
Índice general

Índice general	1
1. Alcance del proyecto	2
1.1. Introducción	2
1.2. Objetivos del proyecto	2
1.3. Estado del arte	3
1.4. Explicación global de la aplicación	4
1.4.1. Navegación	4
1.4.2. Identificación de obras	4
1.4.3. Realidad Aumentada	5
1.5. Sobre el documento	5
Bibliografía	7

CAPÍTULO 1

Alcance del proyecto

1.1. Introducción

En el presente capítulo se habla del alcance del proyecto. Esto es, se plantean los objetivos tal y como fueron formulados originalmente y luego se los clasifica en las tres partes fundamentales del proyecto: *investigación*, *implementación* y *aplicación*. Estas son ponderadas en función de la importancia que tienen en el mismo, así como las dedicación total que se les dió. Luego, se resume la estructura de lo que terminó siendo la aplicación final, de manera de poder comprender a todo el proyecto en su conjunto. Finalmente, se comenta en qué parte de la documentación se detalla cada parte de dicha estructura.

1.2. Objetivos del proyecto

El presente proyecto de fin de carrera cuenta con varios objetivos en conjunto. Por un lado, se busca investigar distintos algoritmos de procesamiento de imágenes con el fin de implementarlos y estudiar su desempeño sobre ciertas plataformas móviles; para lo cual también se deben estudiar dichas plataformas y elegir la más apta de entre las opciones *Android* e *iOS*. Por otro lado, más adelante en el proyecto, se espera utilizar lo investigado para lograr una aplicación de realidad aumentada completa funcionando sobre un dispositivo móvil y en tiempo real. Finalmente se le quiere dar, a la aplicación de realidad aumentada, un marco dentro de un *recorrido interactivo en realidad aumentada* para museos.

Los objetivos anteriores pueden resumirse en las tres partes fundamentales del proyecto, que se expresan a continuación:

1. **Investigación:** comprensión de la arquitectura de las plataformas móviles y de sus plataformas de desarrollo, con el objetivo de embeber los distintos algoritmos y *software* en general en las mismas. Estudio de las diferentes maneras de lograr la realidad aumentada, elección de los algoritmos a utilizar, su comprensión y en algunos casos su implementación total o parcial. Aprendizaje de herramientas en general.
2. **Implementación:** integración de los distintos bloques para lograr la realidad aumentada. Implementación de bloques lógicos accesorios que faciliten la integración de los primeros. Validación de los algoritmos utilizados y desarrollados.

3. **Aplicación:** implementación de una aplicación total en la que el usuario ingrese al museo, se ubique dentro de él, se dirija a un cuadro, reciba información respecto del mismo y finalmente disfrute de la realidad aumentada sobre la obra.

Cada una de ellas se jerarquizó en función de la importancia que se que cree tiene para el proyecto, así como también el tiempo que se les dedicó:

Frente de trabajo	Porcentaje
Investigación	50 %
Implementación	30 %
Aplicación	20 %

Es importante aclarar que en ocasiones el límite entre los tres frentes de trabajo es difuso.

1.3. Estado del arte

Para llevar a cabo los objetivos planteados es bueno tener un contexto del estado del arte en cuanto al desarrollo de aplicaciones de realidad aumentada. Existen en la actualidad múltiples entornos de desarrollo comerciales para aplicaciones de realidad aumentada, en los que de manera sencilla, se logran este tipo de aplicaciones con desempeños verdaderamente muy buenos. Tal es el caso de *Metaio* [56], *Vuforia* [57], *String* [58] y *Aurasma* [60]. Por su parte, *Layar* [59], es también un kit de desarrollo para aplicaciones de realidad aumentada, pero se especializa en el agregado de contenido digital sólo sobre páginas impresas como revistas y catálogos. Ninguna de las aplicaciones anteriores es gratuita y ni siquiera en código abierto. En la figura 1.1 se muestra un ejemplo que incluye por defecto la herramienta *Metaio* y otro que incluye la herramienta *String*.

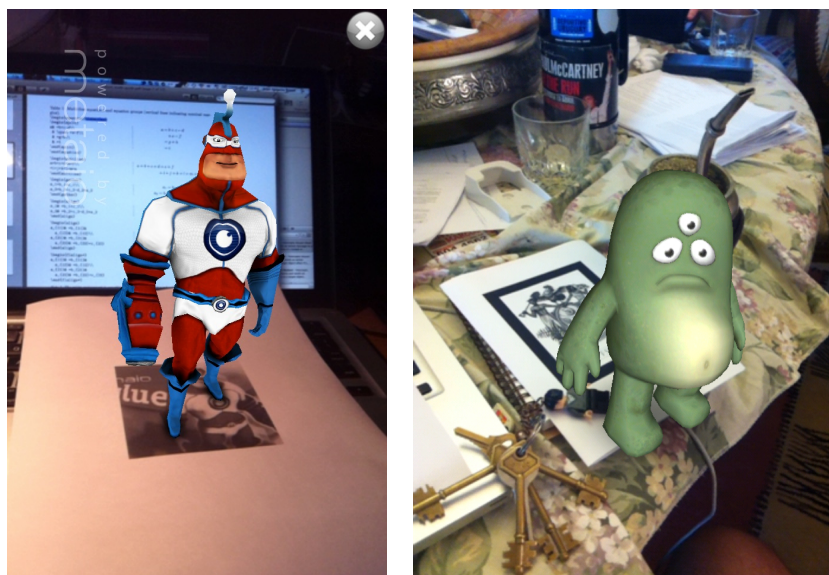


Figura 1.1: Izq.: Ejemplo de realidad aumentada que incluye por defecto el entorno de desarrollo *Metaio*. Der.: Otro ejemplo de realidad aumentada incluido por *String*.

Si bien se probaron algunas de estas herramientas y se tomaron de ellas estándares en cuanto a la *performance* que una aplicación de realidad aumentada debe tener, **no es un objetivo** del proyecto el desarrollo de aplicaciones de realidad aumentada con entornos de desarrollo ya existentes. Se hace hincapié en la investigación de herramientas y adquisición de experiencia propia, con el objetivo de marcar una hoja de ruta para todo aquí que desee continuar la investigación con nuevas ideas y algoritmos. Se deja a disposición de cualquier interesado el código de la aplicación final, los algoritmos implementados o editados, así como toda la documentación.

1.4. Explicación global de la aplicación

Si bien se dijo que la creación de la aplicación integral, correspondiente a un recorrido en realidad aumentada para muros, corresponde tan sólo a un quinto del alcance total del proyecto; la visualización de la aplicación total es quizá la forma más sencilla de comprender el proyecto en su conjunto.

Esta se desglosa en tres grandes bloques:

- **Navegación**
- **Identificación de obras**
- **Realidad aumentada**

que son resumidos individualmente en secciones subsiguientes.

1.4.1. Navegación

La navegación es la ubicación del usuario dentro del museo, útil tanto para el usuario como para la aplicación, ya que sabiendo en qué región del museo este se encuentra, se simplifica un poco la identificación de la obra. Se estudiaron distintas alternativas para la navegación. La primera posibilidad analizada fue la utilización de tres o más *access points*, mediante los cuales, una vez mapeadas las características de las señales en cada uno de los puntos de las salas, se puede ubicar al usuario dentro de las mismas. Otra forma de navegación que se tuvo en cuenta fue la localización a través de la tecnología GPS. Sin embargo, se optó por utilizar códigos QR dada su amplia difusión, practicidad y facilidad de implementación. Esta discusión técnica se ve más en detalle en el Capítulo ??.

1.4.2. Identificación de obras

Por identificación de obras se entiende al proceso mediante el cual la aplicación detecta frente a qué obra se encuentra el usuario para así entonces brindarle información de la misma, una audio-guía y si fuera el caso la posibilidad de desplegar realidad aumentada sobre ella. La forma en la que se implementó este bloque fue mediante un algoritmo de detección de características de imágenes llamado SIFT. Este algoritmo genera descriptores que sirven como identificadores de las imágenes y se verá en profundidad en el Capítulo ??.

1.4.3. Realidad Aumentada

El proceso mediante el cual se logra la realidad aumentada puede verse en el diagrama de bloques de la Figura 1.2. Resulta muy importante aclarar que cada uno de estos bloques es perfectamente intercambiable por otro de idéntica función, tan sólo ajustando las interfaces entre ellos.

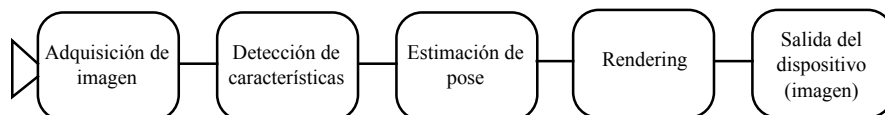


Figura 1.2: Diagrama de bloques del proceso mediante el cual se logra la realidad aumentada.

En el diagrama de bloques de la Figura 1.2, primero la cámara toma una imagen, que luego es procesada con el objetivo de detectar en esta características. Estas características pueden ser segmentos, esquinas, descriptores; que luego son utilizados por un algún algoritmo de estimación de la pose, que busca estimar en qué posición se encuentra la cámara respecto de cierto eje de coordenadas previamente definido y hacia dónde esta apunta. Con la información anterior, se debe poder *renderizar* una escena de manera consistente con la pose de la cámara, para así entonces lograr la salida del sistema, que será una imagen con la información digital añadida.

Para este proyecto se diseñó cierto marcador formado por tres grupos de cuadrados concéntricos, a partir del cual se extraen las características necesarias para la estimación de la pose del dispositivo. Además, el bloque de detección de características está compuesto por los bloques del diagrama de la Figura 1.3.

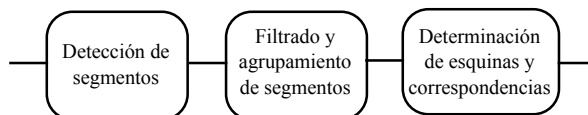


Figura 1.3: Diagrama de bloques de la detección de características utilizada en este proyecto.

En el diagrama de bloques de la figura 1.3, primero se detectan idealmente todos los segmentos que hay en la imagen y luego mediante cierto algoritmo se filtran tan sólo los pertenecientes al marcador antedicho y se agrupan por cuadrado. En el tercer bloque, se hallan las esquinas de estos cuadrados mediante la intersección de los segmentos. Finalmente, mediante cierta lógica, estos puntos son ordenados de una manera predefinida. Detalles respecto de este proceso y sobre el marcador utilizado se ven en el Capítulo ??.

1.5. Sobre el documento

En el presente documento se verán en detalle todos los aspectos técnicos referidos a los temas vistos en este capítulo. Se explica claramente la investigación realizada, los algoritmos utilizados, las herramientas aprendidas y en general, toda la experiencia adquirida a lo largo del proyecto. Se justifican claramente todas las decisiones que hubo que tomar y se comentará además, en cada caso, si estas fueron acertadas o no.

En el Capítulo ?? se comparan ambas plataformas y se justifica la elección de una de ellas. Luego se ven más a fondo algunas características de la plataforma seleccionada y finalmente algunas herramientas genéricas requeridas para el desarrollo. En el Capítulo ?? se ven en detalle las soluciones técnicas para la navegación y la detección de la obra. Los Capítulos ?? y ?? abarcan respectivamente los temas detección de características de las imágenes en general; y detalles respecto de marcadores comunmente utilizados en aplicaciones de realidad aumentada, siendo una adaptación de estos, el marcador utilizado en este proyecto en particular.

Más adelante, en el Capítulo ??, se estudia a fondo un algoritmo de detección de segmentos en imágenes digitales llamado LSD, utilizado para la detección de segmentos representada en el primer bloque de la Figura 1.3. Luego, en el Capítulo ??, se plantea un modelo para la cámara utilizada para la captura y se introducen conceptos básicos para comprender cómo es el proceso de la estimación de la pose de la misma. El algoritmo utilizado en este proyecto para tal fin, se presenta en el Capítulo ??.

La etapa de análisis de desempeño y validación de los algoritmos, en particular los asociados a la detección de características y estimación de pose, se da en el Capítulo ??. En este, se presenta la necesidad de realizar un filtrado tipo *Kalman* sobre la estimación de pose. En el capítulo ?? se presenta el algoritmo de *Kalman* y sus ecuaciones básicas. Finalmente, se realiza un nuevo análisis de desempeño con este nuevo agregado.

En el Capítulo ?? se introduce brevemente el concepto de *render* y se presenta la herramienta utilizada para realizar *renders* en la plataforma escogida. Luego, en el Capítulo ??, se presentan los distintos casos de uso implementados para probar integrar todo los bloques necesarios para la realidad aumentada en pequeñas aplicaciones individuales. Finalmente, en el Capítulo ??, se explican los detalles técnicos respecto de la aplicación de la aplicación completa, tal y como se define en la sección 1.4.

Bibliografía

- [1] A. Etemadi. Robust segmentation of edge data. In *Proceedings of the 4th international conference on Image Processing and its applications*, 1992.
- [2] Y. I. Abdel-Aziz and H. M. Karara. Direct linear transformation from comparator coordinates into object space coordinates in close-range photogrammetry. In *Proceedings of the Symposium on Close-Range photogrammetry*, volume 1, pages 1–18, 1971.
- [3] C. Avellone and G. Capdehourat. Posicionamiento indoor con señales wifi. 2010.
- [4] Jean-Yves Bouguet. Camera calibration toolbox for matlab. http://www.vision.caltech.edu/bouguetj/calib_doc/, November 2012.
- [5] J. Brian Burns, Allen R. Hanson, and Edward M. Riseman. Extracting straight lines. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8:425–455, 1986.
- [6] F. John Canny. A Computational Approach to Edge Detection. 8(6):679–698, 1986.
- [7] Stuart Caunt. Isgl3d homepage. <http://www.isgl3d.com>, November 2012.
- [8] Philip David, Daniel Dementhon, Ramani Duraiswami, and Hanan Samet. Simultaneous pose and correspondence determination using line features. pages 424–431, 2003.
- [9] Philip David, Daniel DeMenthon, Ramani Duraiswami, and Hanan Samet. Simultaneous pose and correspondence determination using line feature. In *CVPR (2)*, pages 424–431, 2003.
- [10] Daniel F. DeMenthon and Larry S. Davis. Model-based object pose in 25 lines of code. *International Journal of Computer Vision*, 15:123–141, 1995.
- [11] Daniel DeMenthon Denis Oberkampf. Posit for coplanar points. http://www.cfar.umd.edu/~daniel/Site_2/Code.html, 1996 - 2004.
- [12] Agnès Desolneux, Lionel Moisan, and Jean-Michel Morel. Meaningful alignments. *International Journal of Computer Vision*, 40(1):7–23, 2000.
- [13] A. Etemadi, J-P. Schmidt, G. Matas, J. Illingworth, and J. Kittler. Low-level grouping of straight-line segments. In Peter Mowforth, editor, *Processings of the British Machine Vision Conference*. Springer-Verlag, 1991.
- [14] Mark Fiala. Artag revision 1, a fiducial marker system using digital techniques. <http://www.artag.net/>, November 2004.
- [15] Martin A. Fischler and Robert C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, June 1981.

- [16] B. Furht. *The Handbook of Augmented Reality*. 2011.
- [17] Martin Giupponi. Kalman robusto aplicado en segmentación de videos con texturas dinámicas. Tratamiento Estadístico de Señales. Facultad de Ingeniería, Universidad de la República, 2009.
- [18] Rafael Grompone von Gioi, Jérémie Jakubowicz, J.-M. Morel, and Gregory Randall. Lsd: a line segment detector. *Image Processing Online*, mar 2012.
- [19] Rafael Grompone von Gioi, Jérémie Jakubowicz, J.-M. Morel, and Gregory Randall. On straight line segment detection. *Journal of Mathematical Imaging and Vision*, 2008.
- [20] Robert M. Haralick, Chung-Nan Lee, Karsten Ottenberg, and Michael Nölle. Review and analysis of solutions of the three point perspective pose estimation problem. *Int. J. Comput. Vision*, 13(3):331–356, December 1994.
- [21] Chris Harris and Mike Stephens. A combined corner and edge detector. In *In Proc. of Fourth Alvey Vision Conference*, pages 147–151, 1988.
- [22] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004.
- [23] Monson H. Hayes. *Statistical Digital Signal Processing and Modeling*. John Wiley & Sons, Inc., New York, NY, USA, 1st edition, 1996.
- [24] Jane Heikkilä and Olli Silvén. A four-step camera calibration procedure with implicit image correction. In *1997 Conference on Computer Vision and Pattern Recognition (CVPR 97), June 17-19, 1997, San Juan, Puerto Rico*, page 1106. IEEE Computer Society, 1997.
- [25] Martin Hirzner. Marker detection for augmented reality applications. October 2008.
- [26] Dr. Hirokazu Kato. Artoolkit: a software library for building augmented reality (ar) applications. <http://www.hitl.washington.edu/artoolkit/>.
- [27] V. Lepetit and P. Fua. Monocular model-based 3d tracking of rigid objects: A survey. *Foundations and Trends in Computer Graphics and Vision*, 1(1):1–89, 2005.
- [28] David G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110, November 2004.
- [29] David G. Lowe. Object recognition from local scale-invariant features. In *Proceedings of the International Conference on Computer Vision-Volume 2 - Volume 2, ICCV '99*, pages 1150–, Washington, DC, USA, 1999. IEEE Computer Society.
- [30] Denis Oberkampf, Daniel F. DeMenthon, and Larry S. Davis. Iterative pose estimation using coplanar feature points. *Comput. Vis. Image Underst.*, 63(3):495–511, may 1996.
- [31] J. García Ocón. Autocalibración y sincronización de múltiples cámaras plz. 2007.
- [32] Matias Tailanian and Santiago Paternain. Autoposicionamiento 3d. <http://sites.google.com/site/autoposicionamiento3d/>, Julio 2011.
- [33] R.Y. Tsai. An efficient and accurate camera calibration technique for 3d machine vision. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 364–374, 1986.

- [34] Daniel Wagner and Dieter Schmalstieg. Artoolkitplus for pose tracking on mobile devices. 2007.
- [35] Zhengyou Zhang. Flexible camera calibration by viewing a plane from unknown orientations. In *ICCV*, pages 666–673, 1999.
- [36] Helmut Zollner and Robert Sablatnig. Comparison of methods for geometric camera calibration using planar calibration targets. In W. Burger and J. Scharinger, editors, *Digital Imaging in Media and Education, Proc. of the 28th Workshop of the Austrian Association for Pattern Recognition (OAGM/AAPR)*, volume 179, pages 237–244. Schriftenreihe der OCG, 2004.
- [37] Blender homepage. <http://www.blender.org/>.
- [38] Blender 2.6 python manual. <http://wiki.blender.org/index.php/Doc:2.6/Manual/Extensions/Python>.
- [39] Image processing on line - lsd: a line segment detector. <http://www.ipol.im/pub/art/2012/gjmr-lsd/>, nov 2012.
- [40] Vlfeat homepage. <http://www.vlfeat.org>, nov 2012.
- [41] iphone 4 support specification. <http://support.apple.com/kb/SP587>, nov 2012.
- [42] iphone 4s support specification. <http://support.apple.com/kb/SP643>, nov 2012.
- [43] ipod touch support specification. <http://support.apple.com/kb/SP594>, nov 2012.
- [44] ipad 2 support specification. <http://support.apple.com/kb/SP622>, nov 2012.
- [45] Apple platform notes. http://developer.apple.com/library/ios/#documentation/3DDrawing/Conceptual/OpenGL_ES_ProgrammingGuide/OpenGL_ES_Platforms/OpenGL_ES_Platforms.html, nov 2012.
- [46] Peter Thoman. *Microcontroller and System-on-a-chip*. University of Innsbruck, second edition, 2009.
- [47] Arm cortex-a processors. <http://www.arm.com/products/processors/cortex-a/index.php>, nov 2012.
- [48] Gpu benchmark. <http://www.anandtech.com/show/4216/apple-ipad-2-gpu-performance-explored-powervr-sgx543mp2-benchmarked>, nov 2012.
- [49] Powervr graphics technology. <http://www.imgtec.com/powervr/powervr-graphics-technology.asp>, nov 2012.
- [50] The objective-c programming language. <http://developer.apple.com/library/mac/#documentation/Cocoa/Conceptual/ObjectiveC/Introduction/introObjectiveC.html>, nov 2012.
- [51] ios technology overview. http://developer.apple.com/library/ios/#documentation/Miscellaneous/Conceptual/iPhoneOSTechOverview/Introduction/Introduction.html#//apple_ref/doc/uid/TP40007898-CH1-SW1, nov 2012.

- [52] Av foundation programming guide. http://developer.apple.com/library/ios/#DOCUMENTATION/AudioVideo/Conceptual/AVFoundationPG/Articles/00_Introduction.html#//apple_ref/doc/uid/TP40010188, nov 2012.
- [53] Advanced memory management programming guide. http://developer.apple.com/library/ios/#documentation/Cocoa/Conceptual/MemoryMgmt/Articles/MemoryMgmt.html#//apple_ref/doc/uid/10000011i, nov 2012.
- [54] Transitioning to arc release notes. http://developer.apple.com/library/ios/#releasenotes/ObjectiveC/RN-TransitioningToARC/Introduction/Introduction.html#//apple_ref/doc/uid/TP40011226, nov 2012.
- [55] Instruments user guide. https://developer.apple.com/library/ios/#documentation/DeveloperTools/Conceptual/InstrumentsUserGuide/Introduction/Introduction.html#//apple_ref/doc/uid/TP40004652, nov 2012.
- [56] Metaio homepage. <http://www.metaio.com/>, 2012.
- [57] Vuforia homepage. <http://www.qualcomm.com/solutions/augmented-reality>, 2012.
- [58] String homepage. <http://www.poweredbystring.com/>, 2012.
- [59] Layar homepage. <http://www.layar.com/>, 2012.
- [60] Aurasma homepage. <http://www.aurasma.com/>, 2012.