

---

# Índice general

<b>Índice general</b>	<b>1</b>
<b>1. Conclusiones y trabajo a futuro</b>	<b>2</b>
1.1. Introducción . . . . .	2
1.2. Conclusiones generales . . . . .	2
1.3. Conclusiones específicas . . . . .	3
1.3.1. Detección de características . . . . .	3
1.3.2. Estimación de pose . . . . .	3
1.3.3. Una solución alternativa: SLAM . . . . .	4
1.3.4. Demás bloques . . . . .	4
1.4. Trabajo a futuro . . . . .	4
1.4.1. Sobre el servidor . . . . .	4
1.4.2. Sobre la realidad aumentada . . . . .	5
1.4.3. Sobre la plataforma . . . . .	5

---

# CAPÍTULO 1

---

## Conclusiones y trabajo a futuro

### 1.1. Introducción

En el presente capítulo se analiza al proyecto en su conjunto, primero desde el cumplimiento o no de sus objetivos y luego se hará una discusión técnica respecto de las elecciones realizadas, sobre todo en cuanto a los algoritmos utilizados para implementar cada uno de los bloques involucrados en la realidad aumentada. Finalmente, se plantea el trabajo que se podría hacer a futuro de manera de mejorar la aplicación final.

### 1.2. Conclusiones generales

En líneas generales, se puede decir que el presente proyecto resultó exitoso. En primer lugar, porque se considera que se cumplieron con todos los objetivos originales. Al día de hoy se cuenta con una documentación de toda la investigación realizada que brinda herramientas y un registro de la experiencia a todo aquel que busque continuar con esta línea de investigación. Se destaca la importancia de contar con implementaciones de algoritmos disponibles en código abierto, ya que implementar todos los algoritmos hubiera sido una tarea demasiado extensa y hubiera limitado la capacidad de resolver el problema completo. También es bueno resaltar el tiempo invertido en analizar el estado del arte de los algoritmos de detección de características, identificación y estimación de pose. Se logró un nivel de comprensión de los mismos que permitió incluso proponer modificaciones que lograron optimizar su performance y extender su aplicación a casos para los que no fueron previstos. Por otra parte, se logró implementar varios casos de uso que solucionan diferentes problemas en los que se puede aplicar realidad aumentada. Finalmente, se implementó el mencionado recorrido interactivo para el museo, que cuenta con los bloques de *navegación*, *reconocimiento de obra* y *realidad aumentada* que se explican en el Capítulo ??.

En segundo lugar, se destaca el hecho de haber logrado desarrollar una aplicación completa que funcione correctamente. Se cree que el desempeño de la misma es muy bueno, y en particular, si bien la realidad aumentada puede mejorar, es suficiente para la misma.

En cuanto al funcionamiento del grupo de trabajo, se cree que fue excelente. Se adquirió una dinámica de subgrupos de trabajo que resultaron muy eficientes. Se respetó mucho la división de tareas y el trabajo se mantuvo perfectamente equilibrado, con integrantes proactivos en todo momento.

Finalmente, la relación con el tutor fue muy buena. Se cree que fue un excelente orientador y que su compromiso con el proyecto fue notable.

### 1.3. Conclusiones específicas

Hay varios temas a discutir respecto de los bloques utilizados para el diseño de la realidad aumentada. Previo a tal discusión es bueno volver a detallar, tal y como se realizó en el Capítulo ??, el diagrama de bloques del proceso mediante el cual se logra la realidad aumentada, en la Figura 1.1.

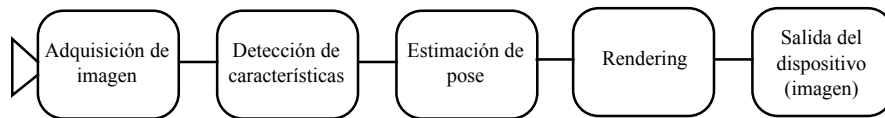


Figura 1.1: Diagrama de bloques del proceso mediante el cual se logra la realidad aumentada.

#### 1.3.1. Detección de características

El bloque de detección de catacterísticas se expande en la Figura 1.2.

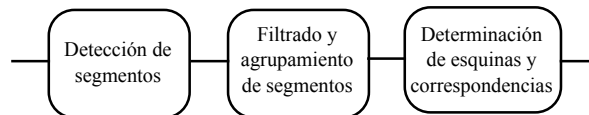


Figura 1.2: Diagrama de bloques de la detección de características utilizada en este proyecto.

Se vio en capítulos anteriores que la detección de segmentos se realizó utilizando LSD. Se está conforme tanto con el desempeño del algoritmo como con la optimización que se le hizo para tiempo real. Sin embargo, se cree que la implementación se puede optimizar aún más.

En lo que respecta a los algoritmos de filtrado y agrupación de segmentos, determinación de esquinas y correspondencias, se puede afirmar que funcionan correctamente. De cualquier manera, se cree que para ciertas aplicaciones en particular se podría utilizar un marcador más sencillo, con una geometría más simple de detectar; lo que simplificaría los algoritmos de detección y aumentaría la aplicabilidad a otros casos de uso.

#### 1.3.2. Estimación de pose

En el Capítulo ?? se explicaron en detalle diferentes versiones de POSIT, el algoritmo que este proyecto utiliza para estimar la pose de la cámara a partir de ciertas características de la imagen. En particular, se utilizó **el POSIT coplanar moderno** que toma como entrada ciertos puntos en un modelo predefinido y sus correspondientes en la imagen. Este es un algoritmo muy preciso bajo ciertas condiciones y veloz respecto de los algoritmos de estimación de pose tradicionales. Sin embargo, resulta no del todo lógico utilizar un algoritmo que tome puntos como entrada si las características extraídas de la imagen primeramente son segmentos.

Existe por su parte, otra versión de POSIT denominada “*Soft POSIT de líneas*”. Esta versión, toma como entrada líneas en una imagen y líneas en un modelo y tiene la habilidad de detectar correspondencias entre ellas y luego estimar la pose de la cámara a partir de dichas correspondencias.

De haberse usado esta versión de POSIT, el segundo y tercer bloque del diagrama de la Figura 1.2 no hubieran sido necesarios.

Sin embargo, *Soft POSIT* de líneas no funciona para marcadores planos como los utilizados en este proyecto. Además, requiere de una estimación de la pose inicial del dispositivo lo que no hace tan trivial su uso. Por estos motivos se optó por utilizar POSIT coplanar y resolver las correspondencias en la etapa de filtrado de segmentos.

### 1.3.3. Una solución alternativa: SLAM

“*Simultaneous Localization And Mapping*” (SLAM), o en español “Localización y Mapeado Simultáneos”, es el problema de ubicar a un dispositivo de visión artificial en un determinado lugar desconocido y que este, de manera incremental, vaya construyendo un modelo del lugar y a la vez vaya ubicándose dentro del modelo.

Este problema no sólo tiene solución, sino que además muchos de los kits de desarrollo de realidad aumentada, como por ejemplo *Metaio*, presentado en el Capítulo ??, lo utilizan. Y por lo que se pudo ver, da muy buenos resultados. En particular, logra resultados sustancialmente mejores a los de este proyecto, en cuanto a los tiempos de procesamiento.

De esta manera, los bloques “Detección de características” y “Estimación de pose” del diagrama de la Figura 1.1, pueden ser sustituidos por un algoritmo que solucione este problema.

### 1.3.4. Demás bloques

El desempeño de los demás bloques involucrados en el proceso mediante el cual se logra la realidad aumentada no requiere de un análisis extra al realizado en capítulos anteriores. ISGL3D, la herramienta utilizada para realizar renders sobre las imágenes capturadas, funciona perfectamente y los dispositivos de captura y despliegue de imágenes del *iPad* son muy buenos. Hubiera sido mejor capturar imágenes de mayor tamaño, para así lograr una mejor calidad a la salida. Pero cuanto mayor es el tamaño de la imagen capturada, también lo es el tiempo de procesamiento de los algoritmos de detección, en particular LSD.

## 1.4. Trabajo a futuro

### 1.4.1. Sobre el servidor

Se cree que la implementación del servidor en la aplicación final estuvo bien lograda, aunque no fue la más ortodoxa. Es poco escalable básicamente por dos razones. En primer lugar, no existe una interfaz de usuario para agregar o quitar obras de la base de datos del servidor, o incluso editar la información asociada a las mismas como por ejemplo imagen, audioguía, descripción, etc. Por eso, cualquier tipo de cambio en el servidor debe ser realizado por un técnico conocedor del sistema en su conjunto. En segundo lugar, al día de hoy toda la información referente a la realidad aumentada y al modelo del marcador es almacenada en la aplicación y por lo tanto, de agregarse algún cuadro a la base de datos, también se requeriría de un técnico conocedor del sistema para agregar esta información a la aplicación.

Un trabajo a futuro importante es, en primer lugar, implementar de manera prolija tanto la interacción entre la aplicación y el servidor, como la forma de almacenar los datos en este. En segundo lugar, realizar la aplicación de usuario que sirva para agregar o quitar información de la base de datos y finalmente, independizar completamente a la aplicación de los cuadros almacenados en la

base de datos.

Actualmente, se esta trabajando en coordinación con un proyecto de fin de estudios en la carrera Tecnólogo Informático de Paysandú, que tiene como objetivo realizar los cambios descritos en el párrafo anterior.

### **1.4.2. Sobre la realidad aumentada**

Lo expuesto en la Sección 1.3 sugiere, en primer lugar, probar diferentes marcadores y hacer un estudio detallado del desempeño de la realidad aumentada con cada uno de ellos para diferentes aplicaciones particulares.

En segundo lugar, un desafío a futuro prometedor es investigar las herramientas que resuelven SLAM y buscar la forma de integrarlas al bloque de realidad aumentada. Se cree que de esta manera, puede llegar a mejorar mucho su desempeño, tanto desde el punto de vista de la precisión como en los tiempos de ejecución.

De forma alternativa, se puede profundizar en la integración de los sensores inerciales del dispositivo a la estimación de la pose del mismo. Estos, mediante un bajo costo computacional, estiman de forma muy precisa cambios en su orientación lo que puede llevar bajar el *frame rate* obtenido en la aplicación.

### **1.4.3. Sobre la plataforma**

Como trabajo a futuro también se considera la migración de todo lo desarrollado, a dispositivos que funcionen con *Android* como sistema operativo. Dado que gran parte del desarrollo fue en lenguaje C, en particular todos los algoritmos de procesamiento de imágenes, es fácilmente portable a Android, que si bien se basa en Java, cuenta con una herramienta llamada NDK que permite reutilizar código de C/C++. Una motivación importante para considerar esta migración a Android es que estos dispositivos son de uso más masivo que los dispositivos *Apple* y en algunos casos tienen prestaciones similares y con precios más accesibles.