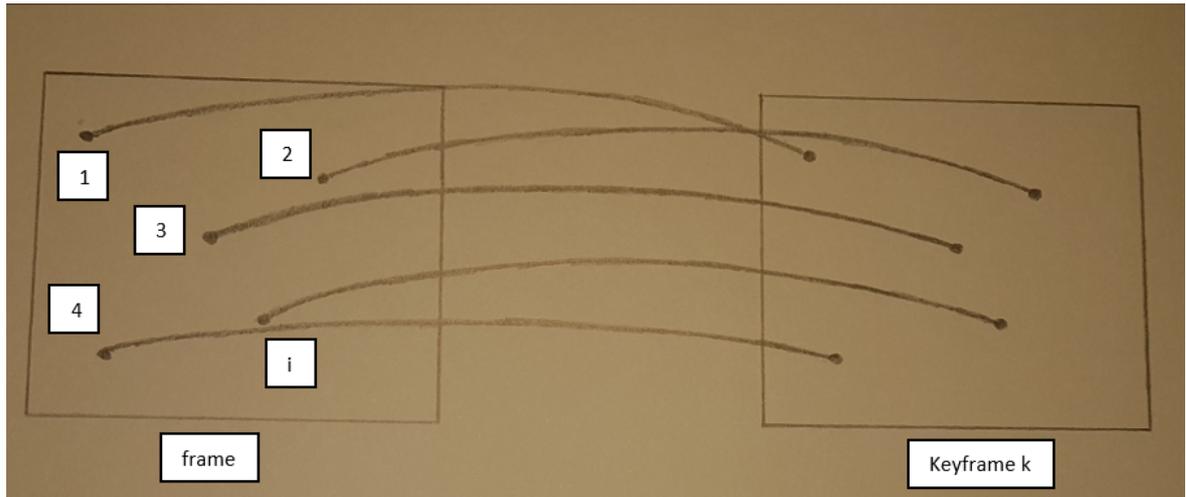
El problema a resolver a lo largo de este proyecto consiste en conseguir la relocalización de una cámara monocular dentro de un escenario deformable. Este problema consiste en la localización de una cámara a partir de una única imagen de la escena. El objetivo final es que el programa logre calcular la deformación a la que esta sometida la escena y la posición que ocupa la cámara con respecto a la misma.

Se empieza a partir del software ORBDTLAM [2]. En primer lugar se tiene la escena con la que se va a trabajar en una configuración indeformada, esta escena es recorrida por una cámara, durante este recorrido el programa guarda algunas imágenes como keyframes, estos keyframes serán utilizados más adelante para la relocalización de la cámara en la configuración deformada de la escena.

Para la escena completa en la configuración inicial indeformada se tiene una malla triangular que define la posición de cada uno de los puntos vistos con respecto a los nodos de dicha malla con sus coordenadas baricéntricas. Para esta malla se construye una matriz \mathbf{A} de regularización, la cual penalizará las transformaciones no rígidas de la malla. Para cada keyframe k se guarda una submatriz \mathbf{A}^k la cual solo contiene la parte de la matriz \mathbf{A} que contiene los nodos y las aristas vistos en dicho keyframe, la construcción de esta matriz se verá más adelante en el capítulo 2.3.1.

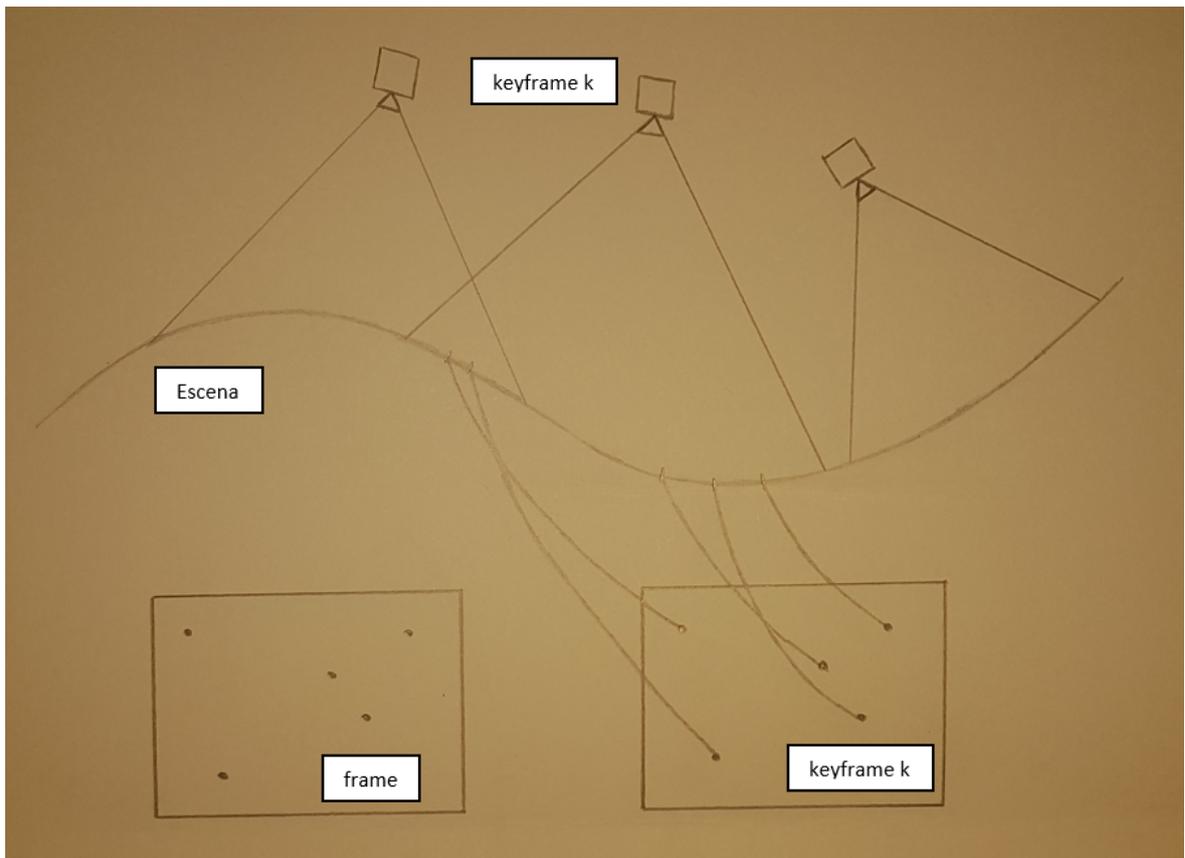
Para la relocalización con un frame i que observa una parte de la escena deformada se encuesta al sistema el cual devuelve el keyframe, k , más parecido a la imagen representada en dicho frame, esto se realiza mediante una codificación con bolsa de palabras binarias DBoW2 [3] que devuelve los matches putativos del frame con cada uno de los keyframes devolviendo el más parecido. Se emparejan algunos de los puntos que aparecen entre el frame y el keyframe como se ve en la figura 1.1, y esto los relaciona con el mapa como se ve en la figura 1.2.

Una vez se tienen los puntos emparejados entre el frame i y a través del keyframe



(a) Emparejamiento entre frame y keyframe

Figura 1.1: Emparejamiento entre puntos entre el frame en estudio y el keyframe devuelto



(a) Emparejamiento entre keyframe y escena

Figura 1.2: Emparejamiento entre i puntos entre el keyframe en estudio y la escena

k , con el mapa tridimensional se puede calcular la matriz \mathbf{H}_i^k . A partir de esta matriz \mathbf{H}_i^k y las coordenadas baricéntricas, las cuales se suponen invariantes a la deformación, es decir, las coordenadas baricéntricas serán las mismas en la configuración deformada

y en la indeformada, se tendrá la relación entre las coordenadas del punto en el frame y las coordenadas de los nodos de la malla entre los que se encuentra el punto, estas relaciones se recogerán en la matriz \mathbf{M}_i^k .

Una vez tenemos \mathbf{M}_i^k y \mathbf{A}^k se puede ensamblar la matriz $\mathbf{M}_{\omega_r}^k$, la cual se utilizara para resolver el problema mediante un sistema de ecuaciones lineales:

$$\mathbf{M}_{\omega_r}^k = \begin{bmatrix} \mathbf{M}_i^k \\ \omega_r \mathbf{A}^k \end{bmatrix} \quad (1.1)$$

A falta de definir el valor de ω el cual define la ponderación de las ecuaciones de \mathbf{A}^k con respecto a las de \mathbf{M}_i^k .

Aplicando un **SVD** a esta matriz $\mathbf{M}_{\omega_r}^k$, se obtendrán las coordenadas de los nodos de la malla la cual es visible en el keyframe k , y a partir de estas coordenadas se obtendría la posición de la cámara cuando se capturó dicho frame.

Durante este trabajo se va a utilizar el dataset [4] representada en la figura 1.3 de forma que el procesamiento sea mas rápido al obtener directamente los puntos emparejados sin necesidad de encuestar a la base de datos para obtener el keyframe más similar a cada imagen.

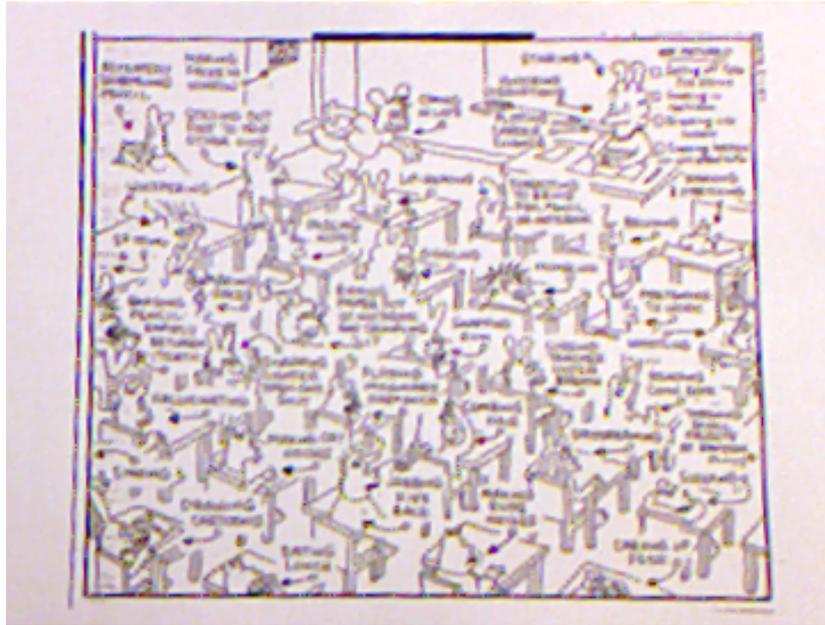
1.2. Relocalización de la cámara a partir de una imagen

El escenario deformable se captura inicialmente en una configuración indeformada y durante este proceso se capturan puntos pertenecientes a unas imágenes llamadas keyframes. Con estos puntos se genera una malla triangular respecto de la que estos puntos se localizan según sus coordenadas baricéntricas.

Para el calculo de la deformación el programa sera capaz de buscar entre los keyframes aquellos que mas se parezcan a la imagen de la configuración deformada en estudio, de estos keyframes seleccionados se sacaran los emparejamientos de puntos con la imagen. A través de la formulación lineal y la formulación laplaciana, al no ser la lineal suficiente para definir el problema, se calcularan las coordenadas de cada uno de los puntos emparejados.

En resumen, los procedimientos necesarios para conseguir la relocalización de la cámara y el calculo de la deformación son:

Construcción del mapa y obtención de los emparejamientos: Es proporcionada por el ORBSLAM, este software proporciona el mapa y los emparejamientos entre los keyframes y cada uno de los frames del vídeo de la configuración deformada.



(a) Escena

Figura 1.3: Escena con la que se va a trabajar

Construcción del sistema lineal: El primer paso para la resolución del problema es el planteamiento del sistema lineal que relaciona la posición de cada punto en el frame en estudio con su posición en el mapa en tres dimensiones, esto se detallara en el capítulo 2.1.

Regularización laplaciana: Como se explicara mas adelante la formulación lineal es insuficiente para resolver este problema incluso en el caso de tener muchos puntos emparejados entre el frame en estudio y el mapa en tres dimensiones.

Ensamblado sistema final y resolución sistema lineal Para la obtención de las coordenadas de los nodos de la malla se empleara un **SVD** que las devolverá según el orden que nosotros hayamos decidido.

Capítulo 2

Relocalización mediante plantilla deformable

En este capítulo se va explicar el algoritmo propuesto por [1]. Se reproduce la formulación de forma que la memoria sea auto-contenida.

La resolución del problema se va a dividir en la construcción del sistema a partir de cada uno de los subsistemas, tanto el lineal como las restricciones laplacianas, y la resolución del mismo mediante un **SVD**.

2.1. Observación de la plantilla por la cámara

Estas ecuaciones relacionan las coordenadas de los puntos en tres dimensiones, \mathbf{X}_j expresados en una referencia solidaria con la cámara, con las coordenadas de ese mismo punto en la imagen i en dos dimensiones capturada por la cámara.

$$\begin{bmatrix} u_{i,j} \\ v_{i,j} \\ 1 \end{bmatrix} = \mathbf{K}_i \mathbf{X}_j \quad (2.1)$$

donde \mathbf{K} es la matriz de calibración de la cámara. Siendo la matriz \mathbf{K}_i una matriz 3×3 definida a partir de los parámetros de calibración f_x, f_y, c_x, c_y . En este trabajo consideramos la misma matriz de calibración para todas las cámaras, por lo tanto esta matriz \mathbf{K}_i sera invariante.

Siendo la matriz $\mathbf{K}_{3 \times 3}$:

$$\mathbf{K}_{3 \times 3} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \quad (2.2)$$

Visto esto podemos escribir la ecuación que relaciona las coordenadas de un punto

en el frame y ese mismo punto en la malla tridimensional:

$$\mathbf{K}(b_1\mathbf{v}_{f,1} + b_2\mathbf{v}_{f,2} + b_3\mathbf{v}_{f,3}) = \mathbf{k} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \quad (2.3)$$

Donde \mathbf{k} es la componente homogénea.

2.2. Estimación de la plantilla deformada a partir de una imagen

Para la estimación de la plantilla a partir de una imagen se van a aplicar las ecuaciones presentadas en el apartado anterior.

Las coordenadas baricéntricas de las que se ha hablado a lo largo de la memoria de este trabajo son las coordenadas de un punto que pertenece a un triangulo a partir de las coordenadas de los vértices de dicho triangulo.

Siendo \mathbf{X}_i las coordenadas del punto que pertenece al triangulo, b_j sus coordenadas baricéntricas, y $\mathbf{V}_{i,j}$ las coordenadas de los vértices del triangulo, la posición del punto se expresa como:

$$\mathbf{X}_i = b_{j,1}\mathbf{V}_{j,1} + b_{j,2}\mathbf{V}_{j,2} + b_{j,3}\mathbf{V}_{j,3} \quad (2.4)$$

De esta forma el movimiento de la posición de un nodo provoca la variación en la posición de todos los puntos relacionados con este nodo.

Con el uso de la malla y las coordenadas baricéntricas, las cuales son conocidas, consigue reducirse el numero de incógnitas del problema sin modificar el numero de ecuaciones.

En el sistema lineal se obtienen dos ecuaciones por cada punto emparejado. Estas dos ecuaciones son las que se explican a continuación.

Una vez tenemos todos los datos arriba explicados se puede proceder a la construcción del sistema lineal. Como ya se ha comentado se tendrán dos ecuaciones por cada emparejamiento y tres incógnitas por cada nodo de la malla.

Por cada emparejamiento ente la imagen y la plantilla obtendremos:

$$\begin{bmatrix} b_1 f_x & 0 & b_1(c_x - u) & b_2 f_x & 0 & b_2(c_x - u) & b_2 f_x & 0 & b_3(c_x - u) \\ 0 & b_1 f_y & b_1(c_y - v) & 0 & b_2 f_y & b_2(c_y - v) & 0 & b_3 f_y & b_3(c_y - v) \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ z_1 \\ x_2 \\ y_2 \\ z_2 \\ x_3 \\ y_3 \\ z_3 \end{bmatrix} = 0 \quad (2.5)$$

Donde $b_{\bullet,i}$ son las coordenadas baricéntricas de dicho emparejamiento, $u_{i,j}$ y $v_{i,j}$ las coordenadas del punto en el frame i , f_x , f_y , c_x y c_y son los parámetros de calibración, y $x_{j,\bullet}$, $y_{j,\bullet}$ y $z_{j,\bullet}$ las coordenadas de los tres nodos de la malla respecto de los que esta referenciado el punto emparejado.

Con lo cual se obtiene un sistema de $2n$ ecuaciones siendo n el numero de puntos emparejados entre el mapa en tres dimensiones y el frame actual. Este sistema va a representarse por la matriz $\mathbf{M}_{2n \times 3m}$, siendo m el numero de nodos en la malla triangular:

$$\mathbf{M}\mathbf{x} = 0 \quad (2.6)$$

Donde \mathbf{x} son las coordenadas de los nodos de la malla $x_{j,\bullet}$, $y_{j,\bullet}$ y $z_{j,\bullet}$.

La solución de este sistema debería ser suficiente para obtener la superficie formada por los nodos de la malla en la configuración deformada, y por tanto mediante las coordenadas baricéntricas obtener la posición de cada uno de los puntos emparejados. Sin embargo, el rango de la matriz \mathbf{M} es insuficiente al ser al menos un tercio de los valores propios extremadamente pequeños.

Para solucionar esto se deben introducir restricciones que penalicen la deformación, estas se explicaran en el siguiente apartado.

2.3. Estimación de la plantilla deformada con regularizador laplaciano

Para solucionar el hecho de que la formulación lineal no es suficiente para resolver el problema se va a introducir la formulación laplaciana. Para ello vamos a suponer que la forma representada en figura 2.1a es la forma inicial de la malla en estudio donde \mathbf{X}_{ref} son las coordenadas de sus nodos.

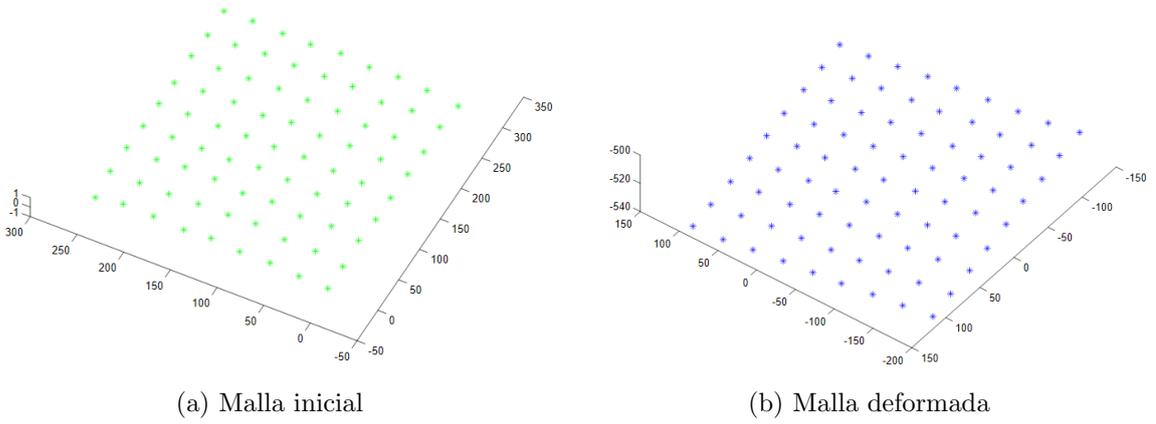


Figura 2.1: Malla regular en reposo y deformada

Se puede definir una matriz de regularización \mathbf{A} tal que $\|\mathbf{Ax}\|^2 = 0$ siendo \mathbf{x} el vector de coordenadas de la eq. 2.6 con $\mathbf{x}=\mathbf{X}_{\text{ref}}$ o siendo una transformación rígida de la misma. Es decir, $\|\mathbf{Ax}\|^2 = 0$ penaliza las transformaciones no rígidas de la malla.

El sistema presentado en la eq. 2.6 es aumentado con el termino regularizador $\|\mathbf{Ax}\|^2 = 0$ dando lugar a al sistema:

$$\min_x \|\mathbf{Mx}\|^2 + \omega_r \|\mathbf{Ax}\|^2, \text{st} \|x\| = 1 \quad (2.7)$$

Donde w_r es el peso que se le quiera dar al regularizador respecto del sistema lineal. Este sistema es equivalente al presentado a continuación:

$$\mathbf{M}_{\mathbf{w}_r} = \begin{bmatrix} \mathbf{M} \\ \omega_r \mathbf{A} \end{bmatrix} \quad (2.8)$$

Este sistema puede resolverse aplicando un **SVD** y y aplicando un factor de escala para obtener una malla con una escala geométrica similar a la de la malla inicial.

2.3.1. Matriz de regularización

Se va a ver como completar la matriz de regularización A de forma que se cumpla que $\|\mathbf{Ax}\|^2 = \|\mathbf{Ax}'\|^2$ siendo \mathbf{x}' una transformación rígida de \mathbf{x} . Se va a realizar en el caso de que la malla sea de forma plana.

Dada una malla triangular como la presentada en la figura 2.1a en su configuración indeformada, para cada par de superficies triangulares que compartan un una arista h como se muestra en la figura 2.2. siempre se pueden encontrar unos pesos $w_{\bullet,h}$ tales que:

$$\mathbf{0} = w_{1,h} \mathbf{V}_{1,h} + w_{2,h} \mathbf{V}_{2,h} + w_{3,h} \mathbf{V}_{3,h} + w_{4,h} \mathbf{V}_{4,h} \quad (2.9)$$

$$0 = w_{1,h} + w_{2,h} + w_{3,h} + w_{4,h} \quad (2.10)$$

$$1 = w_{1,h}^2 + w_{2,h}^2 + w_{3,h}^2 + w_{4,h}^2 \quad (2.11)$$

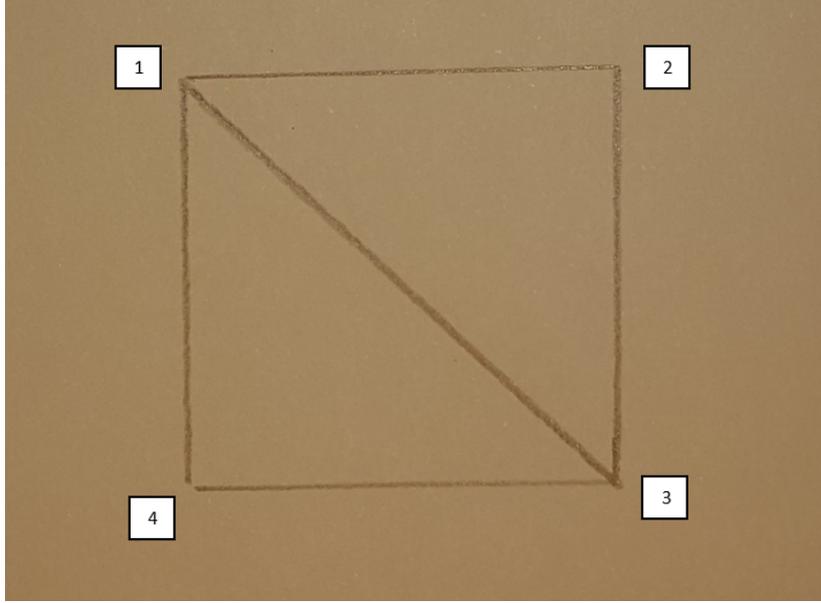


Figura 2.2: Pareja de triángulos de la malla que comparten una arista

Siendo \mathbf{v}_{iref} las coordenadas de cada uno de los cuatro nodos, y repitiéndose la eq. 2.9 para cada una de las tres coordenadas.

La primera ecuación es la encargada de que la matriz penalice las transformaciones no rígidas, es decir, fuerza a la malla a permanecer plana la segunda garantiza la invarianza, y la tercera previene que todos los pesos sean iguales a cero.

Para la construcción de la matriz \mathbf{A} una vez calculados los pesos situaremos cada uno en la columna correspondiente a su nodo y en la fila correspondiente a la arista que comparten. Una vez hecho esto se realiza un producto de Kronecker de la matriz \mathbf{A} por una matriz identidad 3×3 $I_{3 \times 3}$ para extenderla a las tres coordenadas.

2.3.2. Sistema laplaciano

Una vez se tienen los cuatro pesos que corresponden a la arista sobre la que se está trabajando se pueden colocar en el lugar indicado de la matriz \mathbf{A} .

Esta matriz proporciona una ecuación por cada arista que comparte dos triángulos de la malla, descartando las aristas exteriores de la malla, y por cada una de las tres coordenadas, y de la misma forma que en el sistema lineal, sus incógnitas son las coordenadas de los vértices de la malla $x_{j,\bullet}$, $y_{j,\bullet}$ y $z_{j,\bullet}$.

Por cada arista obtendremos:

$$[w_{1,h} \quad w_{2,h} \quad w_{3,h} \quad w_{4,h}] \begin{bmatrix} x_{1,h} \\ x_{2,h} \\ x_{3,h} \\ x_{4,h} \end{bmatrix} = 0 \quad (2.12)$$

Que extendida para cada una de las tres coordenadas espaciales quedara:

$$\begin{bmatrix} w_{1,h} & w_{2,h} & w_{3,h} & w_{4,h} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & w_{1,h} & w_{2,h} & w_{3,h} & w_{4,h} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & w_{1,h} & w_{2,h} & w_{3,h} & w_{4,h} \end{bmatrix} \begin{bmatrix} x_{1,h} \\ x_{2,h} \\ x_{3,h} \\ x_{4,h} \\ y_{1,h} \\ y_{2,h} \\ y_{3,h} \\ y_{4,h} \\ z_{1,h} \\ z_{2,h} \\ z_{3,h} \\ z_{4,h} \end{bmatrix} = \mathbf{0} \quad (2.13)$$

Se debe tener en cuenta que tanto en el caso del sistema laplaciano como en el caso del sistema lineal los términos de cada ecuación deben ordenarse de forma que al completar el sistema queden multiplicados por la coordenada correcta del nodo adecuado, es decir, tanto en la eq. 2.5 como en la eq. 2.13 habrá ceros entre las columnas representadas que corresponderían a las coordenadas del resto de nodos de la malla.

Se ha decidido que ambas ecuaciones estarán ordenadas según la posición de los nodos en la malla indeformada, por tanto la solución obtenida al resolver el sistema estará ordenada de esta misma manera.

En lo referente a cada una de las tres coordenadas del espacio, se ha decidido que irán agrupadas de forma que las primeras posiciones en el vector de solución \mathbf{X} estarán ocupadas de las $x_{j,\bullet}$ de los nodos de la malla en el orden arriba citado, seguidas por las $y_{j,\bullet}$ y finalizando con las $z_{j,\bullet}$.

Capítulo 3

Validación experimental

Para validar experimentalmente el proyecto se ha utilizado el vídeo del “Kinect paper Dataset [4] en el que la escena presentada se deforma y sobre el que se puede observar como la malla calculada es capaz de adaptarse a estos cambios.

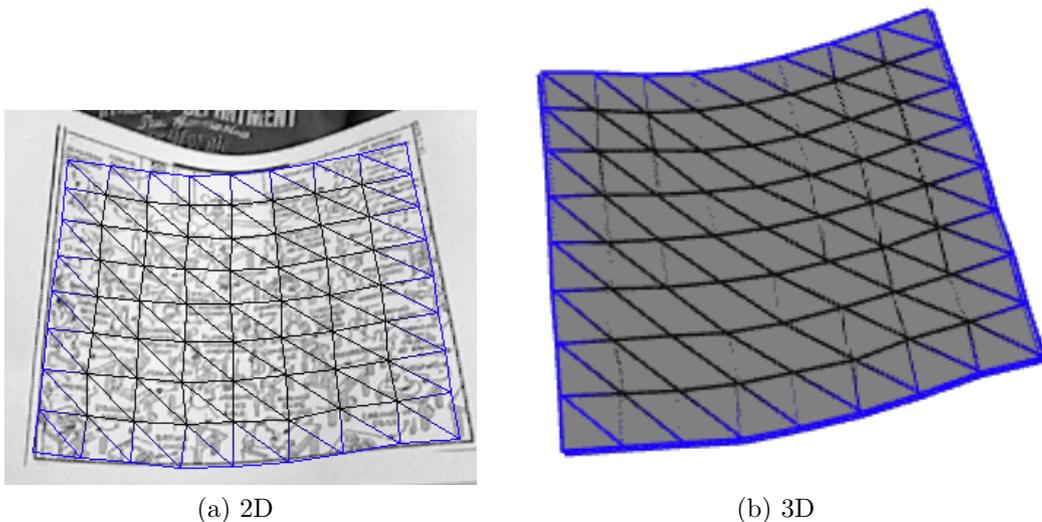


Figura 3.1: Representación de la malla 3D y la malla sobre la escena deformada

El experimento que se ha realizado ha consistido en procesar cada una de las 191 imágenes que componen el dataset, de las que se tienen los emparejamientos y la plantilla con la solución de referencia. Para cada imagen se obtiene la posición real de todos los puntos emparejados.

Para contrastar los resultados obtenidos se tiene una solución de referencia obtenida a partir de una cámara RGB-D la cual capta una imagen de profundidad, a partir de esta podemos obtener las tres coordenadas de cada uno de los puntos de la imagen.

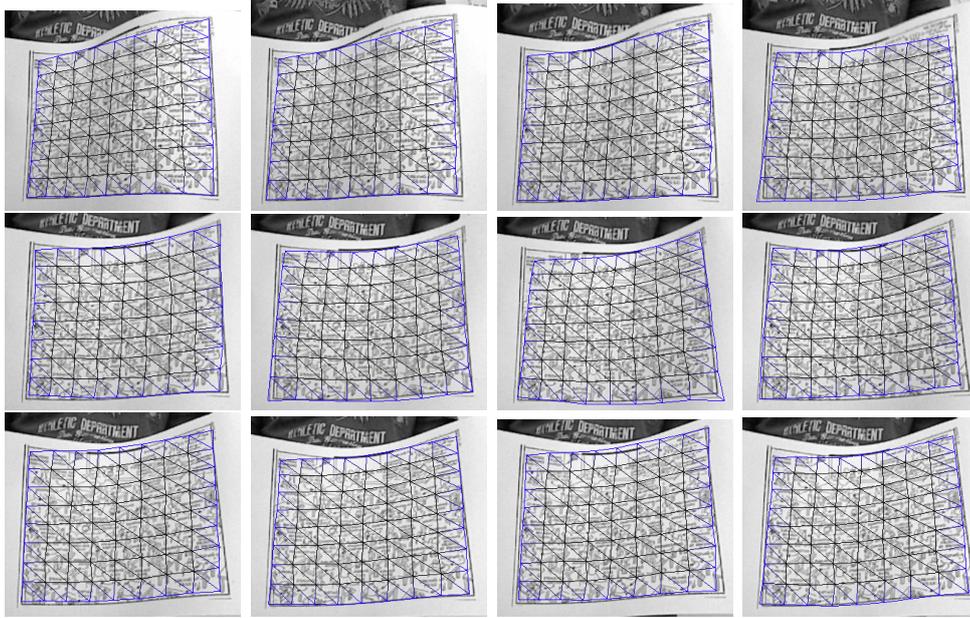


Figura 3.2: Secuencia de imágenes pertenecientes al dataset

3.1. Estimación de la deformación

El modelo calculado es capaz de relocalizarse en una escena completamente deformable como se vera a continuación en un elevado numero de casos, el parámetro que se va a utilizar para medir si la cámara ha sido capaz de relocalizarse o no sera el error **RMS** en tres dimensiones en cada uno de los frames.

Para el calculo del error se obtiene la diferencia entre la posición obtenida durante el procesamiento de cada uno de los puntos emparejados y la posición de referencia obtenida del dataset. Siendo el error de cada punto e_p siendo p el punto considerado. El error **RMS** se obtendrá como:

$$\mathbf{RMS} = \sqrt{e_1^2 + e_2^2 + e_3^2 + e_4^2 + \dots + e_p^2} \quad (3.1)$$

Se ha considerado que la cámara se ha relocalizado en una imagen si el error **RMS** es menor de 5cm en dicha imagen.

Dependiendo del valor dado a ω , es decir, del peso dado al regularizador laplaciano, la capacidad del sistema para relocalizarse cambia como se puede observar en la tabla 3.1:

Para un valor de ω de 24000 se alcanza un máximo de 180 frames relocalizados con respecto a los 191 que forman la secuencia de vídeo alcanzando así un 94,24% de éxito en la relocalización y manteniéndose constante para un rango amplio de valores de ω . Además, en la mitad de las imágenes el error no supera los 2.5cm.

Peso (ω)	Imágenes relocalizadas	%Relocalización
10000	147	77
14000	167	87.43
18000	177	92.67
24000	180	94.24
32000	180	94.24
34000	179	93.72

Tabla 3.1: Comparación de las imágenes relocalizadas en función del peso del regularizador

3.2. Análisis de tiempos

Para obtener la frecuencia a la que sería capaz de trabajar el sistema estima el tiempo de procesado de cada una de las etapas del algoritmos. Los tiempos que se van a obtener son los de ensamblado de cada uno de los sistemas de ecuaciones, y el de resolución del mismo mediante SVD. La tabla 3.2 detalla los tiempos promedio de cada una de las etapas.

Ensamblado (A)	Ensamblado (M)	Resolución del sistema	Total
0.826ms	2.423ms	254.82ms	258ms

Tabla 3.2: Tiempos medios de ejecución de cada una de las partes del sistema en ms

En estos tiempos hay que tener en cuenta que no se ha contabilizado la ejecución del ORBSDTAM para la obtención de los matches y los emparejamientos.

Hay que tener en cuenta además que la matriz \mathbf{A} solo se calcularía una vez, al depender de la distribución de los nodos en la configuración indeformada en reposo y ser esta la misma para todos los frames la matriz no cambia en toda la ejecución del problema.

Con la matriz \mathbf{M} sin embargo no pasa lo mismo ya que su tamaño y su contenido depende de los emparejamientos, esto también provoca que a mayor numero de emparejamientos también aumenta el tiempo de computo de dicha matriz.

En cuanto al tiempo de resolución se puede observar que también depende del numero de puntos emparejados aumentando a la vez que aumenta dicho número.

Por lo tanto el tiempo de ejecución total medio sería de 258 milisegundos.

Capítulo 4

Conclusiones

4.1. Conclusiones

Se ha probado como el algoritmo [1] es capaz de proporcionar una relocalización a partir de emparejamientos de forma rápida y robusta.

Una vez validada la aproximación, el próximo paso sería ver el comportamiento respecto de las correspondencias proporcionadas por el DBoW2 para keyframes de una secuencia procesada con ORBSDTAM, ya que los emparejamientos devueltos de esta manera no son tan precisos ni tan abundantes.

Otro paso a realizar sería el de mejorar de forma iterativa las correspondencias eliminando outliers y volviendo a procesar las imágenes.

Posible refinamiento por optimización no lineal

A largo plazo podría probarse este sistema con una imagen médica al ser este una de sus posibles aplicaciones más prometedoras.

En conclusión, para ser un primer paso en la relocalización en un entorno deformable los resultados son muy buenos pero habría que mejorar algunos ámbitos como es el rechazo de puntos espúreos, y la optimización no lineal del sistema pero es un buen primer paso en este ámbito.

Capítulo 5

Bibliografía

- [1] 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.
- [2] J. Lamarca and J.M.M. Montiel. Camera tracking for SLAM in deformable maps. In *4th Inter. Workshop on Recovering 6D Object Pose. ECCV*.
- [3] Dorian Gálvez-López and J. D. Tardós. Bags of binary words for fast place recognition in image sequences. *IEEE Transactions on Robotics*, 28(5):1188–1197, October 2012.
- [4] Aydin Varol, Mathieu Salzmann, Pascal Fua, and Raquel Urtasun. A constrained latent variable model. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2248–2255. Ieee, 2012.

Lista de Figuras

1.1. Emparejamiento entre puntos entre el frame en estudio y el keyframe devuelto	4
1.2. Emparejamiento entre i puntos entre el keyframe en estudio y la escena	4
1.3. Escena con la que se va a trabajar	6
2.1. Malla regular en reposo y deformada	10
2.2. Pareja de triángulos de la malla que comparten una arista	11
3.1. Representación de la malla 3D y la malla sobre la escena deformada . .	13
3.2. Secuencia de imágenes pertenecientes al dataset	14

Lista de Tablas

3.1. Comparación de las imágenes relocalizadas en función del peso del regularizador	15
3.2. Tiempos medios de ejecución de cada una de las partes del sistema en ms	15