
Análisis y Desarrollo de Sistemas de Realidad Aumentada



UNIVERSIDAD COMPLUTENSE
MADRID

PROYECTO FIN DE MÁSTER EN SISTEMAS INTELIGENTES

Autor: **Héctor López Pombo**

Director: Antonio Navarro Martín

Máster en Investigación en Informática
Facultad de Informática
Universidad Complutense de Madrid

Curso académico: 2009/2010

ANÁLISIS Y DESARROLLO DE SISTEMAS DE REALIDAD AUMENTADA

PROYECTO FIN DE MÁSTER EN SISTEMAS INTELIGENTES

AUTOR: HÉCTOR LÓPEZ POMBO

DIRECTOR: ANTONIO NAVARRO MARTÍN

**Máster en Investigación en Informática
Facultad de Informática
Universidad Complutense de Madrid**

CURSO ACADÉMICO: 2009/2010

El abajo firmante, matriculado en el Máster en Investigación en Informática de la Facultad de Informática, autoriza a la Universidad Complutense de Madrid (UCM) a difundir y utilizar con fines académicos, no comerciales y mencionando expresamente a su autor el presente Trabajo Fin de Máster: “Análisis y Desarrollo de Sistemas de Realidad Aumentada”, realizado durante el curso académico 20XX-20XX bajo la dirección de Antonio Navarro Martín en el Departamento de Ingeniería del Software e Inteligencia Artificial, y a la Biblioteca de la UCM a depositarlo en el Archivo Institucional E-Prints Complutense con el objeto de incrementar la difusión, uso e impacto del trabajo en Internet y garantizar su preservación y acceso a largo plazo.

Madrid, a 13 de Septiembre de 2010-09-02

Héctor López Pombo

A mi hermano David, que con tu matrimonio seas muy feliz

Abstract

In the past, virtual reality has been a technology affordable only to a restricted set of users. However, in the last years its use has been improved and spread. Due to the advance in this technology, and the higher processing capabilities of mobile devices that include digital cameras, last generation sensors and global positioning systems, it has been possible to develop cheaper and more accurate augmented reality systems. This technology permits the mixing of digital information with real scenarios. Nowadays, it is usual to find applications running in mobile telephones that are able to mix specific places with relevant digital information, display additional contents in museums, such as reconstructions, and many other types of applications.

Taking into account the challenges proposed by augmented reality, this project has made a study of this technology, analyzing the processes performed by augmented reality systems. Concepts such as artificial vision techniques, processing of images and mobile devices architectures are analyzed in the first chapters of this report.

In addition, this work provides an analysis of several implemented systems, describing their components, as well as, the architectures employed in every situation. The work analyzes their differences, their advantages and drawbacks, and their suitability in each case. Thus, this study about architectures and configurations of augmented reality systems aims to be a guide for those entities that want to implement an augmented reality system.

Besides the analysis of augmented reality, this work includes the development of two prototypes of augmented reality systems. One oriented in the educational field, and other focused on digital homes. The first prototype is a displayer of additional contents developed for the Computer Science Museum García Santesmases. The second prototype is aimed for the management of home automation machines. In both cases markers and mobile phones are used.

Finally, and as a brief summary, augmented reality is an emergent technology that promotes a broad array of research areas. Taking into account the improvements that this technology can provide to different areas, such as in industry, technology or education, it is quite possible that in the next years augmented reality will be present in the everyday life.

Keywords: Augmented reality, virtual reality, artificial vision, MIGS, mixed reality, home automation, Android.

Resumen

Históricamente, la realidad virtual ha sido una tecnología que se encontraba al alcance de unos pocos. Sin embargo, en los últimos años ha ido mejorando y ofreciendo nuevas posibilidades de uso. Debido al avance de esta tecnología y gracias a que los dispositivos móviles disponen de mayor capacidad de procesamiento e incluyen cámaras digitales, sensores de última generación y sistemas de localización global, ha sido posible desarrollar sistemas más precisos y asequibles de realidad aumentada. Esta tecnología permite algo que hasta hace bien poco era impensable, mezclar información digital con escenarios reales. Actualmente, y cada vez más, es habitual encontrar aplicaciones que funcionan en teléfonos móviles y que son capaces de mezclar lugares de interés con información digital relevante, visualizar contenidos adicionales en museos, tales como reconstrucciones, y un largo etcétera de ejemplos.

Vistas las posibilidades de trabajo que ofrece la realidad aumentada, este proyecto ha realizado un estudio de esta tecnología, analizando los procesos que se llevan a cabo en los sistemas de realidad aumentada. Conceptos como técnicas de visión artificial, tratamiento de imágenes y arquitecturas en dispositivos móviles son analizados en los primeros capítulos de la memoria.

Este trabajo ofrece, además, un análisis de diversos sistemas ya implantados, describiendo sus componentes, así como las arquitecturas empleadas en cada situación. El trabajo analiza las diferencias entre estas arquitecturas, las ventajas e inconvenientes de usar cualquiera de las mismas, y su idoneidad según el caso. Así, este estudio sobre arquitecturas y configuraciones de sistemas de realidad aumentada pretende ser una guía para aquellas entidades que pretendan implantar un sistema de realidad aumentada.

Aparte del estudio y análisis de la realidad aumentada, este trabajo incluye la realización de dos prototipos de sistemas de realidad aumentada orientados a mejorar dos áreas tan diferentes entre sí como son el campo educativo y los hogares digitales. El primero de los prototipos consiste en un visualizador de contenidos adicionales desarrollado para el Museo de Informática García Santesmases. El segundo de los prototipos consiste en un sistema de control de aparatos que se pueden encontrar en el hogar, con el fin de explorar las posibilidades que ofrece la realidad aumentada en esta área de trabajo. En ambos casos se utilizan marcadores y teléfonos móviles.

Finalmente, y a modo de resumen, podríamos decir que la realidad aumentada, es una tecnología emergente que abre un amplio abanico de posibilidades de trabajo y estudio. Así, considerando las posibilidades que puede aportar en campos de conocimiento y de trabajo tan diferentes entre sí, como el industrial, tecnológico o educativo, es muy probable que en los próximos años se encuentre completamente instalada en la vida cotidiana.

Palabras clave: Realidad aumentada, Realidad virtual, Visión artificial, MIGS, Realidad mezclada, Domótica, Android.

Agradecimientos

En primer lugar, agradecer a mi director Antonio Navarro Martín la oportunidad de poder llevar a cabo este proyecto, para mi juicio, tan apasionante y satisfactorio. Gracias a él he conseguido introducirme en el mercado laboral, tarea complicada en los días que vivimos. También agradecerle la paciencia que ha tenido conmigo a lo largo de este trabajo. Y sobre todo por su calidad como persona, pocos educadores que yo haya conocido han despertado mi admiración. De nuevo gracias.

No puedo dejar de citar a mi familia, mi hermano David, a quien deseo lo mejor en su nueva etapa de la vida, a mis padres por todo el apoyo y la infinita paciencia que han tenido siempre conmigo, a mi tía Luci que, junto a mi abuela Carmen, es lo más parecido que he tenido a una segunda madre. Tampoco puedo olvidar a mi novia Ariana que siempre me escucha y pone cara de interés aunque los dos sepamos que le aburro enormemente. Gracias por estar siempre ahí. Y cómo no, a la hermana que gano en breve. Isabel, bienvenida a la familia. No quisiera olvidarme de mi tía Ana que está pasando por un momento muy duro en la vida, seguro que superas este bache.

Desde el punto de vista profesional, agradecer el apoyo a mi supervisor en Telefónica I+D, José Relaño, que siempre aguanta de forma estoica cualquier duda surgida. También agradecer a mis compañeros Carmen, Raúl, Carlos y Luis por haberme echado un cable siempre que lo he necesitado.

También quisiera agradecer el apoyo recibido por el personal de la Universidad Complutense de Madrid, así como de Telefónica I+D Madrid.

Por último, recordar a todos mis amigos, los de toda la vida de Logroño y los que conocí en Madrid, que son como si fuesen de toda la vida. No cito nombres porque es muy posible que me deje alguno, pero los que son ya saben a quienes me refiero.

1. INTRODUCCION	20
2. REALIDAD AUMENTADA	25
2.1 TAREAS EN REALIDAD AUMENTADA.....	25
2.2 CAPTACIÓN DE LA ESCENA.....	26
2.3 IDENTIFICACIÓN DE ESCENAS	27
2.3.1. <i>Reconocimiento por marcadores</i>	27
2.3.2. <i>Reconocimiento sin marcadores</i>	30
2.4 TÉCNICAS DE MEZCLADO DE REALIDAD Y AUMENTO	30
2.4.1 <i>Fundamentos dimensionales</i>	30
2.4.2 <i>Librerías de aumento</i>	30
2.4.3 <i>OpenGL</i>	30
2.4.4 <i>OpenGL ES</i>	31
2.5 VISUALIZACIÓN DE ESCENA.....	31
2.5.1 <i>Sistemas de bajo coste</i>	31
2.5.2 <i>Sistemas de alto coste</i>	32
2.6 RECONOCIMIENTO DE IMÁGENES	33
2.6.1 <i>Adquisición de imágenes</i>	34
2.6.2 <i>Procesamiento digital</i>	34
2.6.3 <i>Segmentación</i>	35
2.6.4 <i>Representación y descripción</i>	37
2.6.5 <i>Reconocimiento e interpretación</i>	38
2.6.6 <i>Una alternativa a las técnicas anteriores: similitud de imágenes</i>	40
2.7 IDENTIFICACIÓN POR POSICIONAMIENTO.....	40
2.7.1 <i>Sistema de posicionamiento global (GPS)</i>	40
2.7.2 <i>Uso de sensores de movimiento</i>	41
2.8 FORMATOS DE SEÑALES DE VÍDEO E IMÁGENES.....	41
2.8.1 <i>Formato RGB</i>	41
2.8.2 <i>Formato YUV</i>	41
2.8.3 <i>Formato YCbCr 420</i>	42
2.9 PLATAFORMAS MÓVILES.....	43
2.9.1 <i>Symbian OS</i>	43
2.9.2 <i>Windows Mobile</i>	43
2.9.3 <i>iPhone OS</i>	44
2.9.4 <i>Android</i>	44
3. ANÁLISIS DE SISTEMAS DE REALIDAD AUMENTADA.....	47
3.1 COMPONENTES HARDWARE Y SOFTWARE.....	47
3.1.1 <i>Componentes Hardware</i>	47
3.1.2 <i>Componentes Software</i>	48
3.2 ANÁLISIS DE SISTEMAS DE REALIDAD AUMENTADA	48
3.2.1 <i>ULTRA</i>	50
3.2.2 <i>MARCH</i>	50
3.2.3 <i>ASTOR</i>	50
3.2.4 <i>POLAR</i>	52
3.2.5 <i>Layar</i>	53
3.2.6 <i>LOCUS</i>	54
3.3 ARQUITECTURAS DE SISTEMAS DE REALIDAD AUMENTADA	55
3.3.1 <i>Arquitecturas en sistemas autónomos de realidad aumentada</i>	56
3.3.2 <i>Arquitecturas en sistemas distribuidos de realidad aumentada</i>	57
4. IMPLANTACION DE UN SISTEMA DE REALIDAD AUMENTADA.....	61
4.1 ARQUITECTURAS PARA PROVEEDORES DE REALIDAD AUMENTADA	61
4.2 ARQUITECTURAS PARA CLIENTES DE REALIDAD AUMENTADA	62
4.3 SISTEMAS AUTÓNOMOS FRENTE A SISTEMAS DISTRIBUIDOS.....	62
4.3.1 <i>Características de un sistema autónomo</i>	64
4.3.2 <i>Características de un sistema distribuido de realidad aumentada</i>	64

4.3.3 Conclusiones.....	66
4.4 CONFIGURACIÓN POR PARTE DEL PROVEEDOR Y DEL CLIENTE EN LOS SISTEMAS DESARROLADOS EN ESTE TRABAJO	67
4.4.1 Requisitos del proveedor.....	68
4.4.2 Requisitos del cliente.....	70
5. APLICACIONES AL E-LEARNING Y A LA DOMÓTICA	73
5.1 MOTIVACIÓN DE LAS APLICACIONES	73
5.2 IDENTIFICACIÓN DE ESCENARIOS.....	74
5.2.1 Técnicas clásicas de visión artificial	75
5.2.2 Uso de redes neuronales	76
5.2.3 Reconocimiento inicial de marcadores	80
5.2.4 Búsqueda laberíntica	84
5.2.5 El problema de las rotaciones y transformaciones: la solución definitiva	87
5.3 TÉCNICAS DE AUMENTO	93
5.4 ARQUITECTURA DEL SISTEMA.....	95
5.4.1 Interfaz de usuario	96
5.4.2 Gestión de búsquedas	98
5.4.3 Gestión de comunicaciones.....	100
5.4.4 Paquete de utilidades.....	101
5.5 MUSEO DE INFORMÁTICA GARCÍA SANTESMASES.....	101
5.6 APLICACIÓN DE DOMÓTICA	105
6. CONCLUSIONES Y TRABAJO FUTURO.....	111
REFERENCIAS	115
AN ANALYSIS OF AUGMENTED REALITY SYSTEMS.....	117

1. INTRODUCCION

Hasta hace pocos años la opción de disponer de información digital añadida sobre cualquier objeto sin alterar su entorno era prácticamente imposible. Sin embargo, hoy en día ya es posible ver casi cualquier tipo de información digital sobre el objeto que sea. Esto se debe a la realidad aumentada, una tecnología a disposición de cualquier persona que permite superponer sobre un escenario real cualquier tipo de contenido digital, sea de la índole que sea, visual, auditivo, etc.

A pesar de que en 1966 Ivan Sutherland inventó el primer casco de visualización (*Head-Mounted Display*, HMD) [SUT 68, VRES10], no ha sido prácticamente hasta el siglo XXI cuando esta área de investigación ha empezado a coger fuerza y relevancia. Esto se debe a que los esfuerzos del siglo pasado estaban más centrados en la realidad virtual, que se diferencia de la realidad aumentada en que todo lo que el usuario observa es una simulación de la realidad.

La especial relevancia que está teniendo en estos tiempos la realidad aumentada no proviene de su naturaleza novedosa, ya que esta tecnología ya venía siendo utilizada en las cabinas de los aviones durante el siglo pasado, como puede verse en la figura 1.1. Lo que ha producido que haya aumentado su importancia ha sido su apertura a la vida cotidiana de las personas. En los últimos años se ha visto que esta área de investigación podía ser muy útil para campos de trabajo muy diferentes. Cada día es más común encontrar museos que utilizan realidad aumentada para exponer a sus visitantes más información sobre sus elementos sin tener que manipularlos. Y esto es sólo un pequeño ejemplo de aplicación de esta tecnología. Sectores tan diversos como el sanitario, educativo, turístico, etc., se han unido al camino de la realidad aumentada.



Figura 1.1. Ejemplo de uso de realidad aumentada en la cabina de un avión.

Muy probablemente la otra gran razón por la que cada día suena con más fuerza la realidad aumentada es que ha dejado de estar al alcance de unos pocos, estando en la actualidad accesible para todo tipo de usuarios. Antes eran necesarios costosos equipos para poder visualizar los contenidos digitales sobre la escena real, mientras que actualmente cualquier teléfono móvil de última generación permite llevar a cabo estas funciones.

A lo largo de esta memoria se expondrá en mayor detalle cómo crear una aplicación de realidad aumentada, aunque básicamente existen dos tareas básicas que se deben acometer: saber qué se está viendo y qué información se debe visualizar como aumento.

Gracias a su aplicabilidad en muchos ámbitos de la vida, muchas empresas y organizaciones han puesto esta tecnología en su punto de mira, realizando inversiones importantes para aprovechar sus posibilidades.

Así, los departamentos de publicidad y marketing de las empresas han decidido coger el tren de la realidad aumentada, apostando fuertemente por ella. Los ejemplos expuestos a continuación son una clara muestra de ello.

Por ejemplo, en Navidades del 2009, la empresa Cruzcampo ofreció a los internautas que dispusieran de una webcam poder visualizar en una cartulina su logotipo de forma interactiva.



Figura 1.2. Ejemplo de funcionamiento del software de realidad aumentada de IKEA.

La empresa láctea Central Lechera Asturiana ha comercializado en 2010 una nueva serie de envases que incluyen un marcador mediante el cual, gracias a una cámara y a un software especial, cualquier cliente puede ver en su ordenador al logotipo de la empresa en 3D encima de su envase.

El gigante sueco de materiales mobiliarios IKEA, ha presentado en 2010 también una nueva aplicación para teléfonos móviles que permite, mediante el uso de marcadores colocados donde el usuario desee en su hogar, poder visualizar la forma en que quedaría realmente (figura 1.2).

También cabe destacar que las aplicaciones que más impacto están teniendo son aquellas que sirven de referencia turística, en las que se puede visualizar en la pantalla del teléfono móvil dónde se encuentran los puntos de interés del turista. Un ejemplo de este tipo de aplicaciones es Layar, que se puede ver en la figura 1.3.



Figura 1.3. Uso de Layar como aplicación de realidad aumentada para turismo.

Sin embargo, la realidad aumentada dista mucho de ser una disciplina madura en la informática. Las líneas de investigación están completamente abiertas desde los métodos de identificación de escenarios hasta las técnicas de visualización.

En el área de investigación en las técnicas de identificación de escenas cabe destacar un estudio realizado en la Boston College que consiste en una técnica de reconocimiento de imágenes diferente a las empleadas hasta el momento [JIA 09]. La idea de estos investigadores es simular el comportamiento neuronal del cerebro en el procesamiento de imágenes. Esta revolucionaria técnica, según afirman sus autores, disminuyen considerablemente los tiempos de ejecución en el reconocimiento e incrementan los porcentajes de identificación de imágenes.

Otro proyecto interesante es el que están llevando a cabo en el Departamento de Defensa de Estados Unidos. Mediante este proyecto se pretende mejorar la infraestructura actual del sistema de geoposicionamiento global [GPS 10], ampliando a tres el número de señales para uso civil, mejorando la precisión del sistema e incrementando las opciones para los sistemas de aviación [DOD 08, GPS 10]. Estos avances y los que vendrán es posible que en un futuro cercano permitan que los sistemas de realidad aumentada en entornos cerrados puedan utilizar técnicas de geoposicionamiento para la fase de identificación de escenarios.

En este contexto el presente trabajo de máster tiene por objetivos:

- Realizar un estado del arte sobre los sistemas de realidad aumentada.
- Definir la infraestructura hardware y software necesaria para proporcionar servicios de realidad aumentada.
- Llevar a la práctica dichos servicios construyendo dos aplicaciones de realidad aumentada: una para el Museo de Informática García Santesmases, y otra para Telefónica I+D, empresa en la que el autor de este trabajo trabaja como becario.

Así, este trabajo pretende ser una guía para cualquier entidad interesada en implementar un sistema de realidad aumentada, como Telefónica I+D.

Para cubrir los objetivos anteriormente mencionados, el capítulo 2 define las tareas involucradas en el proceso de aumento de la realidad, e incluye un estado del arte de las tecnologías relacionadas con estas tareas. El capítulo 3 analiza algunos de los sistemas de realidad aumentada más destacados, incluyendo los componentes hardware y software de los mismos, así como sus arquitecturas. El capítulo 4 define la infraestructura necesaria para la implantación de un sistema de realidad aumentada, analizando las ventajas e inconvenientes de las distintas alternativas posibles. El capítulo 5 implanta dicha infraestructura en dos aplicaciones concretas. Finalmente se presentan las conclusiones y el trabajo futuro.

Cabe destacar que este trabajo ha producido una publicación titulada *An Analysis of Augmented Reality Systems* en The Fifth International Multi-Conference on Computing in the Global Information Technology, congreso cuyas actas son publicadas por IEEE Computer Society.

2. REALIDAD AUMENTADA

Este capítulo define las tareas que se llevan a cabo en el proceso de aumento de la realidad. También incluye un estado del arte de las tecnologías involucradas en dichas tareas.

La sección 1 introduce el concepto de realidad aumentada y describe, a grandes rasgos, el funcionamiento de esta tecnología. Las siguientes cuatro secciones exponen una descripción más amplia de cada una de las tareas expuestas en la sección 1. Así, la sección 2 define las técnicas de captación de escenarios, mientras que en la sección 3 se exponen, a modo de resumen, las técnicas habitualmente empleadas en la identificación de la escena, explicando en qué consiste el reconocimiento mediante marcadores y sin ellos. Las siguientes dos secciones explican las técnicas de mezclado de realidad y aumento, sección 4, y de visualización final, sección 5. En la sección 6 se define en profundidad el proceso de reconocimiento de escenas mediante imágenes, exponiendo a modo de estado del arte las diferentes técnicas existentes en el campo de la visión artificial y su aplicación en los sistemas de realidad aumentada. La sección 7, por su parte, expone las técnicas de reconocimiento de escenas basadas en posicionamiento. En la sección 8 se introducen algunos conceptos básicos sobre las señales de vídeo e imágenes y sus formatos habituales. Por último, puesto que este proyecto pretende construir un sistema para móviles, la sección 9 expone las diferentes plataformas móviles existentes actualmente en el mercado.

2.1 Tareas en realidad aumentada

Los sistemas de realidad aumentada son herederos de una tecnología más antigua: la realidad virtual. Por tanto, antes de dar una definición y una explicación de la tecnología, es necesario introducir al lector en las bases de la realidad virtual.

La realidad virtual es un sistema que interactúa con el usuario simulando un entorno real en un entorno ficticio [SOM 02]. Este tipo de sistemas generan una sensación de que lo que está ocurriendo es real, aunque no lo sea. La realidad virtual es, básicamente, una técnica fotográfica de 360 grados, que permite movimientos y desplazamientos tanto verticales como horizontales, ofreciendo libertad absoluta de acción, lo que genera una amplia sensación de realismo en el usuario mediante una estimulación de los cinco sentidos.

En términos generales, se podría resumir la realidad virtual como un sistema informático que simula un entorno ficticio ofreciendo al usuario la sensación de que es real, así como todo lo que ocurre en él.

Una vez explicado de forma breve qué es la realidad virtual, se puede dar una definición más concisa y clara sobre qué es la realidad aumentada.

La realidad aumentada toma parte de la simulación de un entorno ficticio de la realidad virtual para añadirlo sobre un escenario real. De esta manera la escena que nuestros sentidos detectan en un escenario físico real, está enriquecida con información que no existe en el escenario físico original y que es proporcionada por un sistema informático.

Se podría definir la realidad aumentada como aquellos sistemas informáticos que mezclan información virtual de cualquier tipo, desde imágenes 2D, texto o figuras 3D, con un escenario físico real [CAW 08, BIM 05].

Para facilitar la comprensión de la tecnología se expondrá un sencillo ejemplo que ilustre lo visto hasta el momento. Imaginemos un monumento emblemático como podría ser la puerta de Alcalá en Madrid, España. Nos gustaría que al pasar al lado de él pudiésemos ver información de cualquier tipo sobre sí mismo como, por ejemplo, un rótulo que indicase que, efectivamente, es la puerta de Alcalá. Siendo más ambiciosos, quizá nos gustaría que apareciese información histórica sobre el monumento. Lamentablemente, a día de hoy es imposible que sobre los objetos físicos pueda aparecer información digital sin ayuda de proyectores 2D o proyectores holográficos. Es más, es posible que a mucha gente no le gustase observar los monumentos cargados de información digital. La solución a este problema la ofrece la realidad aumentada. En la actualidad cualquier persona con un teléfono móvil lo suficientemente avanzado podría capturar la escena con su cámara y visualizar el monumento con la información digital aumentada por la pantalla del teléfono. Incluso es posible que sin una cámara también pudiese ver la realidad mezclada sobre su dispositivo. Más adelante se irán exponiendo y ampliando estos conceptos.

Como se ha visto en el ejemplo expuesto, en cualquier sistema de realidad aumentada se requieren dispositivos que identifiquen el escenario real y lo clasifiquen así como que visualicen tanto entorno real como información digital.

Por otro lado, en todo sistema de realidad aumentada son necesarias, al menos, cuatro tareas fundamentales para poder llevar a cabo el proceso de aumento. Estas tareas se explicarán con mayor profundidad en secciones posteriores, pero básicamente son [LOP 10]: (i) captación de escena; (ii) identificación de escena; (iii) mezclado de realidad y aumento; y (iv) visualización. En la figura 2.1 se puede ver de forma esquemática este proceso.

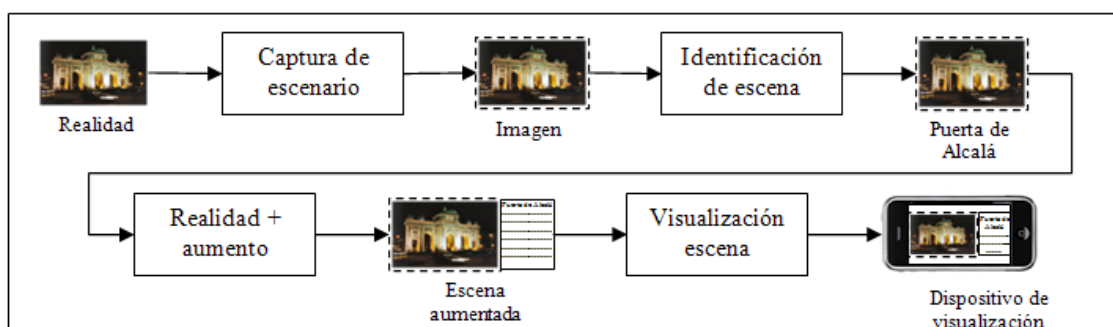


Figura 2.1. Esquema de funcionamiento de un sistema de realidad aumentada típico.

A continuación describiremos estas cuatro tareas necesarias presentes en todo sistema de realidad aumentada. Después se describen técnicas fundamentales para llevar a cabo estas tareas y las principales plataformas móviles que las soportan.

2.2 Captación de la escena

Una de las tareas más importantes en cualquier sistema de realidad aumentada es la de identificar el escenario que se desea aumentar. En el caso de los sistemas que utilicen reconocimiento visual, es indispensable contar con algún mecanismo que permite recoger la escena para que pueda ser posteriormente procesada. En esta sección se analizan los diferentes tipos de dispositivos físicos que permiten captar dicho escenario.

Los dispositivos de captura de imágenes son dispositivos físicos que recogen la realidad que deberá ser ampliada [BIM 05]. A grandes rasgos, estos dispositivos se pueden agrupar, principalmente, en dos conjuntos:

- Dispositivos *video-through*: dentro de este grupo se encuentran aquellos dispositivos que realizan la captura de imágenes o video que se encuentran aislados de los dispositivos de visualización. En este conjunto se encontrarían las cámaras de video o los terminales móviles (siempre y cuando tengan una cámara).
- Dispositivos *see-through*: son los dispositivos que realizan tanto la tarea de capturar la escena real como de mostrarla con información aumentada al usuario. Estos dispositivos acostumbran a trabajar en tiempo real, haciéndolos no sólo más costosos en presupuesto sino también en complejidad. Dentro de este grupo se encontrarían aquellos dispositivos conocidos como *head-mounted*. Cabe remarcar que estos dispositivos *see-through* llevan años siendo usados, por ejemplo, en los *Head Up Displays* (HUDs) utilizados por los aviones de combate para mostrar información al piloto sobre altura, velocidad, identificación de blancos, y otros sin necesidad de apartar la vista de la zona frontal de la cúpula de su cabina.

La figura 2.2 muestra un ejemplo de este tipo de dispositivos.

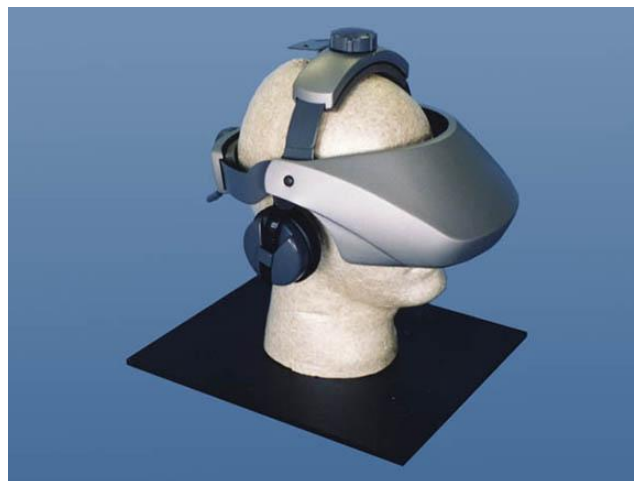


Figura 2.2. Prototipo de HMD.

2.3 Identificación de escenas

El proceso de identificación de escenas consiste en averiguar qué escenario físico real es el que el usuario quiere que se aumente con información digital [BIM 05]. Este proceso puede llevarse a cabo, básicamente, de dos maneras: utilizando marcadores o sin utilizarlos [LOP 10].

2.3.1. Reconocimiento por marcadores

En los sistemas de realidad aumentada, un marcador es un objeto cuya imagen es conocida por el sistema [CAW 08, CHO 09, TAT 07]. Las maneras en que el sistema conoce el marcador se pueden agrupar en tres conjuntos, mediante su geometría, su color o mediante ambas características.

Para llevar a cabo el procesamiento de un marcador se puede escoger cualquiera de los mecanismos que se presentan en la sección 2.6, pero hay que tener en cuenta que, dichos mecanismos, suelen implicar una gran capacidad de cálculo y, por tanto, afecta al rendimiento del sistema.

Habitualmente para el reconocimiento de marcadores se utiliza un primer escaneo sobre la imagen más pesado computacionalmente para localizar el marcador que se busca. Una vez localizado el mecanismo de actuación suele ser el descrito a continuación.

En primer lugar se establece un rango de variación en el movimiento del marcador para el posterior fotograma. En el procesamiento de dicho fotograma, el rango de búsqueda ya se encuentra acotado a un espacio muy inferior al inicial, por lo que el tiempo de procesamiento decae considerablemente. Además, por norma general, se utilizan menos técnicas de reconocimiento, empleando el menor número de cálculos para localizar el marcador. Una vez detectado, se procede a las tareas necesarias de mezclado y aumento en los sistemas de realidad aumentada. Este proceso se efectúa de forma iterativa mientras la aplicación esté en ejecución.

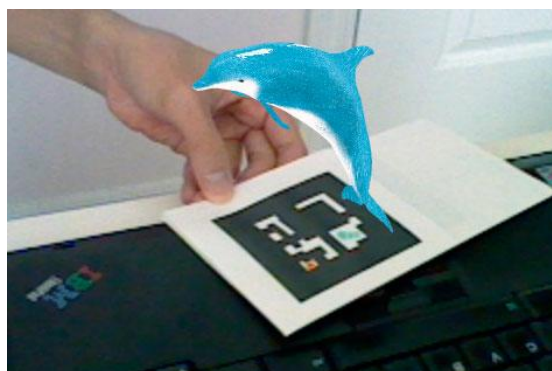


Figura 2.3. Ejemplo de aplicación de marcador en realidad aumentada.

El proceso recientemente descrito sólo modificará su comportamiento si en algún fotograma en la región de búsqueda no se encontrase el marcador. En esta circunstancia existen diversas posibilidades de actuación:

- Realizar un nuevo escaneo sobre toda la imagen en busca del marcador. Este proceso puede ser efectivo si el marcador ha sido desplazado a una posición alejada de la anterior secuencia o si no se encuentra.
- Buscar de forma recursiva en las regiones vecinas el marcador. Esta solución podría ser óptima si el marcador desplazado se encuentra cerca de la región de búsqueda inicial.
- Utilizar predicción de movimiento. Esta tarea se puede llevar a cabo mediante la variación del movimiento analizando las imágenes o bien mediante el uso de acelerómetros. En este proyecto se utilizará la segunda opción.

En ambos casos, si el marcador ha sido detectado se procedería a utilizar el mecanismo iterativo ya expuesto.

Antes de concluir la explicación del reconocimiento por marcadores, es necesario hacer resaltar que el número de marcadores que puede reconocer este tipo de sistemas no es ilimitado, sino que es dependiente del algoritmo utilizado.

2.3.2. Reconocimiento sin marcadores

De la misma forma, es posible identificar la escena mediante reconocimiento de imágenes o mediante la estimación de la posición. También es posible encontrar sistemas que realicen una combinación de ambas en función de la situación. A este tipo de identificación se le denominará híbrida. Dentro de cada uno de estos dos conjuntos de técnicas se pueden encontrar diversas variaciones que dependerán en gran medida de las prestaciones que deba ofrecer el sistema así como de sus posibilidades técnicas.

Puesto que las técnicas habituales en este proceso se encuentran centradas en el reconocimiento visual de la escena, se describirá en mayor profundidad ese tipo de técnicas. No obstante, esto no significa que no se puedan utilizar cualquier otro tipo de técnica más apropiada para cada situación, como podrían ser el análisis en la intensidad de señales de radiofrecuencia o de señales infrarrojas.

El presente proyecto utiliza, un reconocimiento híbrido, combinando detección de marcadores con un motor de detección de movimiento mediante acelerómetros que permite conocer el desplazamiento ocurrido en cada instante para ayudar al sistema de reconocimiento a hacer una búsqueda más inteligente.

2.4 Técnicas de mezclado de realidad y aumento

Una vez descrito el proceso de identificación de escenarios, el siguiente proceso que tiene lugar en los sistemas de realidad aumentada es de sobreponer la información digital que se quiere ampliar sobre la escena real capturada. Cabe resaltar, que esta información digital de aumento puede ser tanto de carácter visual como auditivo o táctil, lo que por lo general, en la mayoría de sistemas de realidad aumentada sólo se ofrecen los de tipo visual.

2.4.1 Fundamentos dimensionales

El primer concepto que hay que diferenciar en este proceso es el tipo de información que se quiere aumentar. En este apartado se puede distinguir entre dos tipos básicos de información: 2D y 3D [CAW 09].

La información bidimensional constituye un plano geométrico digital en el que cada punto del plano se representa por medio de un píxel que, habitualmente, es un vector de colores en escala RGB (rojo-verde-azul) [RGB 98]. Cada uno de los tres valores que conforman el vector oscila en el rango 0-255, siendo la mezcla de los tres valores 0 la representación del color negro y 255 la del color blanco. Dentro de este tipo de información, no sólo se encuentran las imágenes, sino que también serán clasificados los contenidos textuales, debido a sus características similares de representación.

El otro tipo de información digital anteriormente expuesta es la tridimensional. Este tipo de información de la realidad consiste en un conjunto de vectores multidimensionales para cada punto del plano tridimensional (x, y, z). Al igual que sucedía con la información bidimensional, cada vector uno de estos puntos está formada por un vector RGB o cualquier otro tipo de representación del color.

Una vez definidos los tipos básicos de información digital de aumento, se puede realizar la descripción del funcionamiento real de las técnicas de aumento. En los sistemas de realidad aumentada, excepto en los sistemas que utilizan hologramas tridimensionales o similares, los dispositivos de visualización son en dos dimensiones, como pueden ser las pantallas de ordenadores, teléfonos móviles, etc. Este suceso puede llevar a pensar que sólo es posible representar información bidimensional y, aunque esto es cierto, es posible simular la sensación de tridimensionalidad en un plano 2D.

Para realizar la conversión de una imagen en 3D al plano bidimensional se suele utilizar la técnica de proyección de perspectiva (o proyección de puntos). Esta técnica consiste en simular la forma en que el ojo humano recibe la información visual por medio de la luz y cómo genera la sensación 3D. Este proceso consiste en la superposición de dos imágenes bidimensionales captadas desde diferentes ángulos, dando la sensación de profundidad inexistente en imágenes 2D. La ecuación 2.1 adjunta define la forma de trasladar el plano tridimensional al plano bidimensional.

$$u = \frac{f(x)}{z}, v = \frac{f(y)}{z}$$

Ecuación 2.1. Proyección de un punto 3D (x, y, z) en un plano 2D (u, v).

2.4.2 Librerías de aumento

Para el proceso de aumento es necesario disponer de software adecuado para sobreponer a la imagen real la información aumentada deseada. Para este propósito existen diversas librerías disponibles al público. Las más famosas son:

- *ARToolKit*: librería de realidad aumentada que permite la detección de unos marcadores específicos y realiza las tareas de superposición de imágenes [ART 10]. Está desarrollada en lenguaje C++.
- *ARToolKitPlus*: versión más especializada de la librería ARToolKit [ART P 10].
- *JARToolKit*: es la versión de la librería ARToolKit desarrollada en lenguaje Java [JART 10].

Las librerías que se acaban de presentar tienen la ventaja de formar en sí paquetes de realidad aumentada con todas las funcionalidades necesarias para sistemas por marcadores. No obstante, si el reconocimiento de escenarios se realiza por otro mecanismo al de detección de marcadores, estas librerías pierden su utilidad.

2.4.3 OpenGL

OpenGL es una especificación estándar que define una API multilenguaje y multiplataforma para escribir aplicaciones que produzcan y consumas gráficos en 2D y 3D [SHR 05, SHR 00]. Esta interfaz dispone de más de 250 funciones definidas.

El funcionamiento básico consiste en aceptar primitivas tales como puntos, líneas y polígonos y convertirlas en píxeles. Este proceso se lleva a cabo en una pipeline gráfica conocida como la máquina de estados de OpenGL. La figura 2.4 muestra la pipeline que utiliza OpenGL.

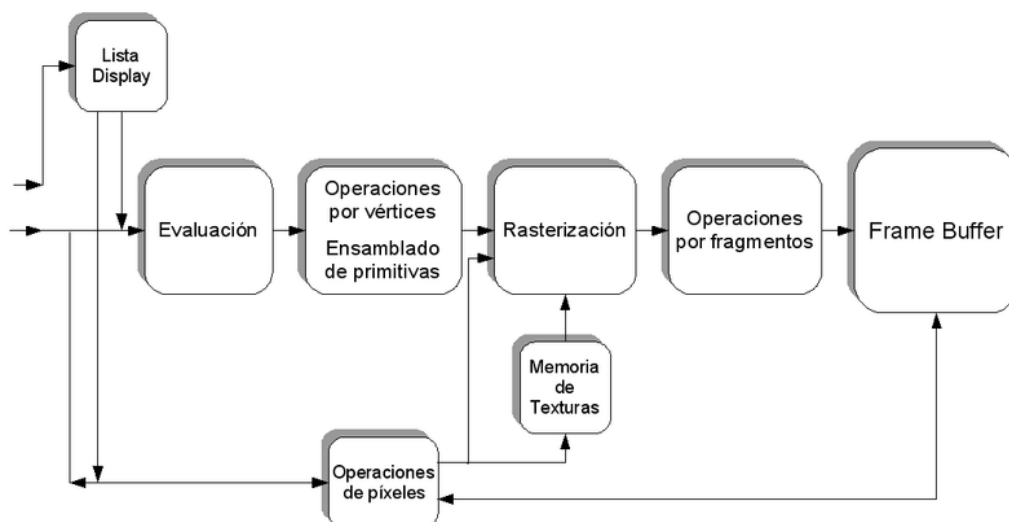


Figura 2.4. Pipeline de OpenGL.

2.4.4 OpenGL ES

OpenGL ES es la variante simplificada de la especificación OpenGL para dispositivos móviles o terminales empujados [GL ES 10]. Debido al éxito de la especificación en sistemas informáticos tradicionales y a la tendencia actual de trasladar a los dispositivos móviles la funcionalidad de los PCs básica, la especificación OpenGL ES ha tenido una altísima aceptación en los fabricantes de teléfonos móviles, siendo la utilizada por los líderes del mercado. Algunos ejemplos de plataformas que utilizan la tecnología OpenGL ES para manejo de gráficos 2D y 3D son:

- Symbian OS.
- Android.
- Maemo del Nokia N900.
- SDK del iPhone.
- PlayStation 3.

2.5 Visualización de escena

Dentro de los sistemas de realidad aumentada, el último proceso que se lleva a cabo, y quizás uno de los más importantes, es el de visualización de la escena real con la información de aumento. Sin este proceso, la realidad aumentada no tendría razón de ser.

En esta sección se describirán los mecanismos de visualización habituales. Para ello se realizará una clasificación de estos entre sistemas de bajo coste y sistemas de alto coste.

2.5.1 Sistemas de bajo coste

Este tipo de sistemas de visualización se caracterizan por estar presentes en la gran mayoría de dispositivos, tanto móviles como fijos y que cualquier usuario puede obtener de forma barata y sencilla. Dentro de este grupo se pueden diferenciar entre dos tipos básicos de sistemas: sistemas móviles y sistemas fijos.

Los sistemas de visualización móviles se refieren a los integrados en dispositivos móviles como teléfonos o PDAs. Estos sistemas se caracterizan por venir integrados en los dispositivos de fábrica, por lo que el usuario no debe preocuparse por adquirir elementos adicionales. En este tipo de sistemas se denominan dispositivos de visualización a las pantallas de dichos terminales móviles. Una de las características más relevantes en este tipo de sistemas es la baja definición y calidad de las imágenes de salida. Esto se debe a que los componentes

hardware integrados en ellos encargados de tareas de carácter gráfico no disponen de gran potencia de cálculo ni de altas memorias integradas.

El otro tipo de sistema de visualización de bajo coste anteriormente expuesto son aquellos dispositivos fijos o que, a pesar de su movilidad, se pueden considerar como ordenadores personales. A diferencia de lo que sucede en los sistemas móviles, este tipo de sistemas suelen disponer de hardware adecuado para realizar tareas de visualización más complejas, generando de esta forma imágenes de salida de mayor calidad.

2.5.2 Sistemas de alto coste

Los sistemas de alto coste son escasos dentro de la realidad aumentada, aunque se pueden encontrar casos de utilización, como es el caso de los Head Up Displays (HUDs) [OKA 96]. Este tipo de sistemas tienen además la característica de ser interactivos con el usuario desde el punto de vista de que se libera a éste de dispositivos de visualización físicos, pudiendo ver la información aumentada mediante proyecciones sobre elementos físicos reales. Para poder hacer realidad este fenómeno se utilizan dispositivos de proyección en 2D o, sobre todo en los últimos años, en 3D. No obstante, también se pueden encontrar dentro de este grupo aquéllos dispositivos de última generación como los empleados por el ejército o en las simulaciones de vuelo que, debido a la criticidad de su servicio y de las altas prestaciones tanto a nivel hardware como software, conllevan un coste bastante elevado.

Por su parte, en el caso de los sistemas de visualización en 3D parece que se está generando un profundo interés en emplear esta tecnología. Las técnicas de representación holográfica en 3D han avanzado considerablemente en los últimos tiempos, siendo posible representar figuras humanas con una elevada calidad, poder interactuar de forma táctil con el holograma e incluso representar en 3D un objeto sin utilizar electricidad, simplemente mediante el uso de espejos. Evidentemente las empresas han visto una fuente de promoción e ingresos destacada en este sector, por lo que casos como el de Virgin dando una gala con un presentador virtual parecen ser solamente el principio de una nueva época en el sector.



Figura 2.5. Holograma interactivo que responde al tacto.

2.6 Reconocimiento de imágenes

El reconocimiento de imágenes consiste en extraer de las imágenes que recibe el dispositivo del usuario la información necesaria para identificar el escenario real que se quiere aumentar [JAV 02].

Dentro de este tipo de técnica de identificación, en el mundo de la realidad aumentada se pueden diferenciar claramente dos conjuntos de técnicas de reconocimiento de imágenes, que son el reconocimiento automático de escenarios mediante técnicas de visión artificial, y el reconocimiento por marcadores específicos definidos y conocidos por el sistema.

En esta sección se expondrán en primer lugar las técnicas de visión artificial empleadas para el reconocimiento de imágenes. La razón por la que no se identificarán cuáles de éstas participan en el reconocimiento automático es porque este tipo de sistemas vienen muy influenciados por el entorno sobre el que operan y sería inviable tratar en este trabajo cómo han implementado todos y cada uno de los sistemas de realidad aumentada estos mecanismos. No obstante, la exposición de las diferentes técnicas dará al lector una visión general de estas tecnologías que podrá ayudarle a elegir unas u otras para cada ocasión.

A grandes rasgos, el proceso de reconocimiento de imágenes consiste en los siguientes pasos [PLA 06, ESC 01, PAJ 01]. En primer lugar se adquiere la imagen mediante algún dispositivo preparado para llevar a cabo esta tarea, como puede ser una webcam. Una vez adquirida la imagen se realiza una etapa de procesamiento para eliminar imperfecciones de la imagen tales como ruido. Cuando se ha preprocesado la imagen se procede a su segmentación para buscar información característica en ella que pueda ser de utilidad a posteriores tareas. Tras la fase de segmentación se procede a la búsqueda de características morfológicas tales como perímetros o texturas. A esta etapa la denominamos representación y descripción. Por último, se procede al reconocimiento e interpretación de la escena mediante redes neuronales, lógica borrosa, etc.

Cabe destacar, que no necesariamente todos los sistemas deben implementar todas estas fases, la decisión vendrá condicionada por las prestaciones que deba ofrecer y el entorno al que se vaya a enfrentar. La figura 2.6 muestra el proceso que se acaba de explicar.

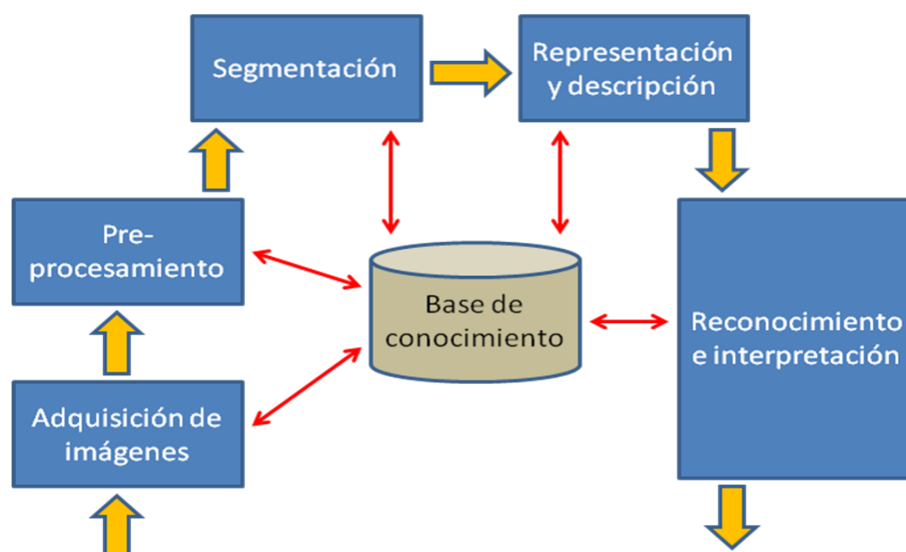


Figura 2.6. Proceso de reconocimiento de imágenes mediante técnicas de Visión Artificial clásica.

A continuación se explica en qué consiste cada fase y que técnicas se encuentran englobadas en ellas.

2.6.1 Adquisición de imágenes

Esta etapa consiste en construir el sistema de formación de imágenes. Con ello se pretende realzar las características visuales de los objetos, como formas o colores, mediante técnicas fotográficas [PLA 06, ESC 01, PAJ 01].

Puesto que esta etapa ya se lleva a cabo de forma automática en los dispositivos de captura de imágenes, no se profundizará más en el tema, ya que sería necesario adentrarse en los fundamentos de la óptica, de la física de sólidos y líquidos, mecanismos de calibración, teoría de señales, etc., y no es el objetivo de este documento explicar estos conceptos que pueden encontrarse en [PLA 06, ESC 01, PAJ 01, JAV 02].

2.6.2 Procesamiento digital

En esta fase se trata de cuantificar y codificar la señal de vídeo o fotográfica recibida en forma de imagen. El objetivo es obtener una nueva imagen que o bien mejore su calidad o que destaque algún atributo significativo de ésta [PLA 06, ESC 01, PAJ 01].

Los problemas de calidad pueden ser por falta o exceso de iluminación o por ruido. Al aplicar realce lo que se pretende conseguir es destacar bordes, regularizar colores, etc.

Las técnicas descritas en este capítulo pueden ser agrupadas en dos conjuntos: las que proceden de las señales y aquellas de carácter heurístico.

En las técnicas para procesado procedentes de las señales se suelen aplicar los siguientes conceptos:

- *Distancias entre píxeles*. Dentro de esta categoría se pueden encontrar las siguientes relaciones:
 - *Relaciones de distancias*: establece la distancia entre píxeles.
 - *Relaciones de conectividad*: establece que dos píxeles adyacentes pertenecen a un mismo elemento.
- *Procesos de convolución y operadores de correlación*: se utilizan para aplicar filtros sobre las imágenes, por ejemplo para eliminar ruido de sal y pimienta.

Por su parte, las técnicas de procesado de carácter heurístico se basan en un conjunto de procedimientos sobre el procesamiento digital de las señales y otros tipos de manipulaciones matemáticas. Este tipo de técnicas se pueden agrupar en tres conjuntos: realce o aumento del contraste, suavizado o eliminación del ruido y detección de bordes. Para llevar a cabo estas técnicas conviene hacerlo sobre imágenes en escala de grises, ya que son efectivas sobre la iluminancia de los objetos. A continuación se describirán brevemente cada conjunto de las técnicas recientemente expuestas.

Las técnicas de realce consisten en aumentar el contraste de las imágenes. Este tipo de procesado se basa en los conceptos de histograma, brillo y contraste [PLA 06, ESC 01, PAJ 01].

Por otro lado, las técnicas de suavizado lo que pretenden es eliminar el ruido que pueda tener la imagen. Existen tres tipos básicos de ruido: gaussiano como el que se muestra en la figura 2.7, impulsional y multiplicativo [PLA 06, ESC 01, PAJ 01]. Durante este proceso se utilizan filtros para eliminar ese ruido, los más comunes son [PLA 06]:

- Filtros paso bajo.
- Filtros gaussianos.
- Filtros basados en la mediana.
- Filtros homomórficos.



Figura 2.7. Imagen con ruido gaussiano.

La última técnica que se va a describir es la detección de bordes. Esta etapa suele preceder a las tareas de segmentación o a la búsqueda de objetos geométricos más complejos. Un borde se define como una región en la que aparece una fuerte variación de la intensidad en los píxeles adyacentes [PLA 06, ESC 01, PAJ 01]. Las técnicas más comunes sobre detección de bordes son las siguientes:

- Técnicas basadas en el operador gradiente, donde se encuentra, por ejemplo, el método de Sobel.
- Operadores basados en la laplaciana.
- Operador de Canny. En la figura 2.8 se ve el efecto de aplicar este tipo de detección.



Figura 2.8. Detección de bordes mediante el operador de Canny.

2.6.3 Segmentación

Esta fase tiene como objetivo agrupar los píxeles por algún criterio de homogeneidad para dividir la escena en regiones de interés. La agrupación de los píxeles se hace a razón de la similitud entre vecinos bajo criterios de luminancia, color, bordes, texturas, ... Una vez realizada esta tarea la unidad más pequeña de trabajo dejará de ser el píxel para convertirse en agrupaciones de píxeles que constituyen objetos [PLA 06, ESC 01, PAJ 01].

Para este proceso se utilizan tres conceptos básicos:

- Similitud entre píxeles agrupados.
- Conectividad entre píxeles.
- Discontinuidad definida por los bordes de los objetos.

En la práctica imponer estas condiciones resulta casi imposible, por lo que se combinan las técnicas basadas en la localización de regiones uniformes con los contornos de los objetos [PLA 06, ESC 01, PAJ 01].

Las técnicas utilizadas en esta fase se pueden agrupar en tres tipos [PLA 06, ESC 01, PAJ 01]: basadas en los contornos de los objetos, basadas en umbrales y en la búsqueda de regiones homogéneas.

El primer tipo de técnicas descritas serán las basadas en los contornos de los objetos [PLA 06]. Para llevar a cabo esta tarea se parte de una detección de bordes anterior que, debido a la presencia de ruido, sombras, etc., no es suficiente para poder segmentar las imágenes. Se trata de agrupar los píxeles etiquetados como bordes empleando la propiedad de conectividad. Habitualmente se utilizan transformadas de Hough para la detección de líneas rectas o círculos [BRA 01].

La siguiente técnica que se describirá es la umbralización [PLA 06, ESC 01, PAJ 01]. Esta técnica es frecuentemente utilizada en entornos industriales, cuando existe una clara diferenciación entre los objetos a extraer y el fondo de la escena. Los principios que rigen son la similitud entre píxeles pertenecientes a un objeto y sus diferencias respecto al resto. Para que se pueda llevar a cabo esta técnica la escena debe tener un fondo uniforme y objetos parecidos.

Por último, la técnica basada en la búsqueda de regiones está basada en los criterios de similitud y continuidad de los píxeles que forman una región [PLA 06, PAJ 01]. El proceso suele consistir en una primera etapa de división de la imagen en árboles cuaternarios en los que los nodos son regiones de píxeles con alguna similitud entre ellos y una segunda etapa en la que se fusionan las regiones similares. La figura 2.9 muestra el resultado de aplicar segmentación sobre una imagen.

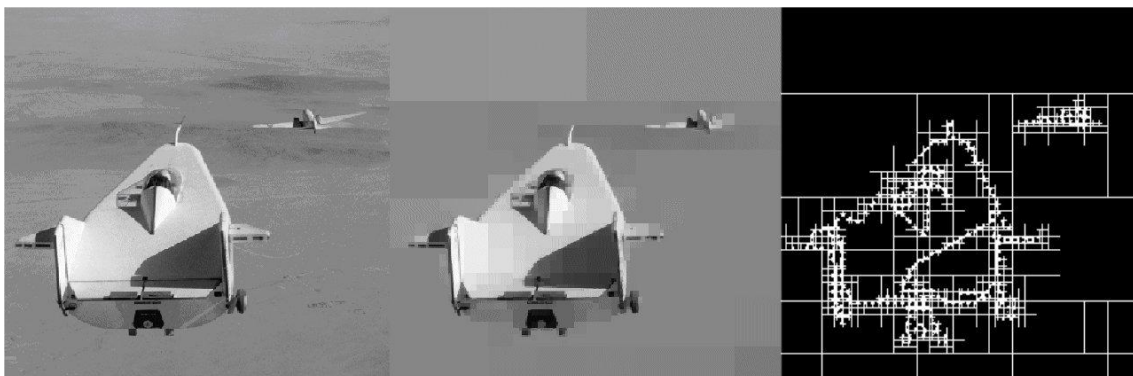


Figura 2.9. División de una imagen en árboles cuaternarios.

2.6.4 Representación y descripción

Durante el proceso de segmentación es habitual que el resultado de la delimitación de objetos no sea preciso debido a que se hayan agrupado de forma incorrecta los píxeles, se solapen regiones o los bordes sean imprecisos [PLA 06, ESC 01, PAJ 01].

Para poder llevar a cabo un procesamiento de alto nivel es necesario realizar una última etapa de procesado con el objetivo de realzar la geometría y forma de los objetos de la escena. Este tipo de procesamiento se conoce como procesamiento morfológico [PLA 06].

Puesto que las técnicas empleadas en el proceso de visión artificial son muy dependientes del contexto sobre el que se opere, el procesamiento morfológico puede ser empleado en fases anteriores o incluso posteriores en el ciclo de vida del reconocimiento visual, como en restauración de imágenes, detección de bordes, análisis de texturas, aumento de contraste y compresión de imágenes.

Entrando en la descripción de este conjunto de técnicas, el procesamiento morfológico se basa en las nociones básicas de la teoría de conjuntos para extraer estructuras geométricas de ellos. Las técnicas más habituales de este proceso son las siguientes [PLA 06]:

- *Erosión binaria*. Consiste en comprobar si un conjunto (normalmente la representación de una forma geométrica) está completamente incluido en la imagen.
- *Dilatación binaria*. Es la transformación dual a la erosión.
- *Apertura y cierre*. Utilizando los dos operadores anteriores se consiguen diseñar operaciones de realce.
- *Gradiente morfológico*. Se utiliza en la localización de bordes y perímetros de los objetos.
- *Top-hat*. Sirve para descubrir aquellas estructuras de la imagen que han sido eliminadas en el filtrado de apertura y cierre. La figura 2.10 muestra el resultado al aplicar top-hat.



Figura 2.10. Top-hat en niveles de grises.

Otra de la forma habitual de realizar el procesamiento morfológico es mediante las conocidas transformaciones geodésicas [PLA 06]. Este tipo de transformaciones se basa en el concepto de conectividad [PLA 06]. Dentro de este conjunto de técnicas podemos encontrar:

- Dilatación y erosión.
- Reconstrucción.
- Watershead.

2.6.5 Reconocimiento e interpretación

Durante esta última fase el objetivo es interpretar automáticamente la escena partiendo de los resultados obtenidos en las anteriores etapas [PLA 06, PAJ 01]. Para llevar a cabo este proceso, se trata de asociar a cada elemento segmentado con un conjunto de valores numéricos o de atributos, denominados vectores de características. Estos vectores sirven de entrada a los sistemas de clasificación de objetos, que les asignarán una etiqueta cualitativa. Por tanto, este proceso, tal y como se describirá en esta sección, consta de las siguientes tareas: en primer lugar se le asocia a cada objeto una etiqueta; una vez etiquetada la imagen se procede a extraer las características del objeto para, por último, asignarle a cada uno una etiqueta cualitativa.

Como se ha explicado, la primera tarea es la de asociar a cada objeto segmentado un valor numérico. Esta etapa se fundamenta en la continuidad de los objetos en el espacio, dada por la relación de conectividad de los píxeles adyacentes. La salida de esta fase es una matriz en la que a los píxeles de cada objeto se les asigna un valor numérico.

La siguiente fase es la de extraer de los objetos características cuantificables para obtener el vector de características. Las propiedades de los objetos pueden ser clasificadas en relación con el contorno de los objetos y con las características propias del conjunto del objeto en su totalidad.

Para poder realizar la cuantificación se utilizan los siguientes descriptores [PLA 06]:

- Descriptores de regiones.
- Descriptores topológicos.
- Texturas como se muestran en la figura 2.11.
- Descriptores de fronteras.



Figura 2.11. Imágenes con diferentes texturas.

La última tarea de la interpretación automática es el reconocimiento de patrones. Para realizar la clasificación es necesario disponer del vector de características y de un amplio conocimiento del universo del problema, ya que para poder asignar una etiqueta de clasificación a un objeto es necesario disponer de reglas de pertenencia a cada clase. Existen fundamentalmente tres métodos de clasificación [PLA 06]:

- Métodos sintácticos.
- Redes neuronales. Un ejemplo del uso de este tipo de técnicas se puede ver en la figura 2.12.

- Métodos estadísticos.



Figura 2.12. Ejemplo de clasificación mediante una red neuronal.

2.6.6 Una alternativa a las técnicas anteriores: similitud de imágenes

Antes de concluir con la descripción del reconocimiento automático de imágenes es necesario describir una técnica bastante más ligera en términos computacionales que las que han sido expuestas hasta el momento.

Debido a que uno de los requisitos fundamentales en aplicaciones en tiempo real sobre dispositivos con un hardware bastante limitado es que éstas se ejecuten en un tiempo lo más rápido posible, las técnicas que se han visto en este capítulo no suelen ser viables [CHO 09]. Una solución a este problema consiste en utilizar técnicas de similitud de imágenes. Para ello es necesario disponer de un número de muestras lo suficientemente alto del escenario sobre el que contrastar las imágenes que el dispositivo envía a un servidor dedicado a realizar este proceso si se trata de un dispositivo móvil con un hardware limitado, o bien desde el mismo terminal si sus prestaciones son las adecuadas.

En este proyecto, de hecho, se intentó utilizar este mecanismo, pero fue desechado debido a las limitaciones en cuanto al procesamiento de los dispositivos móviles, ya que la aplicación realiza todos sus cálculos en local.

Este tipo de técnicas son útiles cuando se dispone de una infraestructura de red capaz de computar a altísima velocidad mediante clústeres la similitud de imágenes.

2.7 Identificación por posicionamiento

La identificación de escenarios mediante posicionamiento se realiza gracias a la estimación en la posición y orientación del usuario, deduciendo de esta forma qué es lo que está viendo [AML 08, DEY 00].

En la mayoría de aplicaciones de realidad aumentada para teléfonos móviles ésta suele ser la solución más extendida [WIKTD 10, LAY 10]. Esto se debe a que las operaciones de cálculo necesarias para realizar reconocimiento de imágenes suelen ser bastante costosas en cuanto a recursos y tiempo, haciendo inviables las aplicaciones en tiempo real. Sin embargo, utilizando técnicas de posicionamiento se consigue descargar a los terminales de grandes operaciones de cálculo, delegando éstos a los servidores [WIKTD 10, LAY 10].

En aplicaciones más sofisticadas, como el sistema POLAR [OLW 05], se utilizan dispositivos de captura del movimiento del usuario, así como cámaras que detectan el movimiento ocular de éste. De esta manera se consigue también estimar la escena que el usuario está visualizando.

De nuevo se aprecia cómo no existe una técnica general para todos los sistemas de realidad aumentada, sino que, cómo se verá en un capítulo posterior, los requisitos dependen estrictamente de las necesidades del sistema.

A continuación veremos los distintos sistemas de posicionamiento que pueden ser utilizados en el proceso de identificación de escenas.

2.7.1 Sistema de posicionamiento global (GPS)

El sistema de posicionamiento global es un servicio de los EE.UU. que proporciona a los usuarios información sobre posicionamiento, navegación y cronometría [XU 07, GPS 10]. El sistema GPS está formado por tres segmentos, el espacial, el de control y el de usuario [XU 07, GPS 10]:

- El *segmento espacial* consiste en un grupo de veinticuatro satélites que envían información sobre la posición y hora de cada satélite.
- El *segmento de control* realiza las tareas de seguimiento y manejo de los satélites desde la tierra.
- El *segmento de usuario* se encuentra localizado en los terminales receptores GPS y determina la posición tridimensional y la hora precisa.



Figura 2.13. Satélite del segmento espacial del Sistema de Posicionamiento Global (GPS).

Para un dispositivo receptor GPS el funcionamiento es el siguiente. Cuando quiere determinar su posición, al menos debe recibir tres señales de tres de los veinticuatro satélites disponibles, que contendrán información sobre la posición del satélite y la hora a la que fue enviada la señal. Con estas señales el receptor sincroniza el reloj GPS y calcula el retraso de las señales, o lo que es lo mismo, la distancia al satélite. Mediante triangulación con las tres señales recibidas, se estima fácilmente la posición relativa respecto a los tres satélites. Gracias al envío de la posición absoluta de cada satélite, se calcula, una vez averiguada la posición relativa del receptor, su posición absoluta sobre el planeta.

Extendiendo la tecnología GPS a los sistemas de realidad aumentada, se puede apreciar claramente cómo, en entornos al aire libre y con ayuda de una simple brújula digital, esta

técnica es una solución fiable para determinar la escena del usuario. Sin embargo, el uso de GPS en sistemas de realidad aumentada en entornos cerrados no resulta aplicable debido al error de estimación que el propio sistema trae consigo. El error medio de estimación es del rango de dos metros y medio, por lo que se hace evidente como, en entornos cerrados en los que es necesaria generalmente una precisión que ronde los centímetros, esta técnica no puede llevarse a cabo y se suele optar por técnicas de reconocimiento de imágenes ligeras.

2.7.2 Uso de sensores de movimiento

Otra de las técnicas utilizadas para la estimación de la posición del dispositivo se realiza mediante el uso de sensores de movimiento [OLW 05]. Por sí sola, este tipo de técnica no es capaz de dar una posición global del dispositivo, aunque en ciertas ocasiones no tenga porqué ser necesario, sino que la aplicación sólo necesite conocer su posición relativa desde el momento que inicia su ciclo de vida. En este tipo de contexto, sensores de movimiento como acelerómetros o giroscopios aportan información manejable para estimar la posición local del dispositivo en cada momento.

2.8 Formatos de señales de vídeo e imágenes

En esta sección se explicarán los formatos empleados habitualmente en los datos multimedia, tanto en imágenes como en vídeo. Existen múltiples tipos de espacios de colores, pero en este documento sólo se abordarán los formatos RGB y YUV, que son aquellos que han sido objeto de estudio para poder llevar a cabo el proyecto.

2.8.1 Formato RGB

El modelo RGB define un espacio de color formado por tres componentes cromáticas [RGB 98]. Estas componentes son los tres colores primarios, el rojo, el verde y el azul. Mediante la unión de estos tres colores es posible formar cualquier color reconocible por el ojo humano.

Cada una de las tres componentes tiene un valor comprendido entre 0 y 255, siendo el blanco la unión de las tres componentes cuando éstas tienen como valor 255 o, lo que es lo mismo, cuando alcanzan su mayor señal de luminosidad, creando una saturación de colores que se transforma en el blanco; y siendo el negro la ausencia de intensidad en la luminosidad de estas tres componentes, es decir, cuando éstas valen 0. Por otro lado, cualquier color de la escala de grises comprendido entre 0 y 255 se forma cuando las tres componentes tienen el mismo valor. Así pues, cuando tanto la intensidad del rojo, como del verde, como del azul valen 25, se corresponderían con el valor 25 de la escala de grises.

En el ámbito del color en los ordenadores, habitualmente se encuentra una variación del RGB tradicional, el ARGB. Este espacio de color define una cuarta componente alfa (A) que especifica la presencia o no del canal transparente sobre el espacio de color.

2.8.2 Formato YUV

El modelo YUV define un espacio de color en tres componentes, una de luminancia (valores en escala de grises) y dos de crominancia [YUV 95, YUV 02]. El modelo YUV imita el modelo de visión humana. Este formato fue originalmente pensado para habilitar compatibilidad entre los televisores en blanco y negro y en color, ya que la señal en escala de grises se transmite por una frecuencia diferente a los valores de crominancia que definen el color de la escena. El espacio de color de la familia de los YUV para las imágenes digitales es el espacio YCbCr,

mientras que para las televisiones analógicas se utiliza el espacio YPbPr. En este documento nos centraremos en el espacio YCbCr que es el utilizado por las cámaras de vídeo de los teléfonos móviles de última generación.

Las siglas YCbCr significan, cada una:

- Y es la componente lumínica. Es un valor entre 0 y 255 que define la intensidad de la luz en cada punto o píxel de la escena. En términos de crominancia, este valor indica el tipo de gris del punto.
- Cb y Cr son las componentes cromáticas. Cb corresponde con la componente de color U del espacio YUV y Cr con la componente V.

De acuerdo al estándar, la conversión entre RGB y YCbCr se encuentra definida en la ecuación 2.2.

$$Y = K_{ry} \cdot R + K_{gy} \cdot G + K_{by} \cdot B$$

$$C_b = B - Y$$

$$C_r = R - Y$$

$$K_{ry} + K_{gy} + K_{by} = 1$$

Ecuación 2.2. Conversión entre espacio RGB y espacio YCbCr

La tabla 2.1 muestra los valores de los coeficientes según los diferentes estándares.

Estándar	K_{ry}	K_{by}
ITU601 / ITU-T 709 1250/50/2:1	0.299	0.114
ITU709 / ITU-T 709 1250/60/2:1	0.2126	0.0722

Tabla 2.1. Coeficientes para calcular la conversión entre RGB e YCbCr.

Cabe, por último, resaltar que cada componente Cb y Cr definen la intensidad cromática de dos valores Y, consiguiendo de esta forma una compresión mayor en los datos.

2.8.3 Formato YCbCr 420

Este tipo de espacio de color es el utilizado por el sistema operativo Android para codificar la señal de vídeo recibida por las cámaras de vídeo. Pertenecce a la familia de los YUV.

El formato de presentación de los datos de la señal viene codificado en un array de bytes en el cual, si la imagen tiene nxm píxels, los nxm primeros bytes contienen la componente Y de la escena, mientras que el resto de bytes contienen las componentes cromáticas Cb y Cr.

2.9 Plataformas móviles

En los últimos años se ha producido un notable crecimiento en el número de plataformas software pensadas en dar soporte a los dispositivos móviles, tales como teléfonos o PDAs, así como en las prestaciones que éstas aportan. Hoy en día, estas plataformas se aproximan más a los sistemas operativos tradicionales que a simples interfaces de acceso a los recursos de los dispositivos móviles. De hecho, la gran limitación que tienen estas plataformas se debe más al hardware al que dan soporte que a sus propias prestaciones.

2.9.1 Symbian OS

Symbian OS es un sistema operativo para terminales móviles [SYMB 10]. Este sistema da soporte a los teléfonos móviles de las empresas Nokia, Sony Ericsson y Motorola, entre otros.

Symbian es un sistema multi-tarea, que realiza protección de memoria. La aproximación a la multi-tarea está basada en un sistema de memoria virtual sobre eventos hacia un servidor asíncronos. De hecho, Symbian es un sistema operativo, en muchos aspectos, muy similar a cualquier otro que se pueda encontrar para ordenadores de escritorio.

El sistema operativo está compuesto de las siguientes capas, ordenadas desde más alto nivel a más bajo:

- UI Framework.
- Servicios de aplicación.
- Servicios OS.
- Servicios base.
- Servicios del Kernel e interfaz de hardware.

El lenguaje nativo de Symbian es C++, por lo que, aunque no sea estrictamente necesario, la práctica habitual es programar aplicaciones en ese mismo lenguaje de programación.

2.9.2 Windows Mobile

Windows Mobile es el sistema operativo para teléfonos móviles de Microsoft [MCPH 06]. Está basado en Windows CE 5.2, sistema optimizado para dispositivos con almacenamiento mínimo. Es un sistema en tiempo real que utiliza 256 niveles de prioridad. Su unidad fundamental es el hilo, lo que simplifica el interfaz de trabajo en tiempo de ejecución.

Al igual que ocurría con el sistema operativo Symbian, Windows Mobile está desarrollado en el lenguaje C++, por lo que, por norma general, el desarrollo de aplicaciones en esta plataforma se realizará también sobre ese mismo lenguaje. En la figura 2.14 se puede ver un escritorio típico de la plataforma Windows Mobile.



Figura 2.14. Captura de un dispositivo móvil que utiliza Windows Mobile.

2.9.3 iPhone OS

iPhone OS es el sistema operativo de los dispositivos móviles de Apple iPod Touch y iPhone [IPH DEV 10]. Está basado en una variante del Mach Kernel [SING 06] que se encuentra en Mac OS X, el sistema operativo de Apple para ordenadores de escritorio.

El sistema consta de cuatro capas de abstracción, que son:

- La capa del núcleo del sistema operativo.
- La capa de Servicios principales.
- La capa de Medios de comunicación.
- La capa de Cocoa Touch.

Para poder desarrollar aplicaciones para este tipo de dispositivos se ha de utilizar el iPhone SDK [IPH DEV 10], un kit de desarrollo específico para esta plataforma.

2.9.4 Android

Android es un sistema operativo basado en una modificación del Kernel de Linux para dispositivos móviles [AND SDK 10]. Aunque el desarrollo inicial fue llevado a cabo por la empresa Android Inc., actualmente la plataforma pertenece a Google.

Android permite el desarrollo de aplicaciones mediante una SDK en lenguaje Java y un kit nativo (NDK) para lenguaje C. Una de las características que hace de Android un entorno atractivo para los desarrolladores es que el código fuente se encuentra bajo licencias de software libre y código abierto.

La arquitectura de Android se puede ver en la figura 2.15 y está formada por los siguientes componentes:

- Aplicaciones.
- Framework de aplicaciones.

- Bibliotecas.
- Runtime de Android.
- Núcleo Linux.

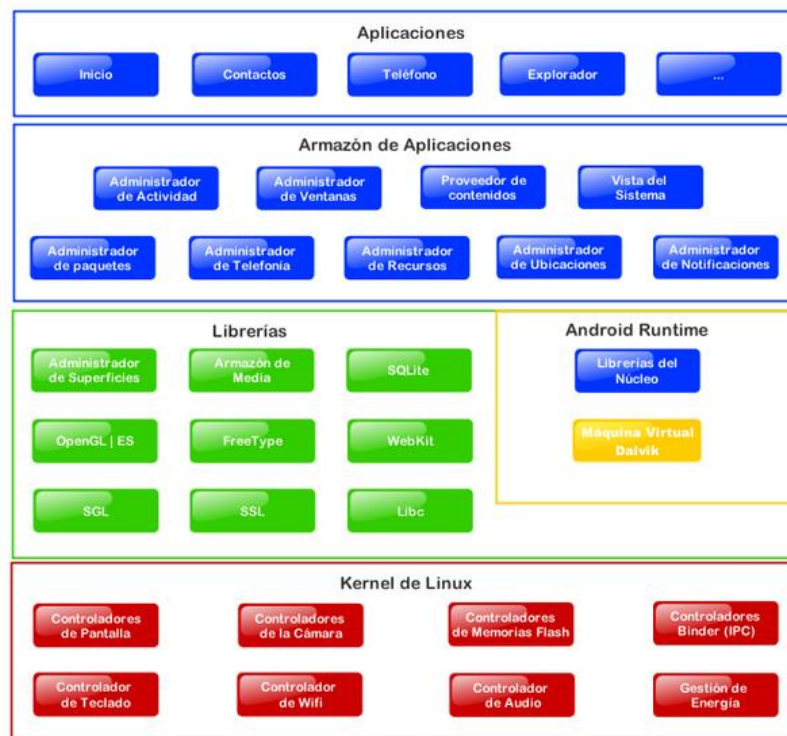


Figura 2.15. Arquitectura de Android.

3. ANÁLISIS DE SISTEMAS DE REALIDAD AUMENTADA

Este capítulo analiza algunos de los sistemas de realidad aumentada más destacados, incluyendo los componentes hardware y software de los mismos, así como sus arquitecturas.

En esta sección se definen los componentes hardware y software necesarios para la implantación de un sistema de realidad aumentada. Después se analizan algunos de los principales sistemas de realidad aumentada identificando los componentes hardware y software de los mismos, así como sus arquitecturas.

La primera sección identifica los componentes hardware y software que habitualmente son necesarios para implantar un sistema de realidad aumentada. En la segunda sección se traslada la identificación de la primera sección a un análisis de diferentes sistemas de realidad aumentada ya desarrollados, obteniendo información sobre el hardware y software empleado. La tercera y última sección define las diferencias entre un sistema autónomo y uno distribuido desde el punto de vista arquitectónico de los sistemas y aplicaciones de realidad aumentada. Las definiciones aportadas se acompañan de un análisis de qué arquitecturas han implantado los sistemas expuestos en la sección 2.

3.1 Componentes hardware y software

Esta sección identifica los componentes hardware y software necesarios para la implantación de un sistema de realidad aumentada. En este apartado, estos componentes sólo se identifican de manera genérica (p.e. librería gráfica).

3.1.1 Componentes Hardware

Como cualquier otro sistema informático, los sistemas de realidad aumentada requieren de componentes hardware sobre los que se vayan a llevar a cabo las acciones deseadas.

Los elementos básicos necesarios para implantar un sistema de realidad aumentada son similares a los de cualquier sistema informático tradicional. Así pues, son necesarios componentes de procesamiento, almacenamiento, comunicaciones, etc. De hecho, la diferencia en el equipamiento básico se encuentra en la finalidad del sistema.

En consecuencia, los elementos hardware necesarios para cualquier sistema de realidad aumentada son: una cámara, una pantalla de visualización o equipos holográficos en 2D o 3D, un equipo de procesamiento y cálculo, componentes de almacenamiento y, por último, dispositivos de transmisión y recepción de datos en caso de que sean necesarios. No hay que olvidar que las prestaciones de cada uno de ellos dependen fundamentalmente del tipo de sistema que se vaya a implantar. La tabla 1 [LOP 10] recoge un resumen de los componentes hardware habituales en un sistema de realidad aumentada.

Tabla 3.1. Resumen de los componentes hardware de un sistema de realidad aumentada.

Técnica	Componente hardware
Captura <i>video-through</i>	Todo tipo de cámaras de vídeo
Captura <i>see-through</i>	Cámaras integradas de vídeo, HMD (<i>Head Mounted Display</i>)
Reconocimiento geo-posicionamiento	Antena GPS, sensores de movimiento
Reconocimiento híbrido	Antena GPS, sensores de movimiento,

Tratamiento imágenes	dispositivos de captura de imágenes Procesadores con velocidad de procesamiento de, al menos, 2 GHz
Almacenamiento	Disco duro convencional
Comunicaciones locales	Tarjeta de red, conectores RJ45, antena wireless, punto de acceso
Comunicaciones móviles	Equipamiento GSM
Visualización <i>video-through</i>	Pantallas de vídeo, monitores, proyectores
Visualización <i>see-through</i>	HMD, teléfonos móviles

3.1.2 Componentes Software

Una vez descritos los componentes hardware se expondrá el software necesario para implantar un sistema de realidad aumentada. Al igual que sucede a la hora de elegir el hardware adecuado, la decisión de utilizar un software u otro viene determinado por las prestaciones y características del sistema.

En términos generales, un sistema de realidad aumentada necesita software de reconocimiento, ya sea por posicionamiento o por imágenes (con o sin marcadores), librerías de procesamiento de imágenes (si es que lo necesitase), al menos una base de datos, librerías de aumento de contenidos digitales y, si fuese necesario, software de comunicaciones. La tabla 2 [LOP 10] muestra un resumen del software genérico necesario para cada etapa.

Cabe destacar que la mayoría de librerías de procesamiento de imágenes, ARTToolkit [ART 10], Java3D [J3D 10], implementan tanto funciones de reconocimiento como de procesado y manipulación de contenidos visuales.

Tabla 3.2. Resumen de los requisitos software de un sistema de realidad aumentada.

Técnica	Componente software
Captura escena	Controladores de cámara de vídeo
Reconocimiento visual	Librerías de reconocimiento de imágenes
Reconocimiento geo-posicionamiento	Brújula digital, software GPS
Reconocimiento híbrido	Brújula digital, librerías de reconocimiento de imágenes, software GPS
Tratamiento imágenes	Librerías de tratamiento de imágenes
Almacenamiento	Base de datos
Comunicaciones locales	Controlador de red
Comunicaciones móviles	Controlador GSM
Visualización contenidos	Software de reproducción de contenidos multimedia, librerías de tratamiento de imágenes

3.2 Análisis de sistemas de realidad aumentada

En esta sección se expondrán una serie de proyectos de realidad aumentada que ya han sido o están siendo llevados a cabo con el fin de exponer la diferencia de arquitecturas y componentes empleados según cada situación.

3.2.1 ULTRA

El primer proyecto que se va a analizar es el sistema ULTRA [ULT 10]. El objetivo de este proyecto fue crear una plataforma de realidad aumentada ultra ligera y compacta que permitiese a los desarrolladores crear sus propios sistemas. La idea fue originalmente creada para que fuese utilizada en dispositivos móviles, en concreto para PDAs.

El reconocimiento del escenario se realiza mediante marcadores. Esto es posible debido a que la plataforma fue ideada para dar soporte a sistemas en entornos industriales y, por tanto, cerrados. Por ello, los requisitos aliviaron la carga del trabajo del sistema, ya que el reconocimiento por marcadores es un proceso más ligero. La plataforma fue pensada para que todo el proceso de realidad aumentada tuviera lugar en el terminal. Debido a esto, el hardware debe poseer una serie de propiedades predeterminadas. En primer lugar, es necesario que la PDA esté conectada a una cámara de 800x600 píxeles. En segundo lugar, debe proveer de aceleración de video y gráficos 3D para poder realizar las operaciones necesarias. La librería gráfica que se escogió fue OpenGL ES, por lo que fue necesario utilizar una implementación específica de EGL como interfaz entre OpenGL ES y la plataforma nativa del sistema de ventanas de la PDA.

Por otro lado, ULTRA permite la creación de contenidos off-line desde un servidor remoto. Por ello, se requieren dos requisitos hardware más. Primero, el dispositivo móvil debe poder establecer comunicaciones W-LAN para comunicarse con el servidor. Segundo, es necesario disponer de un equipo que actúe como servidor en este caso.

Por último, para poder llevar a cabo la función de creación de contenidos es necesario disponer de un experto que se encargue de realizar las imágenes 2D y 3D, texto o contenidos multimedia que se vayan actualizando en el sistema. El motivo de esta citación es que hasta ahora no se había comentado en ningún momento nada sobre los requisitos de recursos humanos necesarios en un proyecto de realidad aumentada.

Tabla 3.3. Resumen de las técnicas empleadas en ULTRA.

Técnica	Componente software	Componente hardware
Reconocimiento marcadores	OpenGL ES	Cámara de teléfono móvil o PDA
Tratamiento imágenes	OpenGL ES	Procesador y memoria del teléfono móvil o PDA
Visualización	Delegado en software del dispositivo	Pantalla del teléfono móvil o PDA

3.2.2 MARCH

MARCH es un proyecto que permite la visualización de contenidos digitales sobre grabados de cuevas prehistóricas [CHO 09]. Este proyecto fue creado para que funcionase en terminales móviles. En la figura 3.2 se puede ver una representación del funcionamiento del sistema.

El primer requisito que aparece es la del reconocimiento de escenas mediante marcadores. Las razones del uso de este mecanismo son dos: poder utilizar el sistema en cuevas reales y aligerar la carga de procesamiento y cálculo del terminal.

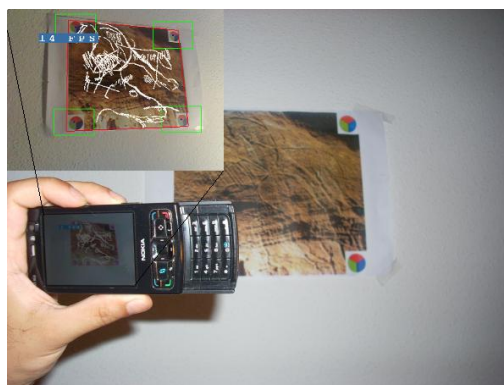


Figura 3.1. Funcionamiento del sistema MARCH.

El sistema consta de dos partes, una off-line y la aplicación en el móvil. La parte off-line se encarga de registrar las imágenes que serán aumentadas en el terminal. Este proceso consiste en añadir sobre la imagen del grabado la interpretación del experto. Una vez realizado este proceso se procede a extraer en una capa de fondo transparente la interpretación del experto. Un requisito por tanto de este sistema es contar con un experto sobre interpretaciones prehistóricas.

Por último, existe un requisito que restringe enormemente las posibilidades de esta aplicación. El sistema utiliza la librería gráfica NokiaCV [NOK CV 10], por lo que sólo los terminales Nokia que implementen esta API pueden soportar la ejecución de esta aplicación.

Tabla 3.4. Resumen de las técnicas empleadas en MARCH.

Técnica	Componente software	Componente hardware
Reconocimiento marcadores	Nokia CV	Cámara de teléfono móvil o PDA
Tratamiento imágenes	Nokia CV	Procesador y memoria del teléfono móvil o PDA
Visualización	Delegado en software del dispositivo	Pantalla del teléfono móvil o PDA

3.2.3 ASTOR

ASTOR es un prototipo de sistema de realidad aumentada en el que la información digital se muestra en unas pantallas transparentes sin que el usuario tenga que utilizar ningún dispositivo [OLW 05].

El sistema utiliza representación holográfica para mostrar la información aumentada al usuario. Para ello se sirve de dos o más proyectores que reflejan la imagen sobre elementos holográficos ópticos (HOE) transparentes tal y como se aprecia en la figura 3.

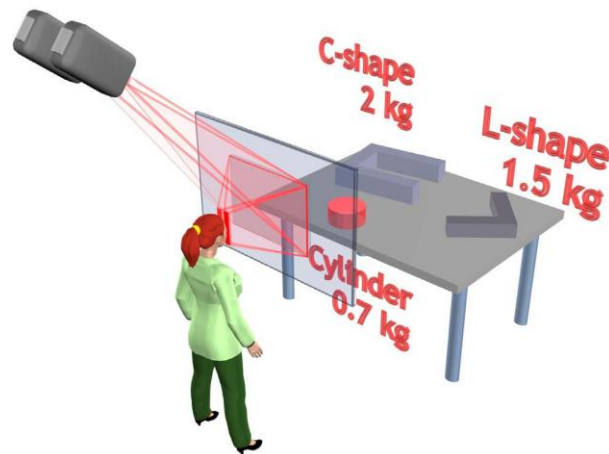


Figura 3.2. Representación del sistema de visualización del sistema ASTOR.

Para hacer posible este tipo de realidad aumentada, el sistema requiere de una serie de elementos, tanto hardware como software y de otra índole.

En primer lugar, el sistema ASTOR necesita unos HOE transparentes sobre los que representar la imagen holográfica. Los HOE son unos platos de cristal fino o incluso unas láminas de plástico transparente que pueden ser acopladas sobre otras superficies, como puede ser una ventana o un expositor.

En segundo lugar, ASTOR está pensado para trabajar en entornos pequeños, aunque podría operar en entornos más grandes, aumentando considerablemente los costes.

En tercer lugar, puesto que el usuario no requiere de ningún dispositivo, todo el hardware se centra en el lado del proveedor. Al ser todavía un prototipo los gráficos no son excesivamente complejos, por lo que no se requiere de un dispositivo especialmente potente. El prototipo ha sido desarrollado utilizando la tarjeta gráfica NVIDIA GeForce4 MX 440-SE y 256 MB de RAM. También necesitaron de proyectores con control remoto para poder guiarlos hacia la escena correctamente.

En el apartado software emplearon la librería gráfica Java3D [J3D 10] y JRMII [JRMII 10] para las comunicaciones cliente-servidor.

Tabla 3.5. Resumen de las técnicas empleadas en ASTOR.

Técnica	Componente software	Componente hardware
Reconocimiento imágenes	Java3D	Cámara de vídeo NVIDIA GeForce MX 440-SE, 256 MB de RAM
Tratamiento imágenes	Java3D	
Comunicaciones	JRMII	Arquitectura cliente-servidor
Visualización	Delegado en software del dispositivo	Proyectores, pantallas HOE

3.2.4 POLAR

POLAR es un sistema de realidad aumentada de bajo coste que permite superponer información a una escena real mediante un mecanismo diferente a los expuestos en este documento [OW 05].

El sistema utiliza un espejo medio-plateado que refleja, por un lado los objetos reales que se desean ampliar, y por otro lado la información digital que se va a mostrar. Para ello juega con los diferentes ángulos de visión del espejo, creando el efecto de superposición de las imágenes virtuales a la escena real. Una representación del funcionamiento se puede ver en la figura 4.

Los requisitos hardware son bastante limitados, siendo necesario para llevar a cabo su implantación de los siguientes elementos: un espejo medio-plateado, un ordenador donde se llevan a cabo los cálculos, una pantalla convencional desde la que se genera la imagen virtual que se refleja en el espejo y una cámara y dispositivo de detección de distancia que detecta el movimiento de la vista del usuario para cambiar el enfoque. Además, se puede añadir dispositivos de audio para mejorar la experiencia del usuario.

Desde el punto de vista del software requerido, el sistema necesita de un módulo de comunicación entre los dispositivos y el PC. En el apartado gráfico se emplea la librería OpenGL ES para llevar a cabo las operaciones pertinentes.

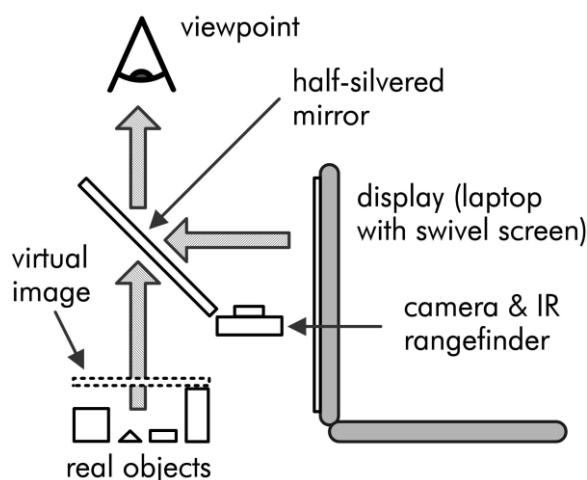


Figura 3.3. Vista general del funcionamiento del sistema POLAR.

Tabla 3.6. Resumen de las técnicas empleadas en POLAR.

Técnica	Componente software	Componente hardware
Reconocimiento imágenes	OpenGL ES	WebCam, sensor de movimiento
Tratamiento imágenes	OpenGL ES	Procesador y memoria del PC
Visualización	Delegado en software del dispositivo	Pantalla del PC, espejo medio reflectante

3.2.5 Layar

Layar es una aplicación para Android que permite la visualización de información digital sobre un entorno real en teléfonos móviles [LAY 10].

El funcionamiento del sistema se explica a continuación. El terminal muestra la escena real mediante su cámara al usuario. Mediante el GPS del móvil, se envía a un servidor la localización del usuario junto a la orientación extraída con la brújula del mismo. Con estos datos, el servidor responde con una capa con la imagen o el texto del lugar que se está enfocando con el teléfono. La figura 5 esquematiza el funcionamiento del sistema.

Para analizar los requisitos para implantar este sistema se hará una división entre los requisitos del cliente o usuario y los requisitos del proveedor.

En primer lugar se explican los requisitos por parte del cliente. Para que el cliente pueda usar la aplicación es necesario que disponga de un teléfono móvil que utilice Android y que tenga una cámara de video integrada en el móvil. Por lo general, todos los teléfonos móviles de última generación disponen de una. Por último, es también indispensable que el terminal tenga localización GPS.

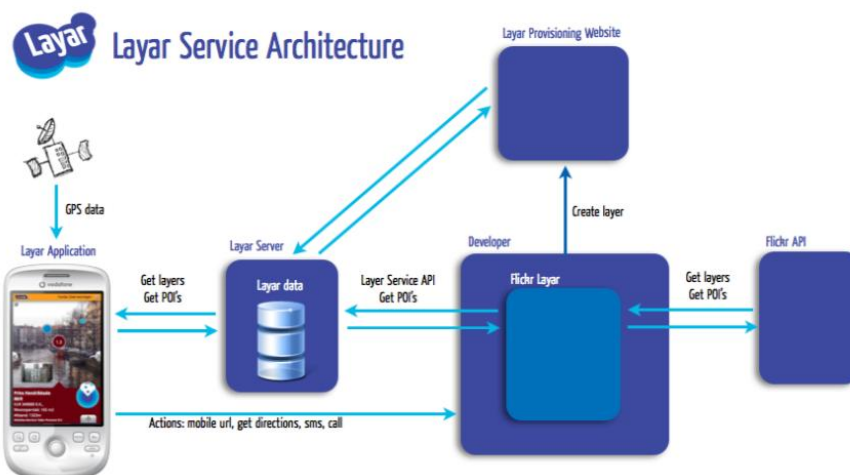


Figura 3.4. Arquitectura del sistema Laya.

En el apartado de requisitos por parte del proveedor existe una enorme diferencia. Para poder dar este tipo de servicio a los usuarios es necesario disponer de un servidor que reciba las peticiones de los terminales y les envíe la información que deberá ser ampliada en el mismo. Por otro lado, hace falta una base de datos que almacene las diferentes capas junto a sus respectivas localizaciones. Esto implica, como en cualquier arquitectura de este tipo, un nuevo requisito, que es el de personal de mantenimiento y actualización de los datos.

Tabla 3.7. Resumen de las técnicas empleadas en Laya.

Técnica	Componente software	Componente hardware
Reconocimiento geo-posicionamiento	Brújula digital del dispositivo, software GPS	Cámara de teléfono móvil o PDA, antena GPS
Almacenamiento de imágenes	Gestor de base de datos	Servidor de base de datos
Visualización	Delegado en software del dispositivo	Pantalla del teléfono móvil o PDA

3.2.6 LOCUS

Es un sistema de realidad aumentada desarrollado para terminales móviles, especialmente para PDAs y teléfonos 3G [LOC 10]. El usuario puede ver información ampliada sobre determinados lugares con su dispositivo gracias a las técnicas de geo-posicionamiento del sistema.

Para hacerlo posible, han desarrollado modelos 3D para aumentar la escena real. Por otro lado, otra de las tareas que se propusieron fue la de crear herramientas con las que pudieran caracterizar el contexto espacial en el que los usuarios iban a requerir realidad aumentada.

Los requisitos del sistema eran los siguientes: almacenar y gestionar la información espacial y posicional de los lugares; disponer de herramientas de modelado urbano con las que pudieran realizar las imágenes en 3D; poder utilizar herramientas de navegación y estimación de rutas; proveer de un sistema de posicionamiento del dispositivo móvil; poseer un interfaz para trabajar con realidad virtual y otro para realidad aumentada.

Tabla 3.8. Resumen de las técnicas empleadas en Locus.

Técnica	Componente software	Componente hardware
Reconocimiento geo-posicionamiento	Brújula digital del dispositivo, software GPS	Cámara de teléfono móvil o PDA, antena GPS
Visualización	Delegado en software del dispositivo	Pantalla del teléfono móvil o PDA

Una vez analizados diferentes proyectos y sistemas de realidad aumentada se ofrecerá, a modo de resumen, una serie de componentes hardware y software específicos habitualmente utilizados en este tipo de sistemas.

Dentro de los componentes hardware encontramos:

- Dispositivos de captura de imágenes: HMD modelo CyberMind, Visette45 SXGA; WebCam SPC 1300 NC.
- Procesamiento y reconocimiento de imágenes: Intel Core 2 Duo 2.33 GHz, 2 Gb RAM, NVIDIA GeForce MX 440-SE.

El resto de hardware no será citado, debido a que o bien se encuentra integrado en los dispositivos (caso de los componentes de comunicación GPS en los teléfonos móviles) o porque el modelo no es relevante (como sucede con el disco duro, pantalla de ordenador, etc.).

Desde el punto de vista software, se pueden encontrar los siguientes modelos:

- Procesamiento y reconocimiento de imágenes: ARToolkit, Java3D.
- Almacenamiento: PostgreSQL, MySQL.

El resto de componentes no serán descritos por los mismos motivos que en el análisis del hardware.

3.3 Arquitecturas de sistemas de realidad aumentada

Los sistemas de realidad aumentada son sistemas informáticos en los que existe una interacción entre el sistema y el usuario. Como en cualquier otro sistema de estas características, el tiempo de respuesta es un componente crítico que determina en gran medida el éxito o fracaso de una aplicación. En función de las prestaciones del sistema y los componentes hardware disponibles y la complejidad de procesamiento de sus procesos, es normal encontrar dos tipos de arquitecturas básicas [LOP 10]: los sistemas autónomos y los sistemas distribuidos. En esta sección se explicarán sin entrar en demasiado detalle estos dos conceptos.

Los *sistemas autónomos* son aquellos en los que la totalidad de las operaciones y tareas del sistema se llevan a cabo en un terminal único tal y como se muestra en la Figura 3.1. Este tipo de aplicaciones suelen ejecutarse en entornos muy limitados en cuanto a espacio y cuya información cabe en un terminal. También puede darse el caso de que el número de muestras a aumentar y la información que se añade sea suficientemente pequeña para poder ser procesada y mostrada en un solo dispositivo. Dentro de este grupo se pueden encontrar desde ordenadores convencionales conectados con una webcam, hasta teléfonos móviles de última generación, pasando por equipos que están compuestos por dispositivos de visualización y procesamiento conectados entre sí. Por lo general, este tipo de sistemas no requieren de equipamiento demasiado costoso debido a lo limitado de sus prestaciones, pudiendo llevarse a cabo en ordenadores domésticos. Este último caso suele ser más propenso a ser utilizado en entornos de prueba o en primeras fases del proyecto.



Figura 3.5. Imagen del videojuego Invizimals [INVZ 10] que utiliza técnicas de realidad aumentada en un solo dispositivo, una PSP.

Los *sistemas distribuidos* son aquellos en los que se delega en un equipo denominado servidor la tarea de realizar la mayor parte de los cálculos, rebajando en los terminales de visualización la carga de procesamiento. Habitualmente en este tipo de arquitectura los terminales se encargan de la recepción de la escena por cualquiera de los métodos ya expuestos en la sección primera de este documento y de la visualización de la realidad aumentada. El proceso intermedio se realiza en el servidor, aunque existen excepciones en las cuales parte de este trabajo se lleva a cabo en los propios terminales. Esta responsabilidad depende también de la aplicación que se está llevando a cabo, así como de las posibilidades del terminal. Por ejemplo, en aplicaciones implantadas en terminales móviles de última generación una posible estrategia es la de realizar parte del procesado de la imagen como detección de bordes mediante transformadas de Hough o métodos más ligeros e imprecisos. De esta forma, se envía al servidor los datos sobre los bordes de la imagen, evitando sobrecargas en la red. No obstante, no suele ser lo habitual por lo que se suele enviar la

imagen al completo o comprimida al servidor. Por otra parte, aquellas aplicaciones que envíen datos al servidor de forma inalámbrica requieren de dispositivos de envío y recepción de señales. En entornos cerrados lo habitual suele ser utilizar conexiones W-LAN, por lo que los terminales necesitan antenas de transmisión y recepción y en el lado del servidor hace falta un punto de acceso. En aplicaciones para teléfonos móviles se requieren de equipos de transmisión y recepción GSM en ambos lados.

A continuación se analizan los sistemas de realidad aumentada desde un punto de vista arquitectónico.

3.3.1 Arquitecturas en sistemas autónomos de realidad aumentada

En esta sección se muestran las conclusiones alcanzadas tras analizar la arquitectura interna de las aplicaciones: ULTRA [ULT 10], MARCH [CHO 09], POLAR [OW 05] y ARTToolKit [ART 10].

Este tipo de aplicaciones coincidían todas en que el reconocimiento está basado en la detección visual de algún elemento significativo. Puesto que las librerías de reconocimiento y tratamiento de imágenes han sido estudiadas en el capítulo 3, en esta sección nos centraremos en la perspectiva funcional de cada uno de los módulos que componen este tipo de aplicaciones.

La arquitectura interna de este tipo de aplicaciones suele estar compuesto por, al menos, los siguientes módulos:

- Captación de escenario.
- Tratamiento de imágenes.
- Reconocimiento visual.
- Matching de patrones.
- Mezclado de realidad y aumento.
- Visualización.

La figura 3.6 muestra una arquitectura interna típica de una aplicación de realidad aumentada que funciona mediante reconocimiento visual.

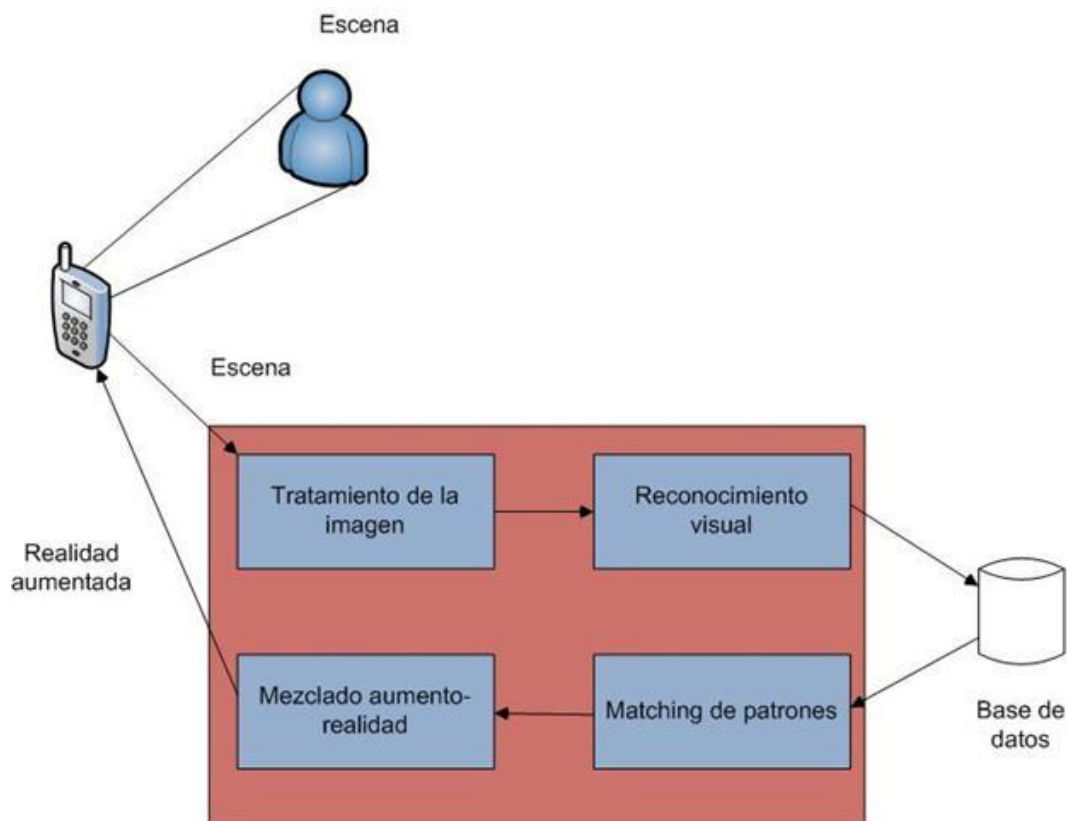


Figura 3.6. Arquitectura típica de una aplicación de realidad aumentada con reconocimiento visual.

Para terminar, cabe destacar que si no se describe la arquitectura de una aplicación con reconocimiento por geo-posicionamiento es porque, a nuestro juicio, no tiene sentido implantar una aplicación local con estas características. Las razones de estas conclusiones son básicamente dos:

- La cantidad de información sobre lugares suele ser bastante amplia, siendo necesario el uso de bases de datos clusterizadas.
- La información sobre lugares suele ser cambiante. Por ejemplo, una aplicación podría tener configurado que en cierto punto del mapa hay un restaurante que por algún motivo ha dejado de existir, ofreciendo información falsa.

3.3.2 Arquitecturas en sistemas distribuidos de realidad aumentada

En esta sección se analizarán las arquitecturas de los sistemas distribuidos considerados en esta sección.

ASTOR

El proyecto ASTOR consiste en un sistema distribuido en el que contribuyen al funcionamiento de éste varios elementos. Estos elementos son:

- Cámaras de vídeo. Se encargan de percibir la escena y la posición del usuario.
- Un equipo de procesamiento. Su función es la de realizar las tareas de tratamiento de imágenes. Actúa de servidor.
- Módulo de visualización. Compuesto por una serie de proyectores y pantallas HOE.

La figura 3.7 muestra la arquitectura de este sistema.

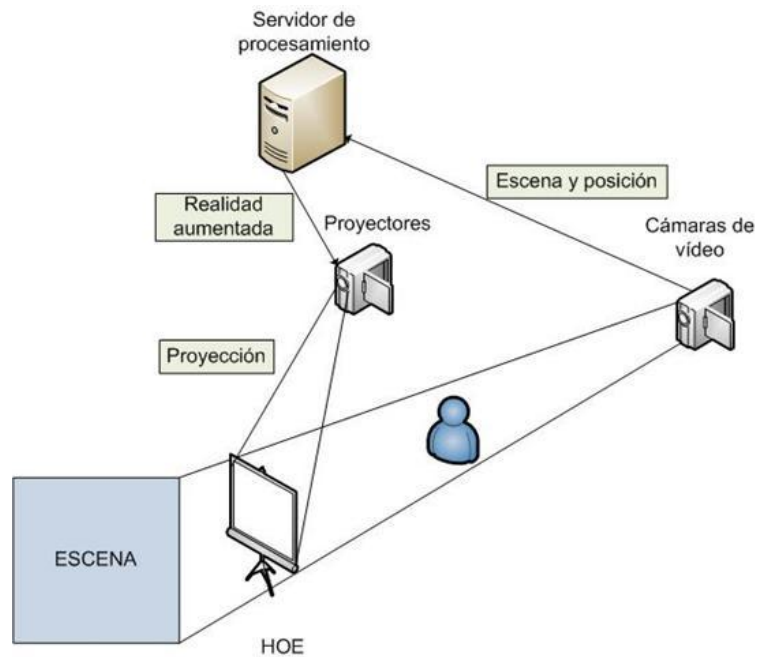


Figura 3.7. Arquitectura del sistema ASTOR.

LAYAR

El sistema LAYAR tiene una naturaleza distribuida en la que se pueden añadir contenidos o Layares al sistema, haciéndolo todo lo escalable que se desee.

Para su funcionamiento, LAYAR basa las tareas de reconocimiento por geo-posicionamiento en el teléfono móvil, delegando la elección de la capa o Laya de superposición a los servidores dedicados.

La figura 3.8 muestra la arquitectura del sistema LAYAR.

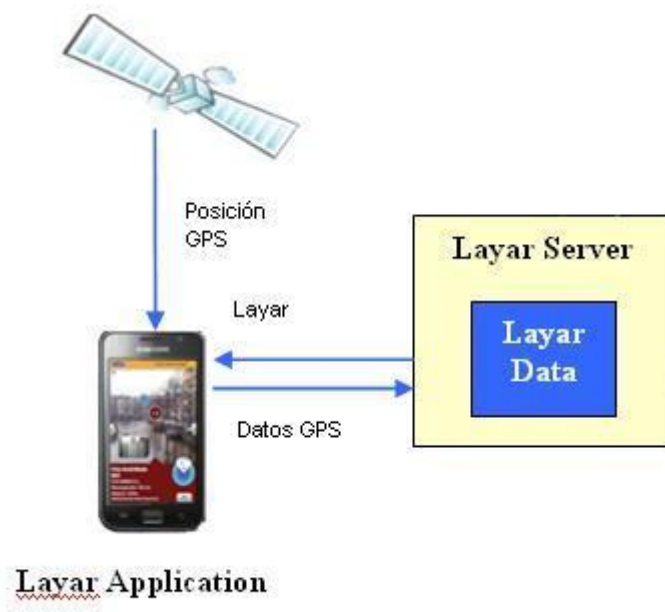


Figura 3.8. Arquitectura del sistema LAYAR.

LOCUS

El último sistema cuya arquitectura se va a analizar es el sistema LOCUS. Este sistema tiene dos componentes:

- La aplicación del teléfono móvil. Se encarga de recibir las señales GPS y de superponer la información.
- Un sistema de almacenamiento de datos sobre los lugares que se pueden aumentar.

Puesto que el funcionamiento de este sistema es muy similar al del sistema LAYAR, su arquitectura es también bastante parecida.

4. IMPLANTACION DE UN SISTEMA DE REALIDAD AUMENTADA

Este capítulo propone las arquitecturas para proveedores y clientes de realidad aumentada y lleva a cabo un estudio sobre las ventajas e inconvenientes de arquitecturas autónomas y distribuidas. Finalmente propone una arquitectura que ha sido implantada como prototipo en un desarrollo de Telefónica I+D.

La primera sección define las características de las arquitecturas para aquéllos proveedores que deseen implantar un sistema o aplicación de realidad aumentada. Por su parte, en la segunda sección se definen los requisitos desde el punto de vista del cliente para utilizar este tipo de sistemas. En la tercera sección se expone una discusión sobre la diferencia entre implantar un sistema autónomo contra uno distribuido, exponiendo aquellos escenarios en los que cada configuración resulta ser la más adecuada. Por último, en la cuarta sección se trasladan los conceptos introducidos a lo largo del capítulo a un escenario específico, el del sistema de domótica desarrollado para Telefónica I+D.

4.1 Arquitecturas para proveedores de realidad aumentada

Por lo general, como se vio en el capítulo anterior de este documento, existen dos tipos de sistemas de realidad aumentada, los autónomos y los distribuidos. En esta sección se expondrá la arquitectura que debe utilizar los proveedores de los sistemas de realidad aumentada en función del tipo de sistema que vaya a implantar.

En el caso de los sistemas autónomos los proveedores no participan en el ciclo de vida de la aplicación salvo en su creación y, quizás, en su mantenimiento. En este tipo de aplicaciones, por tanto, los requisitos implicados sólo tendrán que ver con estas tareas.

Durante la creación de la aplicación, el proveedor debe de disponer de los elementos, o en su defecto de una simulación de éstos, sobre los que dará el servicio. Por tanto, los requisitos a este nivel son tanto hardware como software. Desde el punto de vista del hardware, el proveedor deberá tener a su disposición dispositivos de captura de imágenes o de geoposicionamiento en función del tipo de identificación que quiera realizar y de visualización sobre el que mostrar los resultados de las diferentes implementaciones que se vayan haciendo del sistema. Desde el punto de vista del software necesario, deberá disponer de las librerías de reconocimiento adecuadas, así como de, al menos, una base de datos de muestras y software de procesamiento y visualización de los contenidos.

En la fase de mantenimiento, si es que la hubiese, el proveedor debe poder dar soporte a los usuarios de sus sistemas ante imprevistos que pudieran surgir. Los requisitos de esta fase no difieren de los necesarios en cualquier otro tipo de aplicación. Puesto que ya existe mucha literatura al respecto no será objetivo de este documento describir este proceso.

Por su parte, los sistemas distribuidos, como ya hemos comentado, son aquellos que delegan parte del trabajo en el usuario y el resto en el proveedor. En los sistemas distribuidos de realidad aumentada, y sobre todo en aquellos para dispositivos móviles, esto se traduce a que desde el terminal se realizan las tareas de captación de la escena y de presentación de resultados, mientras que en los servidores del proveedor se lleva a cabo el resto de procesamiento, desde identificación de escenarios a composición de imágenes. Debido a esta descarga de trabajo sobre los terminales de los usuarios, los equipos del proveedor deben ser más potentes. Los requisitos se hacen, por tanto, evidentes. El proveedor deberá disponer, en primer lugar, de una red de comunicación entre sus sistemas y los usuarios. En segundo lugar, necesita una arquitectura de servidores que se encarguen de recibir peticiones y procesarlas o

encaminarlas a otra arquitectura para que lleve a cabo este proceso. Cuando el número de usuarios es elevado es necesario disponer de un clúster acorde a los requerimientos de servicio. Por otro lado, si la aplicación requiere gran cálculo y tiempos de respuesta muy bajos, también será necesario disponer de un clúster dedicado sólo al procesamiento de resultados. Por último, destacar que en los sistemas distribuidos también están involucrados los requisitos derivados de las fases de creación y mantenimiento que han sido descritos al hablar de los sistemas empotrados.

4.2 Arquitecturas para clientes de realidad aumentada

En esta sección se introducirán los componentes involucrados en la arquitectura que deben correr por cuenta de los clientes de los sistemas de realidad aumentada. Al igual que sucedió al describir los requisitos por parte de los proveedores, al hablar de los requisitos de los clientes es necesario realizar una distinción entre los sistemas autónomos y los sistemas distribuidos.

En los sistemas autónomos todo el proceso se lleva a cabo en el terminal. Por decirlo de alguna manera, se podría decir que el dispositivo se encuentra aislado del resto de dispositivos. Según la funcionalidad de la aplicación, el dispositivo necesitará cumplir unas características u otras. Si la identificación de escenarios se realiza mediante reconocimiento de imágenes el cliente deberá disponer de una cámara que capte la escena y de hardware de procesamiento lo suficientemente potente para ejecutar la aplicación de la mejor forma posible. Si, por el contrario, la identificación de escenario se realiza mediante técnicas de geoposicionamiento, el cliente deberá disponer del adecuado hardware y software de comunicación GPS. No obstante, si utiliza este último mecanismo, el hardware de procesamiento no tiene porqué ser tan potente. Por otro lado, puesto que el terminal se encuentra aislado del resto de dispositivos, las muestras e información relativas al entorno sobre el que opera la aplicación deben estar alojadas en el terminal. Esto implica disponer de una base de datos con toda esa información así como de software de acceso y contraste de resultados. En último lugar, el cliente debe disponer de hardware y software adecuados para poder visualizar los contenidos reales mezclados con la información virtual, tales como pantallas de ordenador, proyectores, etc.

Como ya se ha descrito anteriormente, los sistemas distribuidos delegan parte del trabajo en el terminal del cliente y el resto en los servidores del proveedor. Puesto que, típicamente, en el cliente se llevan a cabo sólo los procesos de captación de escena y de visualización, los elementos necesarios son los dispositivos de captura, tales como cámaras de vídeo, Webcams o receptores GPS y pantallas de visualización o cualquier otro de los mecanismos expuestos en este documento. Por último, puesto que tiene que realizar comunicaciones con el proveedor, deberá disponer tanto de hardware, software y servicios que le permitan tener acceso a dichos servidores.

4.3 Sistemas autónomos frente a sistemas distribuidos

En el presente apartado se discutirá en qué situaciones es recomendable la implantación de un sistema de un sistema autónomo o uno distribuido. La figura 4.1 muestra la arquitectura de sistema autónomo.

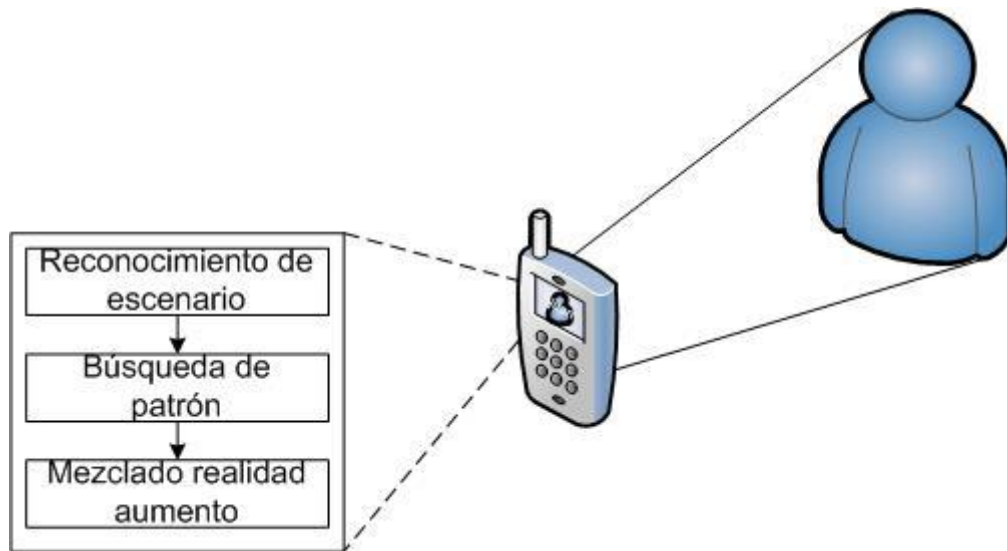


Figura 4.1. Arquitectura de un sistema autónomo..

La figura 4.2 muestra la arquitectura de un de sistema distribuido.

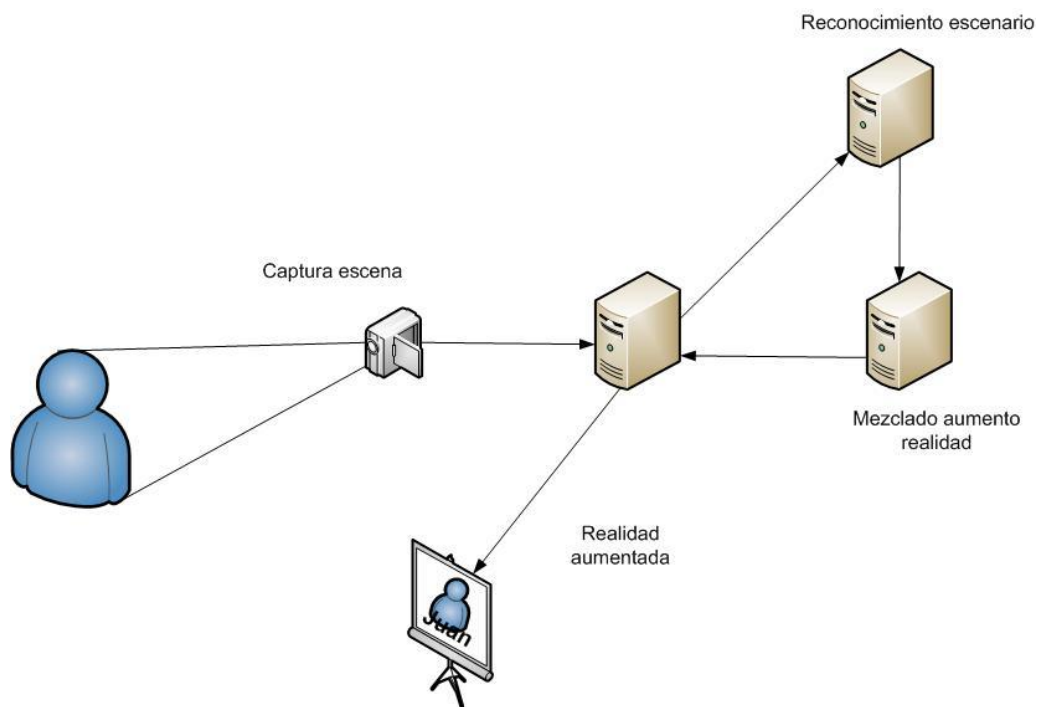


Figura 4.2. Arquitectura de sistema distribuido.

Una vez definido qué es una aplicación y qué es un sistema en realidad aumentada se procederá a describir en qué situaciones es recomendable la implantación de cada opción.

4.3.1 Características de un sistema autónomo

Como se ha ido viendo, un sistema autónomo vive aislado de agentes externos. Esta característica permite que pueda ser utilizado en cualquier momento y lugar donde tenga sentido utilizarlo, ya que no necesita de funcionalidades externas.

El hecho de que no requiera de agentes externos le permite garantizar el servicio ante problemas externos como pudieran ser problemas en la red, datos corruptos, etc.

No obstante, el tener su ciclo de vida local y no disponer de comunicación con otras entidades acarrea una serie de restricciones sobre su uso. Uno de los principales problemas con qué se encuentran los sistemas autónomos es la falta de monitorización por parte de agentes dedicados, siendo más complicado y molesto establecer un mecanismo de control de errores y mantenimiento que pudieran ser necesarios en determinados momentos.

Por otro lado, el encontrarse aislado puede ser tanto una ventaja como un inconveniente en cuanto a prestaciones. La forma en que influya dependerá, en gran medida, de la complejidad de sus cálculos y de la disponibilidad y cercanía de entidades externas que pudieran ayudar en el procesamiento. Para aplicaciones en las que se requieran gran cantidad de cálculo, quizá el tener que realizarlos en local pueda afectar de forma notable al rendimiento y tiempo de respuesta de la aplicación que, contando con agentes externos que paralelizasen el procesamiento, podría verse mermado. Por el contrario, si el número de cálculos es bajo, realizar una inversión en entidades externas podría suponer un gasto ineficaz. La decisión depende de las prestaciones del equipo donde vaya a correr la aplicación y de la complejidad de los cálculos que se vayan a llevar a cabo.

Por último, destacar que un sistema autónomo nunca afectará a otros equipos, pues no existe comunicación con ellos, por lo que esta situación podría llegar a ser una ventaja en ciertas ocasiones.

4.3.2 Características de un sistema distribuido de realidad aumentada

Por su parte, un sistema distribuido de realidad aumentada está compuesto por diversos agentes funcionales externos que realizan una serie de tareas de forma conjunta.

Debido a su naturaleza distribuida, los sistemas distribuidos de realidad permiten una mayor complejidad en los cálculos, reflejada en la capacidad de procesamiento o en una mejor experiencia del usuario. Gracias a las mayores velocidades en la transmisión de datos, los sistemas distribuidos de realidad aumentada han sufrido en los últimos tiempos un mayor impacto en los usuarios, llegando a los teléfonos móviles. Un claro reflejo de esta circunstancia son los sistemas Layar [LAY 10] o Google Goggles [GOGG 10].

El primero de ellos, Layar, que ya se vio en un capítulo anterior, delega la selección de la capa a mostrar a un servidor externo, siendo tarea de la aplicación móvil la extracción de la localización del mismo, así como de mezclar realidad y aumento. En la figura 4.3 se puede ver un ejemplo del funcionamiento del sistema Layar.



Figura 4.3. Superposición de contenidos en Layar.

Por su parte, Google Goggles es un sistema distribuido de reconocimiento visual de objetos. Para que funcione, la aplicación móvil se encarga de la captura de la escena y del mezclado de información, mientras que la tarea de reconocimiento se lleva a cabo en los servidores de Google, dando un servicio rápido y espectacular de reconocimiento de objetos. La figura 4.4 muestra un ejemplo del funcionamiento de Google Goggles.



Figura 4.4. El sistema Google Goggles en proceso de reconocimiento.

De estos dos sistemas se pueden extraer dos características de los sistemas distribuidos de realidad aumentada. En primer lugar los sistemas distribuidos de realidad aumentada permiten un mayor número de contenidos que los que permitiría un sistema autónomo. Esto se debe a que este tipo de arquitecturas distribuidas son escalables, pudiendo aumentar el número de equipos conectados a ella que pueden alojar y servir contenidos multimedia.

En segundo lugar, las arquitecturas distribuidas de este tipo de sistemas permiten utilizar un mayor número de equipos de procesamiento, siendo posible realizar tareas de alto coste computacional como es el caso del reconocimiento visual en tiempos mínimos.

Por otro lado, los sistemas distribuidos de realidad aumentada permiten tener equipos dedicados a tareas específicas. Esto puede suponer una mejora significativa en, por ejemplo, la experiencia del usuario frente al uso del sistema. Claro ejemplo de este hecho son los sistemas holográficos de tres dimensiones, que utilizan proyectores dedicados a la tarea de reconstrucción de contenidos multimedia.

Al igual que sucede en la mayoría de arquitecturas distribuidas, la posibilidad de replicación de equipos permiten garantizar un servicio continuo, así como de mejora en los tiempos de latencia. Estas características también se encuentran en los sistemas distribuidos de realidad aumentada, haciéndolas mucho más fiables que los sistemas autónomos, susceptibles a fallos internos que pueden llegar a dejar inoperativo el servicio que deben ofrecer.

Por último, los sistemas distribuidos de realidad aumentada suelen ser susceptibles al estado de la red. Aquellos sistemas que requieran de funcionalidad externa y cuyo medio de intercambio de información sea la red se ven expuestos a varias circunstancias negativas que pueden llegar a paralizar el servicio. Algunas de estas eventualidades son:

- Ataques externos. Entidades externas al sistema pueden realizar ataques contra el mismo, ocasionando pérdidas de rendimiento e incluso detenciones del servicio.
- Retrasos en la red. Dependiendo de la intensidad de las señales, ciertas funcionalidades pueden verse solapadas con otras si no se adoptan las convenientes medidas de sincronismo. Además, en sistemas de tiempo real, el servicio puede verse ralentizado, haciendo incluso que no se pueda utilizar el mismo.
- Susceptibilidad a conexiones de red. Si los sistemas dependen de agentes externos con los que se comunican por red, ante la imposibilidad de conexión el servicio podría quedar inutilizado. Es conveniente poder garantizar una serie de funcionalidades mínimos ante esta eventualidad.

4.3.3 Conclusiones

En esta sección se han descrito las diferencias entre implantar un sistema autónomo de realidad aumentada o uno distribuido. También se han visto las ventajas e inconvenientes de cada una de las dos opciones.

La elección de una de las dos configuraciones no es una tarea sencilla, ni tampoco tiene por qué ser una decisión inflexible. En nuestro proyecto sin ir más lejos se decidió realizar una sistema autónomo en las primeras fases, mientras que en última instancia se decidió implantar un sistema distribuido debido a las necesidades de éste, ya descritas al principio del capítulo.

En base a nuestra experiencia y al conocimiento extraído del análisis de diferentes proyectos, concluimos que la elección de una u otra configuración depende en gran medida de las prestaciones de la aplicación o sistema y a las posibilidades económicas del proyecto. Implantar un sistema con un clúster de servidores elevado puede suponer una inversión muy alta que puede no estar a la altura de los presupuestos.

Por lo general, implantar un sistema distribuido puede ser la medida más eficaz si se quiere realizar tareas complejas en cuanto a capacidad de cálculo u ofrecer una experiencia de uso espectacular que requiera de equipos especializados.

Por último, la implantación de un sistema autónomo puede ser la solución perfecta si lo que se desea es dar un servicio de realidad aumentada en entornos pequeños y bien conocidos, como podría ser el caso de un pequeño museo.

4.4 Configuración por parte del proveedor y del cliente en los sistemas desarrollados en este trabajo

En este apartado se describen los elementos empleados en la implantación del sistema desarrollado junto a Telefónica I+D.

El presente proyecto hace referencia a un sistema de realidad aumentada que permite la visualización de contenidos digitales mediante el uso de un teléfono móvil con sistema operativo Android sobre una serie de muestras relativas al museo de informática García Santesmases (MIGS) que se encuentra en la facultad de informática de la universidad complutense de Madrid [G SANT 10]. De igual manera, se ha desarrollado un sistema de control de elementos domóticos a los que se accede mediante realidad aumentada, mostrando información relevante sobre ellos.

Para el reconocimiento de los objetos se ha utilizado una técnica propia de reconocimiento de marcadores en blanco y negro que será expuesta en un apartado posterior de esta memoria. El usuario, una vez distinguida la escena visualizada, ve en la pantalla de su teléfono una serie de datos informativos y educativos relativos a la muestra del museo señalada.

Desde el punto de vista arquitectónico del sistema, éste está compuesto por tres entidades funcionales:

- La aplicación instalada en el teléfono móvil. Este agente se encarga del reconocimiento de la escena, así como de la superposición de contenidos.
- Un repositorio externo en el que se encuentra la información de aumento de cada muestra. Este repositorio consiste en un servidor que aloja dicha información y que establece una comunicación con el dispositivo móvil.
- Un sistema de reconocimiento de voz para las funciones de accesibilidad a los contenidos. Se utiliza para esta misión el servicio de Google dedicado [AND TTS 10]. Este servicio permite transformar una fuente de audio procedente de la voz en un texto. En nuestro sistema el texto será tratado de forma local en el móvil para procesar órdenes y peticiones del usuario.

La figura 4.5 muestra la relación entre los agentes involucrados en nuestro sistema.

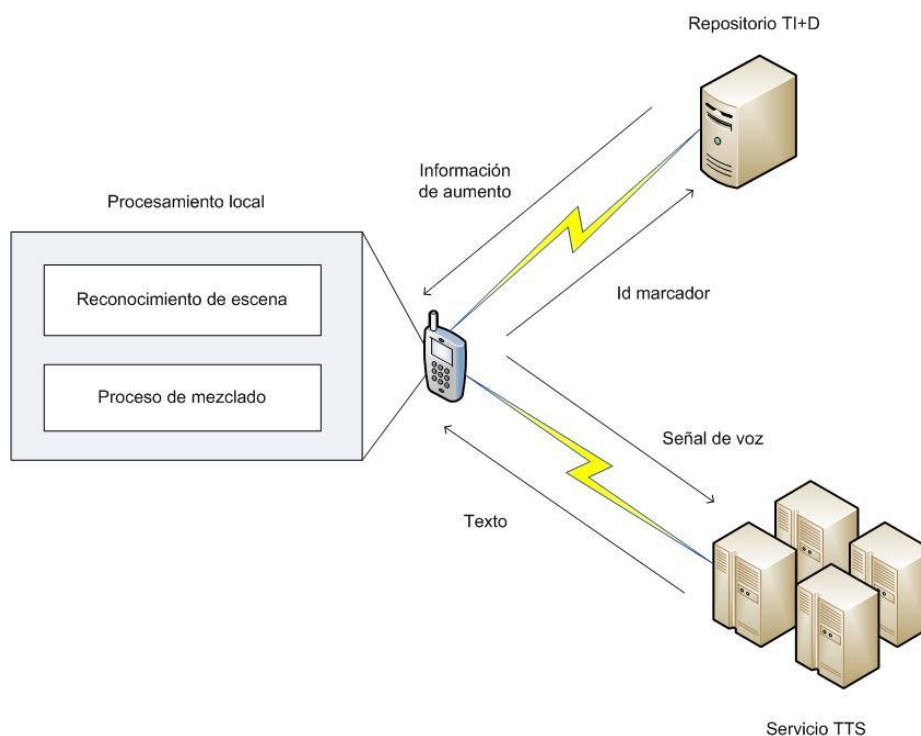


Figura 4.5. Relación entre agentes funcionales en un sistema distribuido.

A continuación se especificarán los requisitos que deben cumplir tanto el proveedor, Telefónica I+D, como el cliente del servicio, tanto para el sistema del MIGS como para el sistema de control domótico.

4.4.1 Requisitos del proveedor

Para poder explotar al 100% el rendimiento y las características de nuestro sistema es necesario disponer de conexión a Internet. Antes de continuar con la explicación de los requisitos por parte del proveedor, cabe destacar que el sistema puede funcionar como aplicación local, dando un servicio mínimo de realidad aumentada, de modo que si el teléfono no dispone de conexión a Internet no sea una aplicación inerte. Este servicio mínimo garantiza una pequeña información de muestra sobre los elementos del museo.

Como se comentaba anteriormente, el sistema necesita una velocidad mínima de conexión a Internet para poder funcionar al 100% de sus características. Es decisión del cliente la elección del proveedor de acceso a Internet. No obstante, la facultad de informática de la UCM ofrece un servicio gratuito de acceso a Internet mediante WIFI que se puede utilizar en cualquier lugar de dicha facultad y Movistar ofrece una amplia variedad de servicios de conexión.

Siguiendo en la misma línea, gracias a la conexión a Internet, el sistema puede mostrar una cantidad de información mayor. El servidor que actúa como repositorio es un equipo perteneciente a Telefónica I+D. El proceso de selección de contenidos es el siguiente. En primer lugar, la aplicación localiza el marcador y selecciona el patrón que mejor encaja. Una vez distinguido qué marcador es el visualizado, se establece una comunicación con el servidor pidiéndole que envíe la información pertinente a la muestra seleccionada. El servidor, por su parte, procesa la petición, busca qué información ha sido solicitada y la envía por la red hasta el teléfono móvil. Éste, al recibir la respuesta, realiza la mezcla entre realidad y aumento y se la muestra al usuario. Las figuras 4.6 y 4.7 muestran el proceso en local y en el servidor.

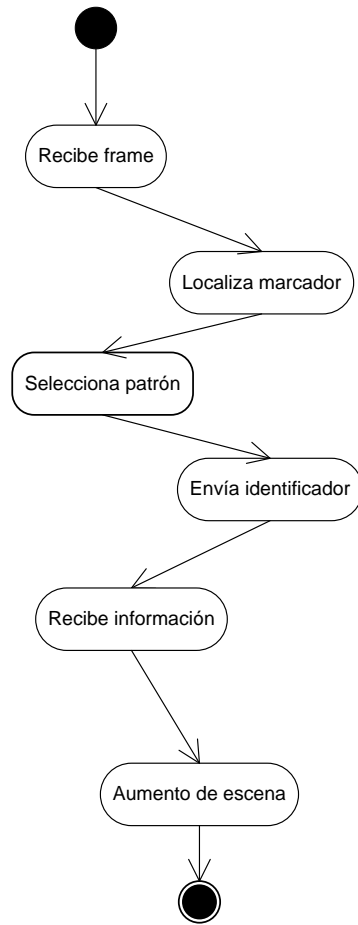


Figura 4.6. Diagrama de actividad del proceso local.

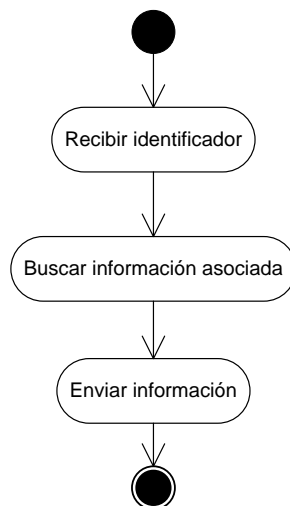


Figura 4.7. Diagrama de actividad del servidor.

Una vez descrito el proceso de aumento de información, se hacen evidentes dos factores críticos de éxito del sistema en este apartado. Por un lado es necesario que el proveedor, en este caso Telefónica I+D, garantice un servicio continuo, o que al menos sea acorde al horario

de visita del museo, para ofrecer los contenidos de aumento y que el sistema tenga una buena acogida entre los usuarios, permitiendo que éstos puedan disfrutar al máximo de las características del sistema. Por otro lado, el ancho de banda de las redes móviles es un elemento clave en el uso del sistema. En una red lenta, los contenidos podrían llegar con retraso, haciendo que la experiencia del usuario fuera bastante incómoda. Aunque este problema se ha tenido en cuenta, usando contadores de tiempo para establecer el modo de servicio mínimo ante grandes retrasos de la red, hoy en día las redes de comunicaciones dan un soporte bastante eficiente y rápido, permitiendo tiempos de respuesta lo suficientemente bajos, de forma que el servicio sea cómodo para el usuario. Además, es necesario destacar que las futuras mejoras en las redes móviles permitirán unos tiempos de respuesta mucho más óptimos, pudiendo ofrecer una cantidad mayor de contenidos, enriqueciendo aún más la escena.

El otro elemento dependiente del acceso a Internet es, como se ha comentado anteriormente, la función de traducción de órdenes de voz. Este servicio lo ofrece Google y depende íntegramente del estado de sus servidores.

Por su parte, puesto que, por regla general, los sistemas de realidad aumentada basados en reconocimiento visual son bastante sensibles a la iluminación, el museo debe asegurar una iluminación estable a cualquier hora del día. De lo contrario, el sistema de reconocimiento de marcadores no puede asegurar su disponibilidad en unos márgenes altos.

Por tanto, a modo de resumen, los elementos que debe aportar el proveedor son:

- Servidor repositorio de información disponible durante apertura de museo.
- Acceso a Internet.
- Disponibilidad del servicio Text-to-speech de Google.

Las siguientes tablas muestran los componentes hardware y software que habitualmente deben tener los proveedores.

Tabla 4.2. Resumen de los componentes hardware desde el punto de vista del proveedor.

Técnica	Componente hardware
Almacenamiento	Servidor de base de datos
Comunicaciones locales	Tarjeta de red, conectores RJ45, antena wireless, punto de acceso
Comunicaciones móviles	Servidor de peticiones, infraestructura de telefonía móvil

Tabla 4.2. Resumen de los requisitos software desde el punto de vista del proveedor.

Técnica	Componente software
Almacenamiento	Base de datos
Comunicaciones locales	Controlador de red
Comunicaciones móviles	Controlador GSM, software de servidor

4.4.2 Requisitos del cliente

Por su parte, el cliente necesita cumplir una serie de requisitos mínimos para poder utilizar los servicios del sistema de realidad aumentada. Estos requisitos están centrados principalmente en la funcionalidad del teléfono móvil que utilice. Aunque el procesamiento es

bastante rápido, se recomienda disponer de un teléfono con una velocidad superior a los 200 Mhz y una memoria de 128 Mb de RAM.

Por otro lado, como ya se comentó anteriormente, es requisito indispensable que el teléfono funcione bajo el sistema operativo Android. Para futuras versiones se tiene en mente ofrecer el mismo servicio para otras plataformas, como iPhone.

Otra de las características que debe cumplir el teléfono es, evidentemente, que disponga de cámara de vídeo. Puesto que el sistema realiza el reconocimiento de forma visual y no mediante técnicas de geo-posicionamiento, este elemento es fundamental para que el sistema pueda funcionar.

Dado que gran parte de las mejoras del servicio requieren de conexión a Internet, el cliente debería tener contratado un servicio de acceso a Internet. No obstante, como se comentó con anterioridad, la facultad de informática ofrece acceso gratuito a sus estudiantes mediante WIFI. A pesar de ello, se recomienda la contratación de un servicio, puesto que las transmisiones WIFI no son todo lo rápidas que se desearía, pudiéndose notar un leve decremento en la velocidad a la que se muestran los contenidos.

Las dos siguientes tablas muestran los requisitos hardware y software que deben cumplir los clientes de sistemas de realidad aumentada.

Tabla 4.3. Resumen de los componentes hardware desde el punto de vista del cliente.

Técnica	Componente hardware
Captura <i>video-through</i>	Todo tipo de cámaras de vídeo
Captura <i>see-through</i>	Cámaras integradas de vídeo, HMD (<i>Head Mounted Display</i>)
Reconocimiento geo-posicionamiento	Antena GPS, sensores de movimiento
Reconocimiento híbrido	Antena GPS, sensores de movimiento, dispositivos de captura de imágenes
Tratamiento imágenes	Procesadores con velocidad de procesamiento de, al menos, 2 GHz
Comunicaciones móviles	Equipamiento GSM
Visualización <i>video-through</i>	Pantallas de vídeo, monitores, proyectores
Visualización <i>see-through</i>	HMD, teléfonos móviles

Tabla 4.4. Resumen de los requisitos software desde el punto de vista del cliente.

Técnica	Componente software
Captura escena	Controladores de cámara de vídeo
Reconocimiento visual	Librerías de reconocimiento de imágenes
Reconocimiento geo-posicionamiento	Brújula digital, software GPS
Reconocimiento híbrido	Brújula digital, librerías de reconocimiento de imágenes, software GPS
Tratamiento imágenes	Librerías de tratamiento de imágenes
Comunicaciones locales	Controlador de red
Comunicaciones móviles	Controlador GSM
Visualización contenidos	Software de reproducción de contenidos multimedia, librerías de tratamiento de imágenes

5. APLICACIONES AL E-LEARNING Y A LA DOMÓTICA

En este capítulo se describen dos aplicaciones de la realidad aumentada desarrolladas en el contexto de este trabajo. Una se aplica al Museo de Informática García Santesmases, y la otra a un proyecto de domótica desarrollado en la empresa Telefónica I+D. La arquitectura sigue la propuesta definida en el capítulo anterior.

Así mismo, en este capítulo se explican los diferentes pasos seguidos para realizar el proyecto que esta memoria aborda. Para ello se expondrán las diferentes etapas de su construcción, explicando cuáles fueron los resultados y motivos de aquellas decisiones incorrectas y detallando los contenidos de las medidas adoptadas.

La estructura del capítulo es la siguiente, el primer punto es una introducción al por qué ambas aplicaciones. En la segunda sección se explican las técnicas de reconocimiento de escenas elegidas. Siguiendo la línea temporal de los acontecimientos, se expondrán primero las alternativas desechadas para concluir en la descripción de la técnica finalmente empleada. La siguiente sección expondrá, de igual manera, la forma en que se aumenta la escena física real. El cuarto punto explica la arquitectura global del sistema construido. El objetivo es poner de manifiesto las ventajas e inconvenientes de la aproximación seguida para que puedan ser tenidas en cuenta por aquellas personas interesadas en implantar una aplicación de realidad aumentada. Las dos últimas secciones muestran el funcionamiento de ambos sistemas, en primer lugar se expone el sistema para el Museo García Santesmases y finalmente se expone el sistema de domótica realizado para Telefónica I+D.

Por último, cabe destacar que como ambas aplicaciones comparten un diseño común, se hará una descripción común para ambas en cuanto a la técnica empleada.

5.1 Motivación de las aplicaciones

En esta sección se expone la funcionalidad de los dos sistemas utilizando lenguaje natural.

El objetivo de ambos sistemas es que al enfocar con la cámara del teléfono móvil sobre cualquiera de los marcadores conocidos, éstos muestren información relacionada sobre los elementos que representan los marcadores. La figura 5.1 muestra dos ejemplos de funcionamiento de aumento sobre los marcadores.

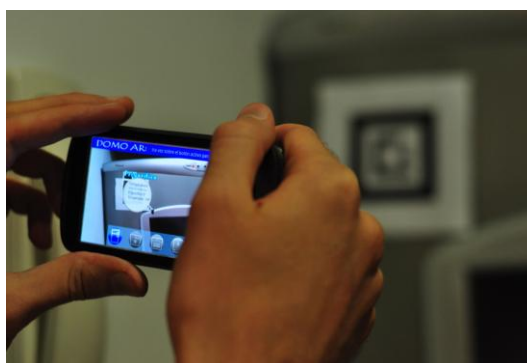


Figura 5.1. Ejemplo de aumento básico de ambos sistemas.

De la misma forma, se ofrece la opción de mostrar contenidos más amplios con un interfaz simple y de forma muy intuitiva. En la figura 5.2 se puede observar una exposición de contenidos más amplia ofrecida por ambos sistemas.

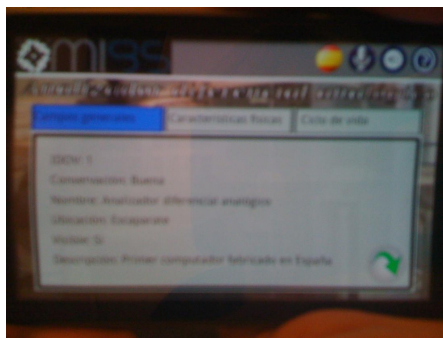


Figura 5.2. Ejemplo de aumento ampliado de ambos sistemas.

Aparte de los contenidos de aumento, ambos sistemas ofrecen la posibilidad de cambiar entre los idiomas español e inglés, así como de la interacción con el sistema mediante comandos vocales.

En las tres próximas secciones se describen las técnicas empleadas para la construcción de ambos sistemas.

5.2 Identificación de escenarios

El primer paso en la construcción del sistema de realidad aumentada fue la elección de la técnica de reconocimiento de escenario que se iba a emplear.

En este punto la primera duda que surgió fue si utilizar técnicas de identificación visuales, de geo-posicionamiento o híbridas. En el caso del museo de informática, debido a su localización en el interior de la facultad de informática de la UCM y dado que las técnicas GPS no son efectivas en interiores, se desearon las opciones de reconocimiento mediante geo-posicionamiento e híbridas. Por tanto, se decidió utilizar técnicas de reconocimiento visual.

En el caso de la aplicación de domótica para Telefónica I+D, el problema fue similar. Puesto que la aplicación estaba diseñada para ofrecer un servicio en hogares, y debido a que en el interior de éstos no suele ser habitual disponer de suficiente precisión en técnicas de geo-posicionamiento, se decidió que el reconocimiento de escenarios debía hacerse mediante técnicas de reconocimiento visual.

Debido a la similitud entre ambas aplicaciones, se decidió utilizar la misma técnica para la identificación de escenarios mediante reconocimiento visual para las dos aplicaciones.

En esta sección se describe siguiendo la línea temporal de los acontecimientos las diferentes técnicas empleadas en las distintas fases del proyecto, explicando cuáles fueron los problemas por los que se desearon.

En primer lugar se describen las técnicas de visión artificial clásica. En la segunda subsección se describe el resultado de emplear redes neuronales en el reconocimiento de escenarios. El resto de apartados describen los diferentes algoritmos diseñados para el reconocimiento de marcadores.

5.2.1 Técnicas clásicas de visión artificial

Las primeras pruebas que se realizaron fueron utilizando las técnicas clásicas de visión artificial expuestas en el capítulo 2 de esta memoria.

Este estudio se realizó utilizando el flujo de vídeo de una webcam y empleando para el procesamiento un ordenador con 2Gb de RAM y un procesador Pentium IV a 3 GHz. Todos los programas fueron diseñados en Java y ejecutados bajo la JVM.

En primer lugar se probaron las siguientes técnicas de filtrado:

- Filtro de la media.
- Filtro de la mediana.
- Filtro gaussiano.

Este tipo de pre-procesado no aportó grandes mejoras en el formato de la imagen que debía ser procesada, proporcionando únicamente mayores tiempos de cálculos. Debido a que la naturaleza en tiempo real del sistema, el reconocimiento debía ser lo más rápido posible, por lo que estas técnicas fueron desechadas en el primer instante.

La figura 5.3 muestra los resultados de aplicar los diferentes filtros.



Figura 5.3. Efecto de aplicar diferentes filtros clásicos. En la parte superior, la imagen original. De izquierda a derecha, filtro de media 5x5, filtro de mediana 5x5 y filtro gaussiano 5x5.

El siguiente paso fue probar las técnicas clásicas de detección de bordes. Las técnicas puestas a prueba fueron:

- Detección horizontal.
- Detección vertical.
- Sobel.

Estas técnicas demostraron ser un arma eficaz para obtener información de la escena, pero por el contrario resultaron ser demasiado costosas en cuanto a tiempo de cálculos, alcanzando, en los peores casos, tiempos superiores al minuto. Los elevados tiempos de procesamiento requeridos fueron la razón por la que se descartó utilizar técnicas de visión

artificial clásica en el reconocimiento de escenarios. La figura 5.4 muestra los resultados de aplicar alguna de estas técnicas de detección de bordes.

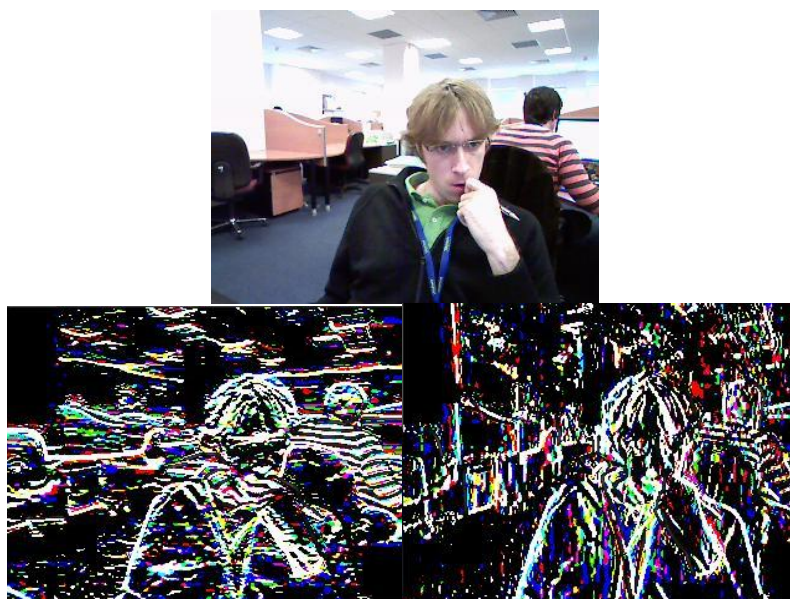


Figura 5.4. Efecto de aplicar detección de bordes horizontal (imagen inferior izquierda) y vertical (imagen inferior derecha) sobre la imagen original.

Finalmente, se probó a realizar un último intento utilizando técnicas de segmentación. Si bien los resultados fueron bastante satisfactorios y el tiempo de procesamiento, aunque alto, fue más bajo que con la detección de bordes, surgió un nuevo problema: el aumento de memoria. A medida que la segmentación va construyendo el árbol de la imagen, la pila de memoria crece con él. Si bien en el ordenador en el que se hicieron las pruebas no hubo ningún problema, puesto que las técnicas de identificación deberían llevarse a cabo más adelante en terminales móviles con una memoria mucho más limitada, esta opción fue descartada.

5.2.2 Uso de redes neuronales

La siguiente alternativa llevada a cabo fue la del uso de redes neuronales para la clasificación de muestras.

Históricamente, las redes neuronales han resultado una herramienta eficaz para la optimización de funciones y la clasificación de patrones. Gracias a la segunda capacidad citada de esta familia de algoritmos, se intentó utilizarlas para el reconocimiento de escenarios.

La principal razón de escoger utilizar una red neuronal para la clasificación de imágenes estuvo basada en la idea de que ante un número lo suficiente elevado de muestras que abarcasen cualquier escena captada por el usuario, ante una imagen de entrada ya conocida, la red sería capaz de clasificarla adecuadamente. Si bien es cierto que este proceso podía haberse hecho mediante comparación de imágenes en bruto, ante un número tan alto, y puesto que la capacidad de cálculo de un móvil es bastante limitada, el tiempo de respuesta hubiese sido demasiado alto. Además, las redes neuronales, sobre todo las de aprendizaje supervisado delegan los tiempos de carga elevados en la fase de aprendizaje, siendo el tiempo de respuesta de clasificación bastante bajo.

Características de la red

El tipo de red escogida fue *una back-propagation* [HIL 95, PAT 96, HAY 94], ya que este tipo de redes se ajustan de forma muy cómoda tratando de encontrar la función de ajuste más adecuada en cada momento.

El algoritmo de aprendizaje utilizó ajuste por mínimos cuadrados, utilizando el algoritmo de Levenberg Marquadt [MAT 10, LEV 44], que se basa en derivadas posteriores para la predicción de los pesos de la red.

La red recibiría 48 entradas relativas al contenido de la imagen. Por tanto, la red debía ajustar una función de 48 dimensiones.

Las 48 entradas recientemente citadas se extraen de la descomposición en rejilla de la imagen de entrada. Para ello, la imagen se descompone en 48 cuadrantes, de los cuales se extrae de cada uno de ellos la media y se normaliza para que sus valores sean continuos en el rango [0,1]. Cada uno de estos 48 valores sería, más adelante, cada una de las entradas a la red. En la figura 5.5 se puede ver la descomposición de una imagen en 48 regiones.



Figura 5.5. Separación en 48 regiones de una imagen de entrada.

La razón por la que se descompuso la imagen en 48 dimensiones fue la de descomponer la escena en regiones lo suficientemente características, sin un alto ruido contextual. Cabe destacar que en un primer intento se probó a realizar los entrenamientos de la red introduciendo 80 entradas (8x10 regiones). Esta vía fue desechada debido a que las prestaciones de los equipos de entrenamiento eran limitadas, sucediéndose los problemas de alojamiento de memoria.

La siguiente tarea fue escoger el número de salidas de la red. Para las pruebas se optó por una configuración que devolviese 3 salidas binarias, dando lugar a 8 posibles patrones a clasificar.

Por otro lado, para que la red ajustase de la mejor forma posible, se tomaron alrededor de 20.000 muestras, de las cuales un 80% se destinaron a las tareas de entrenamiento y validación, y el restante 20% a tareas de validación. Cabe destacar que las muestras fueron tomadas a distintas horas del día. La razón de esta decisión fue la de abarcar los escenarios en cualquier momento del día, con el fin de fortalecer el posterior algoritmo de reconocimiento.

En última instancia, resaltar que el trabajo de extracción de muestras fue una tarea ardua y pesada, ya que, en un primer momento, contábamos con alrededor de 30.000 muestras que fueron filtradas de forma completamente supervisada, eliminando aquellas con ruido por movimiento u otras eventualidades. Cabe destacar que la tarea de obtención de muestras fue lo más amplia posible, con el fin de poder contar con todas las panorámicas posibles de cada muestra. Este proceso, por sí solo, requirió de, aproximadamente, 25 horas.

Prueba 1

Para el primer intento se diseñó una red de una sola capa utilizando como función de transferencia una función logaritmo sigmoide.

En cuanto a la arquitectura de la red, se probaron diferentes configuraciones, desde una red con 4 neuronas en capa oculta hasta una de 30, siendo en todas ellas el error de clasificación demasiado elevado, siendo, en el mejor de los escenarios, el error en torno al 80%. No se siguió probando más arquitecturas, ya que a partir de la arquitectura con 25 neuronas el error dejó de disminuir o, mejor dicho, la relación tiempo-disminución de error dejó de ser eficiente.

Prueba 2

Una vez determinado que una arquitectura de una sola capa no era una solución válida se optó por una configuración de dos capas ocultas. En el primer intento se mantuvo como función de transferencia la función logaritmo sigmoide. Tras diversas pruebas, la mejor arquitectura encontrada fue una con 12 neuronas en la primera capa oculta y 6 en la segunda. La figura 5.6 muestra la arquitectura de la red.

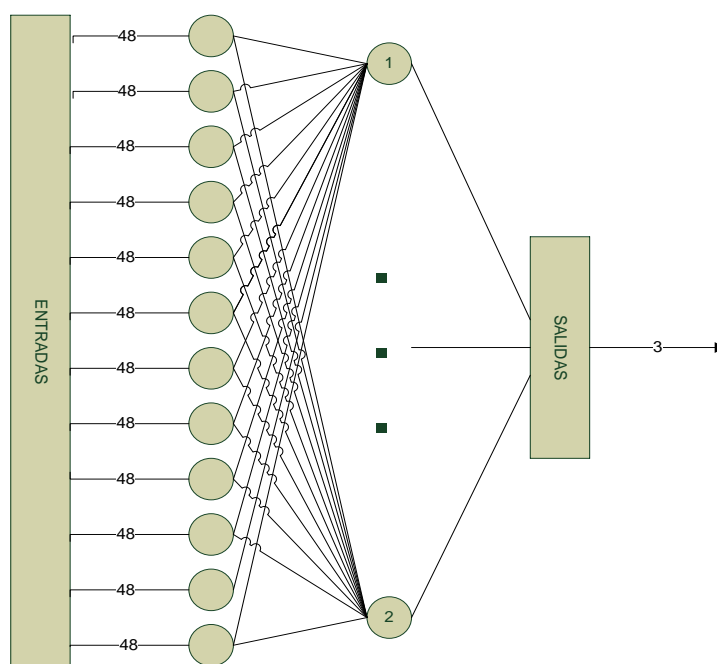


Figura 5.6. Arquitectura de la red neuronal utilizada.

Los resultados de esta arquitectura fueron bastante más prometedores, alcanzando solamente un 30% de error.

Prueba 3

A pesar de que en la recientemente expuesta arquitectura se obtuvieron unos buenos resultados, se detectó que la red podría adaptarse mucho mejor de lo que lo estaba haciendo.

La razón por la que no estaba funcionando de la mejor forma posible era que la función de transferencia, la función logaritmo sigmoide, tiene una mayor pendiente y que los valores de la función oscilan entre 0 y 1, dejando poco lugar para la clasificación de 20.000 muestras.

Para solucionar el problema se introdujo la función tangente hiperbólica como función de transferencia, que tiene una pendiente más suave y un mayor rango de clasificación. La figura 5.7 muestra las diferencias entre las funciones tangente hiperbólica y logaritmo sigmoide.

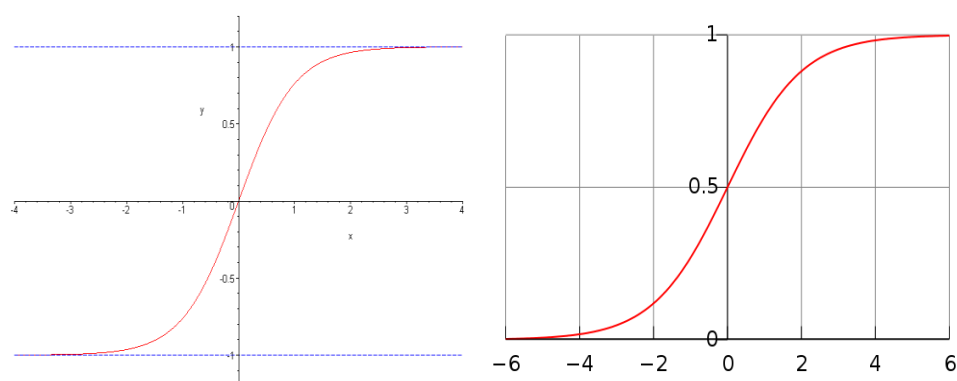


Figura 5.7. Diferencias entre la tangente hiperbólica y el logaritmo sigmoide.

Como arquitectura se utilizó la misma que ya ha sido expuesta en la anterior prueba, 12 neuronas en la primera capa oculta y 6 en la segunda. Los resultados de esta configuración fueron sorprendentes, ya que estuvieron en torno al 5% al contrastar la red con los datos de validación.

Pruebas en entorno real

En vista de los últimos resultados expuestos se hicieron las pruebas de reconocimiento en el entorno real. Para ello se instaló en el dispositivo móvil un programa con la red neuronal que recibía en tiempo real el flujo de vídeo de la cámara.

El rendimiento del reconocimiento fue bastante bueno, cada procesamiento de una entrada de vídeo por parte de la red tuvo un tiempo aproximado de 200 ms, en contra de los, aproximadamente, 100 ms de período que tiene la cámara en lanzar un nuevo *frame* a la aplicación, permitiendo un reconocimiento cada dos *frames*, una velocidad muy buena.

No obstante, en contraposición al rendimiento de la red, su eficacia no fue tan positiva. Los resultados obtenidos oscilaban en un 50% de error en el reconocimiento. Posteriormente se analizaron los datos obtenidos y se observó que el problema estaba en que con un número tan elevado de muestras la red había sufrido sobre aprendizaje, siendo incapaz de adaptarse ante nuevos estímulos y, que ante muestras no recogidas previamente, de clasificar correctamente estos estímulos. La única solución posible para seguir con esta vía de clasificación era tomar un

número mayor de muestras, de forma que abarcase todo el espectro posible de entradas a la red. Esta solución resultó ser inviable, no sólo por el alto esfuerzo de recoger un número tan alto de muestras, sino porque, además, seguiría siendo vulnerable a cualquier tipo de ruido imprevisto, tales como reflejos, sombras de personas, averías eléctricas en los equipos de iluminación, etc.

5.2.3 Reconocimiento inicial de marcadores

Tras probar a reconocer la escena mediante técnicas de reconocimiento visual sin uso de marcadores, se llegó a la conclusión de que, si bien era posible acometerse, o bien suponía un alto coste computacional reflejado en la usabilidad del producto, o bien requería de un esfuerzo elevado para la preparación del reconocedor. Puesto que no se disponía de ninguna de las dos características, se planteó la posibilidad de utilizar marcadores en la escena para reconocer la misma.

Como ya se ha visto en anteriores secciones de esta memoria, un marcador no es sino un objeto en la escena conocido por el sistema. Un marcador puede tener muchas formas y colores, aunque lo más habitual es que sean blancos y negros y de formas cuadradas. En la figura 5.8 se pueden ver diferentes tipos de marcadores.

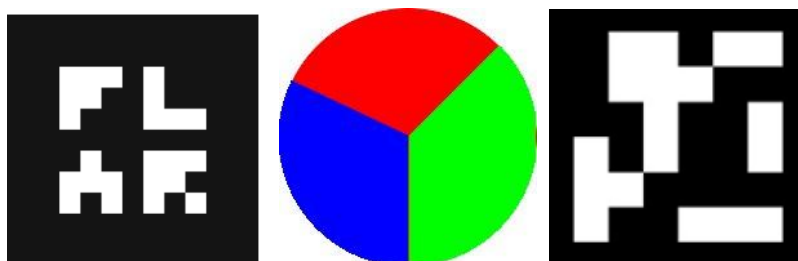


Figura 5.8. Distintos tipos de marcadores.

Desde el primer momento se escogió utilizar marcadores negros cuadrados con figuras geométricas blancas en su interior, ya que sería más sencillo realizar las búsquedas transformando la imagen de entrada a una escala de grises.

Búsqueda mediante cuadrados

En la primera prueba se diseñó un algoritmo que reconocía aquellos objetos en la escena que pudieran ser candidatos a marcador. Para ello era suficiente poder encontrar un cuadrado con los bordes negros para que fuese candidato. El algoritmo de búsqueda era el siguiente:

0. $X=0, Y=0$.
1. Explorar píxel $[X, Y]$. Si es negro $X=X+1$ y repetir paso 1. Si es blanco ir a paso 2.
2. Volver a X original y repetir paso 1 con incremento en Y . Si se vuelve a 2 y la distancia horizontal es igual a la distancia vertical recorrida ir a paso 3. Si no ir a siguiente píxel en paso 1.
3. Repetir búsqueda para el lado horizontal inferior y para el lado vertical derecho. Si todos los lados son iguales es marcador.

Este algoritmo no obtuvo los resultados deseados, aunque en entornos con una iluminación y contexto muy controlado dio buenos resultados, tal y como se muestra en la figura 5.9.



Figura 5.9. Resultado de búsqueda del primer algoritmo.

Otro de los inconvenientes del uso de este algoritmo fue el tiempo de respuesta de la aplicación. Al tener que recorrer la imagen de forma matricial en su totalidad y realizar una serie de análisis, el número de operaciones fue bastante elevado, viéndose incrementado considerablemente el tiempo de análisis, oscilando éste en torno a los 5 segundos en cada reconocimiento en el ordenador de pruebas anteriormente descrito.

Encaje de patrones matriciales

Puesto que este primer algoritmo no dio los resultados esperados se puso en prueba una segunda estrategia. Esta estrategia estaba ideada en reducir el número de cálculos que el equipo tuviese que realizar y disminuir la cantidad de memoria necesaria para llevarla a cabo. Para ello, se descubrió mediante el análisis de decenas de imágenes convertidas a blanco y negro que, en imágenes relativamente grandes superiores a los 200x300 píxeles, este tipo de marcadores eran detectables si se analizaba uno de cada 25 píxeles, representando éste a todos sus píxeles colindantes. Esta estrategia disminuye, por tanto, en 25 el número de operaciones y de memoria necesaria. La figura 5.10 muestra el efecto de aplicar esta estrategia.

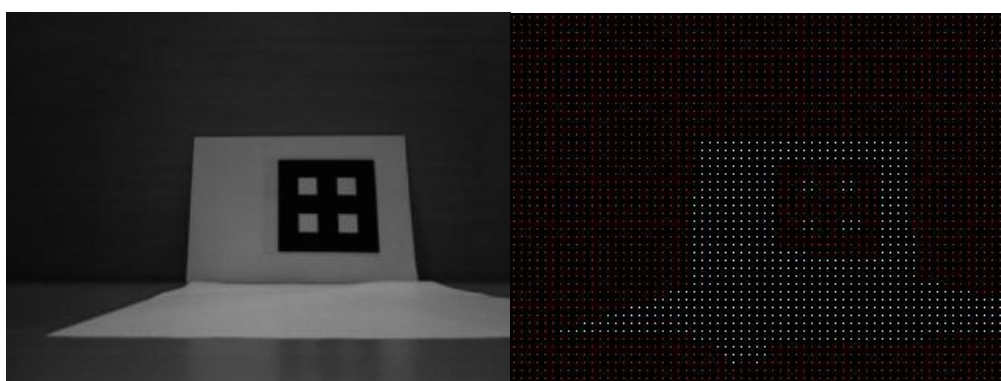


Figura 5.10. Reducción de una imagen por 25.

Puesto que la estrategia recientemente descrita reducía considerablemente el espacio matricial sobre el que realizar la búsqueda y, gracias a su aspecto visual intuitivo, se ideó que se podía intentar encajar la entrada reducida con las matrices relativas a los patrones de los marcadores.

El proceso de búsqueda fue el siguiente: la matriz se recorría de forma normal, explorando cada píxel. Si el píxel explorado encajaba con alguna de las esquinas de las matrices de los patrones, entonces se procedía a la comparación de la sub-matriz de la imagen con origen en el píxel explorado con la matriz del patrón almacenado. Si la comparación superaba un cierto umbral entonces el marcador había sido localizado. Si ninguno de los anteriores requisitos se cumplía se continuaba la búsqueda iterativamente por toda la matriz hasta, o bien alcanzar el final o hasta encontrar una sub-matriz que encajase con el patrón. En la figura 5.11 se puede ver el resultado de aplicar el algoritmo sobre las imágenes de la figura 5.10.



Figura 5.11. Resultado de la búsqueda por encaje de patrones matricial.

Por su parte, las matrices de patrones tenían el siguiente formato, eran matrices binarias en las que un 1 representaba el valor negro y un 0 el valor blanco. La matriz que encajaba con el marcador de la figura 5.10 tenía la siguiente forma:

```
1 1 1 1 1
1 0 1 0 1
1 1 1 1 1
1 0 1 0 1
1 1 1 1 1
```

Este algoritmo demostró ser eficaz con aquellos marcadores cuyo lado superior horizontal estuviese en paralelo al eje X de la imagen y cuyos lados verticales se encontraran a 90º de dicho eje. La razón de estos requisitos era que, sobre todo ante inclinaciones, para que ajustase a un patrón, cualquier eventualidad en forma de giro o inclinación debía estar almacenada de forma matricial en la aplicación.

Otro de los problemas que tenía era la dependencia a la escala del marcador. Viendo la figura 5.10 se puede apreciar de forma intuitiva que, cuanto más cerca estuviese el marcador

de la cámara, mayor serán el número de 1s y 0s que tiene que tener la matriz y, por tanto, un número de patrones mayores a comparar y almacenar.

Por último, otro de los grandes problemas que presentaba este algoritmo era el ruido lumínico. Para realizar la conversión de una imagen a color a su correspondiente equivalencia en blanco y negro era necesario determinar un umbral sobre el que los valores mayores correspondiesen al blanco y los menores al negro. Esta umbralización es muy dependiente de la iluminación, pudiendo la más leve sombra o reflejo modificar los valores de blanco o negro respectivamente. Para poder controlar esta problemática era necesario incluir cada una de estas eventualidades de forma matricial como un patrón más de la aplicación.

En definitiva, para que la aplicación pudiese reconocer un solo marcador de forma aceptable era necesario contar con patrones en diferentes inclinaciones, giros, escalas, ruidos lumínicos y las posibles combinaciones entre ellos, siendo el abanico de posibilidades muy elevado y, por tanto, haciendo inviable adoptar esta estrategia.

Búsqueda de esquinas interiores

La última estrategia fallida llevada a cabo fue la de buscar cuatro esquinas interiores que se ajustasen al patrón:

Esquina superior izquierda

$$\begin{aligned}[x, y] &= 1, \\ [x+1, y] &= 1, \\ [x, y+1] &= 1, \\ [x+1, y+1] &= 0.\end{aligned}$$

Esquina superior derecha

$$\begin{aligned}[x, y] &= 1, \\ [x+1, y] &= 1, \\ [x, y+1] &= 0, \\ [x+1, y+1] &= 1.\end{aligned}$$

Esquina inferior izquierda

$$\begin{aligned}[x, y] &= 1, \\ [x+1, y] &= 0, \\ [x, y+1] &= 1, \\ [x+1, y+1] &= 1.\end{aligned}$$

Esquina inferior derecha

$$\begin{aligned}[x, y] &= 0, \\ [x+1, y] &= 1, \\ [x, y+1] &= 1, \\ [x+1, y+1] &= 1.\end{aligned}$$

Como se puede ver, lo que se trataba de encontrar era cuatro cuadrados blancos interiores que actuaran como esquinas para el reconocimiento.

Este algoritmo funcionaba de la siguiente forma: se recorría la matriz reducida buscando cuatro puntos que cumpliesen los requisitos de la esquina superior izquierda. Si no los cumplían se seguía buscando hasta encontrar alguno que si cumpliese o bien hasta alcanzar el final de la matriz. Si por el contrario se encontraban cuatro puntos que ajustasen se procedía a

la búsqueda de los otros tres patrones, dando por localizado el marcador si el resultado de la búsqueda era positivo y continuando el proceso si sucedía lo contrario.

Este algoritmo fue bastante eficaz por dos razones:

- El tiempo de búsqueda se disminuyó considerablemente al tener que encajar menos puntos (sólo 4 por cada patrón) que en la búsqueda por encaje matricial.
- Discriminaba escalas e inclinaciones, ya que la búsqueda la hacía hasta un límite fijado, pudiendo devolver lados del cuadrado mayores que otros o figuras geométricas más complejas.

No obstante, esta estrategia tenía un punto débil. Los marcadores que se utilizasen debían contener 4 cuadrados blancos que actuaran como esquinas interiores, limitando considerablemente el número de marcadores que el sistema pudiese reconocer, ya que los marcadores debían ser un objeto no demasiado intrusivo en la escena física real. En la figura 5.12 se puede ver un ejemplo de este tipo de marcadores.



Figura 5.12. Marcadores con cuadrados blancos actuando de esquinas interiores.

5.2.4 Búsqueda laberíntica

En esta sección se describe el siguiente algoritmo de búsqueda desarrollado para el sistema de reconocimiento de marcadores.

A pesar de que las estrategias descritas hasta el momento no produjeron resultados satisfactorios, de ellas se pudieron extraer una serie de conclusiones beneficiosas para diseñar un algoritmo mucho más potente y eficaz de reconocimiento de marcadores. Las características que se explotaron fueron:

- Reducción de la escena original por una matriz 25 veces menor.
- Búsqueda de esquinas para la elección de un candidato a marcador.

Aprovechando estas dos características se procedió al diseño del algoritmo de búsqueda de marcadores sobre la escena. La estrategia consiste en:

- Umbralización y reducción de la escena original.
- Búsqueda de esquinas exteriores.

- Reconocimiento laberíntico del interior del marcador.

Puesto que el controlador de la cámara genera eventos de forma asíncrona en los que lanza el flujo de vídeo, es necesario que la búsqueda se realice en un hilo paralelo de forma controlada, es decir, existiendo una sola instancia de cada búsqueda ejecutándose en paralelo. Esto es también necesario debido a la naturaleza propia del sistema que permite interacción del usuario, que genera eventos asíncronos también sobre el sistema.

Umbralización y reducción

Este proceso es el primero que se lleva a cabo durante el ciclo de vida del reconocimiento. Para mejorar el rendimiento del sistema, se aprovecha el flujo de datos que se recibe del controlador de la cámara de vídeo. Este flujo de datos viene en formato YCbCr 420 que, como se explicó en una sección previa, ofrece la información de la escena en escala de grises en los *Alto_imagen* x *Ancho_imagen* primeros píxeles. Gracias a esta información en escala de grises, el proceso posterior de umbralización en blanco y negro es relativamente más sencillo. Para ello se establece un umbral sobre el cual los valores mayores valen 1=blanco y los menores o iguales 0=negro. Este proceso se realiza en paralelo a la reducción de la imagen por 25, calculándose así solo aquéllos píxeles que serán posteriormente analizados.

De esta tarea se genera una matriz reducida binaria con información sobre la escena que será tratada por el buscador de esquinas.

Búsqueda de esquinas exteriores

Una vez generada la matriz binaria reducida se procede a la búsqueda de las esquinas exteriores del marcador. Para ello, el único requisito que se exige en cuanto a su forma exterior es que sea un cuadrado con bordes negros dentro de un cuadrado blanco. La figura 5.13 ejemplifica esta circunstancia.

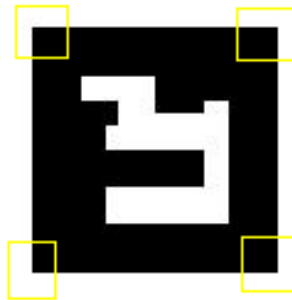


Figura 5.13. Requisitos sobre la forma exterior de los marcadores empleados.

La búsqueda de las esquinas se hace de forma análoga a la descrita en el anterior punto, con la diferencia de que en esta ocasión se buscan las esquinas negras sobre fondo blanco, siendo los valores descritos simplemente invertidos ($0 \Rightarrow 1$, $1 \Rightarrow 0$).

Si el sistema localiza un candidato, la información sobre éste (posición de las esquinas) se pasa a la siguiente fase, la de establecer, en primer lugar si es un marcador válido y, si lo fuera, determinar cuál es. En caso contrario, se informa al sistema de que la búsqueda no ha tenido éxito y se procede a analizar el último flujo de datos emitido por la cámara de vídeo.

Reconocimiento laberíntico

La última etapa del reconocimiento consiste en encontrar un patrón laberíntico en el interior del candidato extraído. Al utilizar el término laberinto, nos referimos a que el recorrido que sigue el algoritmo en el interior del candidato se asemeja a una estrategia de búsqueda de la salida en un laberinto.

En esta etapa se utiliza la región de la matriz original del flujo de vídeo inscrita en las 4 esquinas detectadas. La razón por la que se vuelve a utilizar una resolución mayor es que al utilizar una matriz reducida se perdía precisión en la búsqueda, perdiéndose detalles importantes del interior de los marcadores.

Para la descripción del algoritmo se utilizará una analogía entre lo que realmente se procesa y el recorrido laberíntico que se realiza, con el ánimo de esclarecer al lector su funcionamiento.

La estrategia de búsqueda es la siguiente: comenzando por la esquina superior izquierda se procede a la búsqueda del primer píxel blanco que se encuentre en el interior del candidato y que esté más próximo a dicha esquina. A partir de ese píxel se recorre la matriz de forma horizontal hasta alcanzar el primer punto negro o primera pared vertical por el camino. Una vez encontrado, la búsqueda continúa en dirección vertical por las dos direcciones. Si la distancia del primer punto negro en dirección hacia arriba es mayor que hacia abajo, el marcador no es válido y se termina la búsqueda. En caso contrario de que la pared de abajo esté más alejada que la de arriba el proceso continúa.

Como el lector habrá podido observar, en el interior del marcador se establece un requisito fundamental para que pueda ser procesado como tal, y es que durante el recorrido interior, las primeras paredes con las que se encuentra el sistema sean la pared de la derecha y la pared de abajo. La figura 5.14 muestra un ejemplo de este requisito.



Figura 5.14. Requisito sobre la forma interior de los marcadores empleados.

Una vez superados los requisitos de búsqueda, se procede a la extracción del tipo de marcador visualizado. Para ello, se continúa recorriendo el laberinto interior, en primer lugar en sentido horizontal, extrayendo una distancia mayor, izquierda o derecha; a continuación, en sentido vertical con recorrido mayor hacia arriba o hacia abajo; después otra vez en sentido horizontal... Y así hasta llegar al límite del laberinto. La figura 5.15 muestra varios ejemplos de los marcadores que reconoce el sistema.

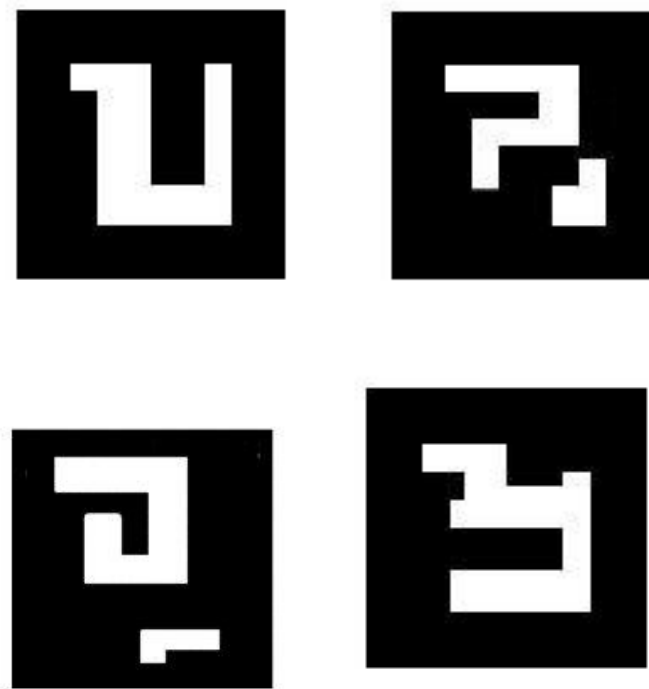


Figura 5.15. Marcadores reconocidos por el sistema.

El sistema determinará el tipo de marcador en función de los máximos recorridos en cada dirección y en cada momento. Esta estrategia permite que el sistema detecte de forma un número exponencial de marcadores, siendo 2 el número máximo si sólo analiza en la primera dirección, 4 si analiza también la segunda, 8 con la tercera, etc.

Una vez localizado qué marcador es y dónde se encuentra, ambas informaciones son enviadas a un gestor de superposición de contenidos que será descrito más adelante.

5.2.5 El problema de las rotaciones y transformaciones: la solución definitiva

A pesar de los buenos resultados obtenidos con el algoritmo recién descrito, aún continuaba un problema muy importante: el sistema era incapaz de reconocer los marcadores cuando éstos aparecían rotados o cuya proyección no fuese paralela al eje z de la cámara.

Para encontrar una solución se replanteó una vez más el problema y se analizaron alrededor de 50 matrices que correspondían a la información con la que el sistema realizaba el reconocimiento. En la figura 5.16 se pueden ver algunas de estas matrices.



Figura 5.16. Extracto de matrices que utiliza el sistema para el reconocimiento. En color verde aparecen los 0s de la matriz que se corresponden al negro y en blanco los 1s que se corresponden al blanco tras realizar la umbralización.

La primera estrategia que se pensó en utilizar fue la de mantener el algoritmo empleado anteriormente y modificar la dirección del recorrido sobre la matriz, siendo ésta descendente en 30°, 45° y 60°. No obstante, esta estrategia presentaba dos inconvenientes:

- En primer lugar el número de operaciones se incrementaba notablemente, ya que, además de realizar la misma búsqueda hasta cuatro veces, el recorrido de las matrices en esas direcciones produce diversos saltos de memoria debido a la naturaleza de las propias matrices.
- No era del todo fiable, ya que la representación de los marcadores sobre las matrices rara vez se ajustaban mediante una recta entre sus esquinas siguiendo cualquiera de las cuatro direcciones utilizaba, siendo bastante elevado el error de reconocimiento.

A pesar de todo, mediante un exhaustivo análisis de las muestras se localizó un patrón común que era independiente de la posición que tuviese el marcador sobre la escena. Este patrón era encontrar un punto negro (0 en la matriz) que al buscar en las direcciones 90°, 180°, 270° y 360° (esto es, hacia arriba, izquierda, abajo y derecha) cumpliera con que se encontrase, al menos, tres saltos del blanco al negro y viceversa en al menos tres de las direcciones. La figura 5.17 ilustra este fenómeno.



Figura 5.17. Patrón que debe cumplir un punto para comenzar la exploración. A la izquierda imagen de un punto que lo cumple. A la derecha se puede observar un punto de la matriz que no lo cumple.

Sin embargo, ese patrón no era suficiente para determinar que la vecindad del punto fuese un marcador. En función del contexto de la escena o ruidos en la iluminación podían tomarse otros puntos de la matriz como candidatos a ser marcadores. Para paliar este problema, se decidió obtener todos los puntos que ajustaban con el patrón y se descubrió que en el interior de los marcadores se almacenaban el mayor número de puntos candidatos. En la figura 5.18 se muestra el diagrama de actividad de este proceso.

Para obtener un ajuste lo más fiable de esta búsqueda se implantó una estrategia de elección de conjuntos de puntos candidatos. A cada punto que el algoritmo va encontrando se asigna al conjunto cuyos puntos estén más próximos a él. Si ninguno de los conjuntos tiene puntos próximos se crea un nuevo conjunto y se le mete dentro.

Una vez determinados los conjuntos el sistema elige el conjunto que tenga más puntos que cumplen el patrón.

El siguiente paso del algoritmo consiste en analizar los puntos blancos vecinos del conjunto ganador que cumplan el mismo patrón que el recién descrito. Este proceso aporta al algoritmo una medida de fiabilidad adicional para la localización del marcador.

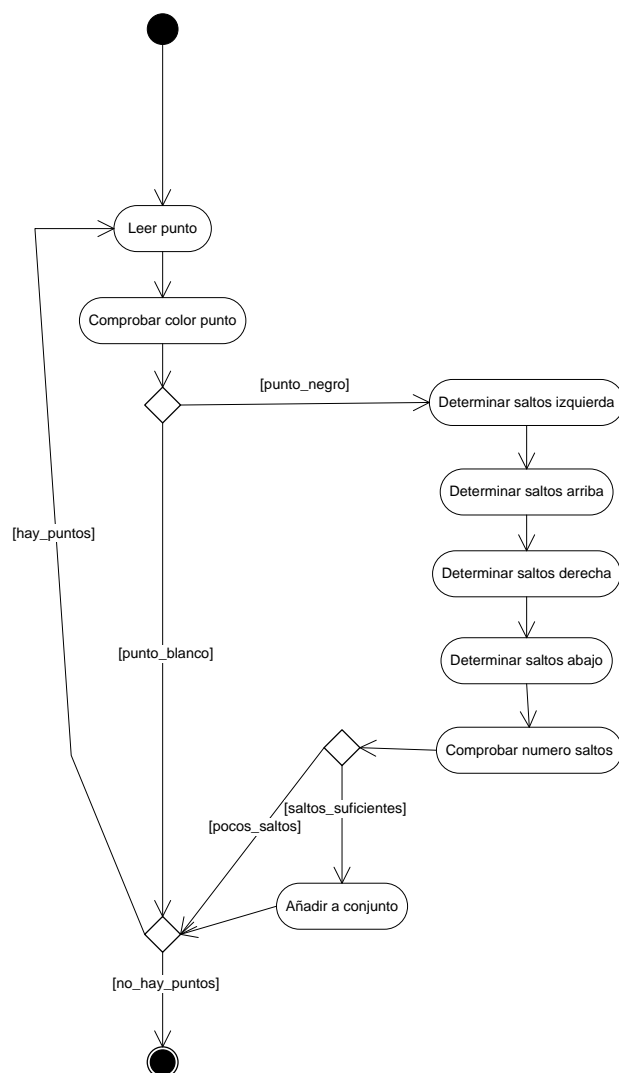


Figura 5.18. Diagrama de actividad que muestra el proceso de selección de puntos negros candidatos.

Si el paso anterior es satisfactorio se extrae cualquiera de los puntos negros del conjunto seleccionado. A partir de ese punto se realiza una búsqueda de las cuatro paredes exteriores del marcador en las cuatro direcciones múltiplos de 90°. Una vez localizados se almacenan temporalmente para el siguiente paso del algoritmo.

Una vez determinadas las cuatro paredes el algoritmo realiza una búsqueda paralela en cada punto para determinar el contorno del marcador. Para ello cada punto traza una ruta en la que busca el punto siguiente del exterior del marcador en sentido horario hasta alcanzar de nuevo el punto de origen, es decir, a sí mismo. La algoritmo da por satisfactoria la búsqueda si y sólo si en la misma iteración dos puntos llegan a su destino. Esto se debe a que, si dos puntos distintos pertenecen al contorno, el número de puntos por los que pasan es el mismo.

El último paso del algoritmo en la búsqueda de los marcadores consiste en determinar las cuatro esquinas a partir de la ruta obtenida en el paso anterior. Con esto se consigue determinar la orientación y rotación del marcador y eliminar muestras que erróneas.

Una vez obtenido el candidato a marcador, se le aplica un último filtro para determinar qué marcador es el visualizado. Para realizar este paso se cambió el diseño de los marcadores. La estructura de éstos es la siguiente:

- Se pinta un cuadrado negro.
- Sobre el cuadrado negro se pinta un cuadrado blanco más pequeño y concéntrico.
- Sobre el cuadrado blanco se dibujan conjuntos denominados islas de color negro.

La figura 5.19 muestra varios ejemplos de los nuevos marcadores.

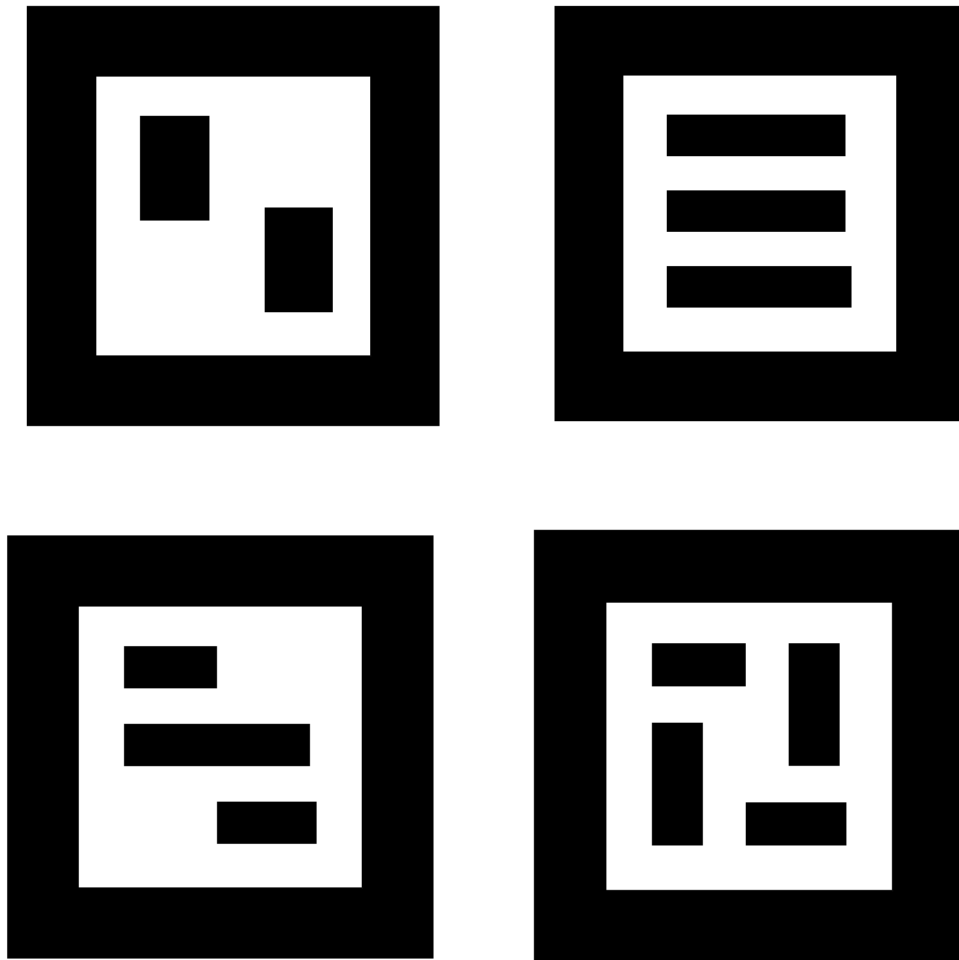


Figura 5.19. Diseño de la última versión de marcadores.

La estrategia para determinar qué marcador es el visualizado consiste en extraer los conjuntos cuyos puntos sean adyacentes entre sí del conjunto anteriormente localizado. Una vez determinadas las “islas negras” que tiene el interior del marcador se calcula la relación de tamaño que existe entre ellas que, unido al número de islas, establece el patrón de marcador al que pertenece.

La figura 5.20 muestra un diagrama de actividad que refleja, a modo de resumen, el algoritmo de búsqueda.

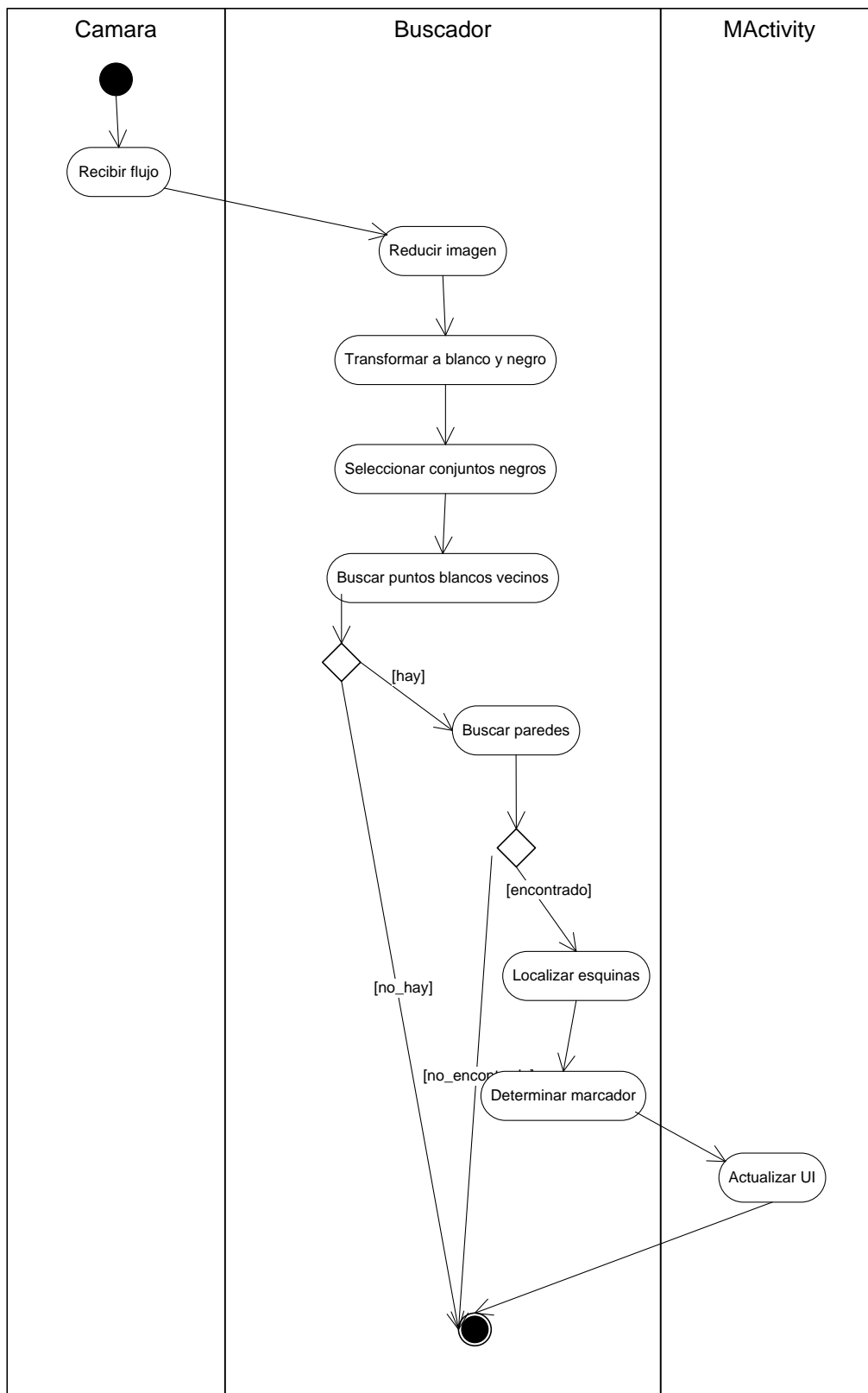


Figura 5.20. Diagrama de actividad del algoritmo de búsqueda

5.3 Técnicas de aumento

La siguiente tarea que era necesario abordar era la de la superposición de contenidos digitales sobre la escena real captada por la cámara.

Una vez reconocido el escenario que se desea ampliar gracias al reconocimiento del marcador adecuado, el sistema debía elegir qué información era más adecuada mostrar en ese momento.

En un gran número de aplicaciones y sistemas de realidad aumentada se utiliza OpenGL para dibujar figuras en 2D y 3D. A pesar de que en nuestra aplicación no se consideró oportuno incluir este tipo de figuras geométricas, ya que los contenidos iban a estar más orientados a la impresión de texto y material fotográfico, se decidió realizar una prueba con el estándar gráfico soportado por las arquitecturas Android, OpenGL ES.

Los resultados fueron bastante óptimos, ya que el pintado de figuras se realiza en un modalidad con privilegios dentro del sistema, ofreciendo tiempos de respuesta bastante bajos y acordes con la necesidad de que el sistema funcionase en tiempo real. En la figura 5.21 se puede ver un extracto de los primeros usos de OpenGL ES para superposición de contenidos.

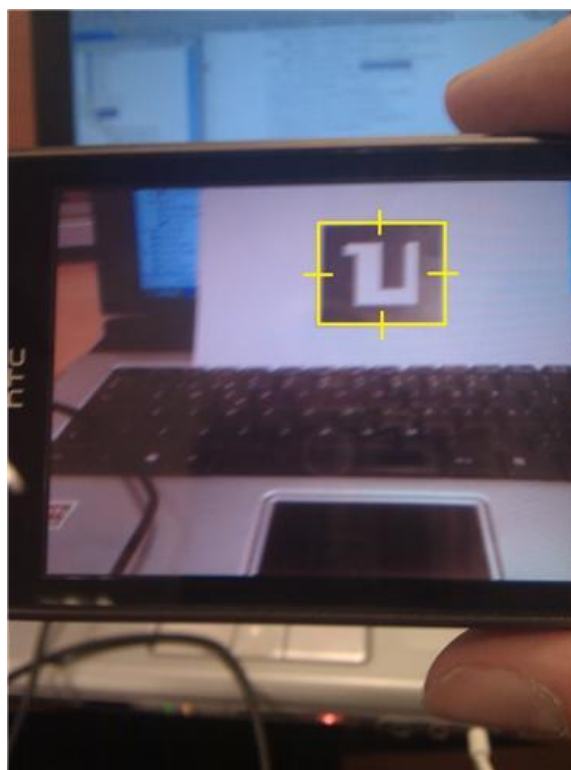


Figura 5.21. Ejemplo de aumento sencillo utilizando OpenGL ES.

No obstante, la velocidad de procesamiento que aportaba OpenGL ES no era su único punto fuerte. La sencillez para pintar dibujar geométricas que ofrece su API es enorme, pudiendo realizar figuras en 3D en muy poco tiempo. Además permitía variar el punto de vista desde el que se mira la figura sin tener que preocuparse de cómo es el lado que se está visualizando en ese momento.

A pesar de estas grandes ventajas, OpenGL ES tenía un inconveniente muy grave. Nuestra necesidad de cubrir los aumentos mediante contenidos textuales era una tarea ardua y pesada para abordarse mediante esta librería. La razón de esto es que OpenGL ES no está pensado para tal propósito. Esto no significa que no se pudiese hacer, sino que la construcción de texto mediante esta librería no era una tarea precisamente sencilla, ya que OpenGL sólo reconoce un par de objetos básicos, que son:

- Figuras geométricas basadas en vértices y aristas.
- Imágenes basadas en texturas.

Por tanto, las dos únicas formas de pintar contenido textual era o bien mediante la preparación geométrica de cada letra y su posterior pintado, o bien mediante la generación de la textura de cada texto, tarea nada sencilla.

Fue por esta razón principalmente por la que se decidió finalmente desechar esta vía de trabajo. Además, desde un primer momento se pensó que la construcción y administración del sistema fuera lo más sencillo posible, de forma que quien estuviera interesado pudiera continuar con el trabajo y, por desgracia, OpenGL no es una tecnología que cuente con tantos adeptos como, por ejemplo, Java.

Una vez decidido que OpenGL ES no sería la forma en la que se realizaría la superposición de contenidos se optó por utilizar el modelo de capas de las interfaces de usuario de Android.

El modelo de Android se basa en una estructura jerárquica de vistas anidadas unas con otras. La figura 5.22 muestra la estructura jerárquica de los interfaces de usuario en Android.

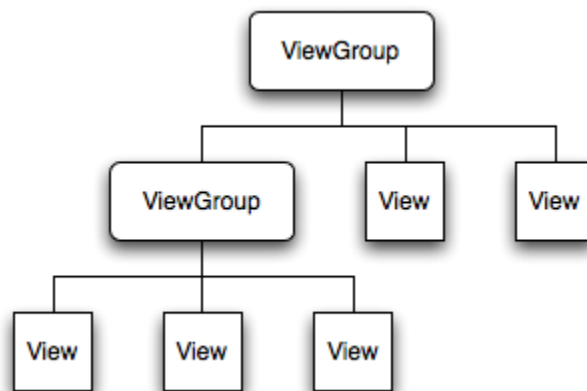


Figura 5.22. Estructura jerárquica de capas del interfaz de usuario de Android.

Al hilo de lo comentado recientemente, una de las características con las que se quería que contara el sistema era su escalabilidad y mantenimiento. La manera en que Android maneja los componentes de los interfaces de usuario es muy cómoda para los programadores, ya que se permite la definición de la interfaz en un fichero XML denominado *main.xml*. En este fichero se pueden modificar atributos como el ancho y alto de una vista, su posición, color, etc.

Otro de los motivos por los que se utilizó la forma en que Android muestra los contenidos fue la facilidad para manejar de forma dinámica cualquier contenido textual o fotográfico. Para manejar el primer tipo de datos basta con utilizar un objeto *TextView* e ir modificando su posición o contenido de forma dinámica según la situación del sistema. De igual manera se realizan las operaciones análogas con los objetos multimedia que, en el caso de las imágenes,

se pueden alojar en un objeto *ImageView* o en un *Button* que permita recibir eventos del usuario. La figura 5.23 muestra un ejemplo del uso de la estructura jerárquica de Android para el manejo de la interfaz de usuario.

Por último, cabe destacar que Android realiza una excelente gestión de las actualizaciones en la interfaz de usuario. Para ello maneja objetos del tipo *Handler* que gestionan de forma asíncrona dichas actualizaciones mediante el envío de mensajes entre procesos, de forma que el sistema escoge el momento más adecuado y seguro para realizar las funciones necesarias, de forma que el programador no tenga que preocuparse por tan tediosas y complicadas tareas.

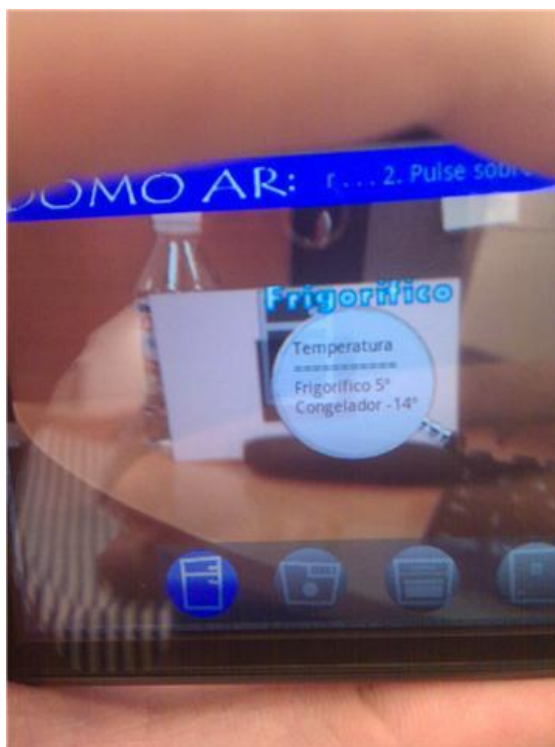


Figura 5.23. Ejemplo de uso de capas en Android para superposición de contenidos en un proyecto paralelo de realidad aumentada en hogares digitales.

5.4 Arquitectura del sistema

En esta sección se describe la arquitectura y diseño del sistema de realidad aumentada implantado de forma conjunta con Telefónica I+D.

Como ya se vio en un apartado anterior de la memoria, desde el punto de vista de la relación entre agentes funcionales, se diseñó un sistema que constaba de los siguientes módulos:

- I. Aplicación local instalada en el teléfono móvil.
- II. Sistema de contenidos de Telefónica I+D.
- III. Sistema de transformación de voz a texto de Google.

Puesto que la arquitectura global y la definición de los agentes externos ya fueron abordadas en el capítulo 4 de esta memoria, en esta sección se explicará la arquitectura interna de la aplicación instalada en el teléfono móvil.

La arquitectura interna está compuesta por 4 módulos básicos, que son:

1. Interfaz de usuario.
2. Gestión de búsquedas.
3. Gestión de comunicaciones.
4. Paquete de utilidades del sistema.

La figura 5.24 muestra la arquitectura interna de la aplicación.

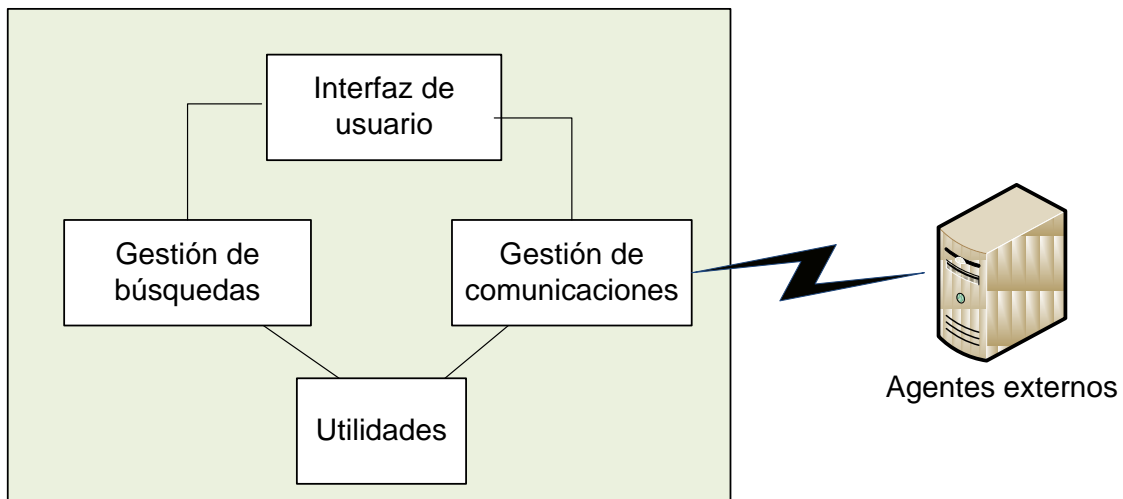


Figura 5.24. Arquitectura interna de la aplicación local.

A continuación se explica detalladamente el contenido y funcionalidad de cada uno de los módulos que componen la arquitectura interna de la aplicación local.

5.4.1 Interfaz de usuario

Este módulo es el encargado de gestionar la correcta visualización de todos los elementos visuales de la aplicación. Existen dos tipos de elementos visuales. En primer lugar se encuentran aquellos cuya naturaleza radica en la funcionalidad del sistema, como botones, menús, etc. Este tipo de elementos están destinados a producir eventos asíncronos sobre el sistema, en ocasiones para modificar el interfaz de usuario y en otras para producir eventos externos o entre procesos.

El otro tipo de elementos visuales son aquellos relacionados con los aumentos de información y aquellos eventos producidos a raíz de estas situaciones.

Por otro lado, dentro de este módulo está incluido el tratamiento de cuatro tipos de eventos:

- I. Eventos de usuario. Producidos por el usuario al pulsar los botones de la aplicación.
- II. Eventos de la cámara de vídeo. Producidos por la recepción de un nuevo flujo de vídeo.
- III. Eventos de pintado entre procesos. Producidos por la actualización del estado de búsqueda, la aplicación entra en un estado dinámico de refresco de los componentes de la interfaz de usuario.

IV. Eventos de recepción de información externa. Producidos por la recepción de contenidos no actualizados desde el servidor externo de contenidos.

Por otro lado, como se comentó con anterioridad, el núcleo de Android no permite el manejo de los componentes de la interfaz de usuario desde cualquier proceso. Para poder actualizar los componentes, la actualización debe hacerse o bien desde el hilo principal de la interfaz de usuario o bien invocando un objeto *Handler*. Este objeto actúa como receptor de mensajes de otros procesos internos, preparando los datos de actualización que se encargará de comunicar él mismo al hilo principal.

Por último, cabe destacar que la arquitectura Android para aquellas aplicaciones que requieran de un interfaz de usuario delega el hilo principal en las clases *Activity*, que actúan a su vez de hilo principal y contenedores de los componentes visuales. En la figura 5.25 se expone el ciclo de vida de un objeto *Activity*.

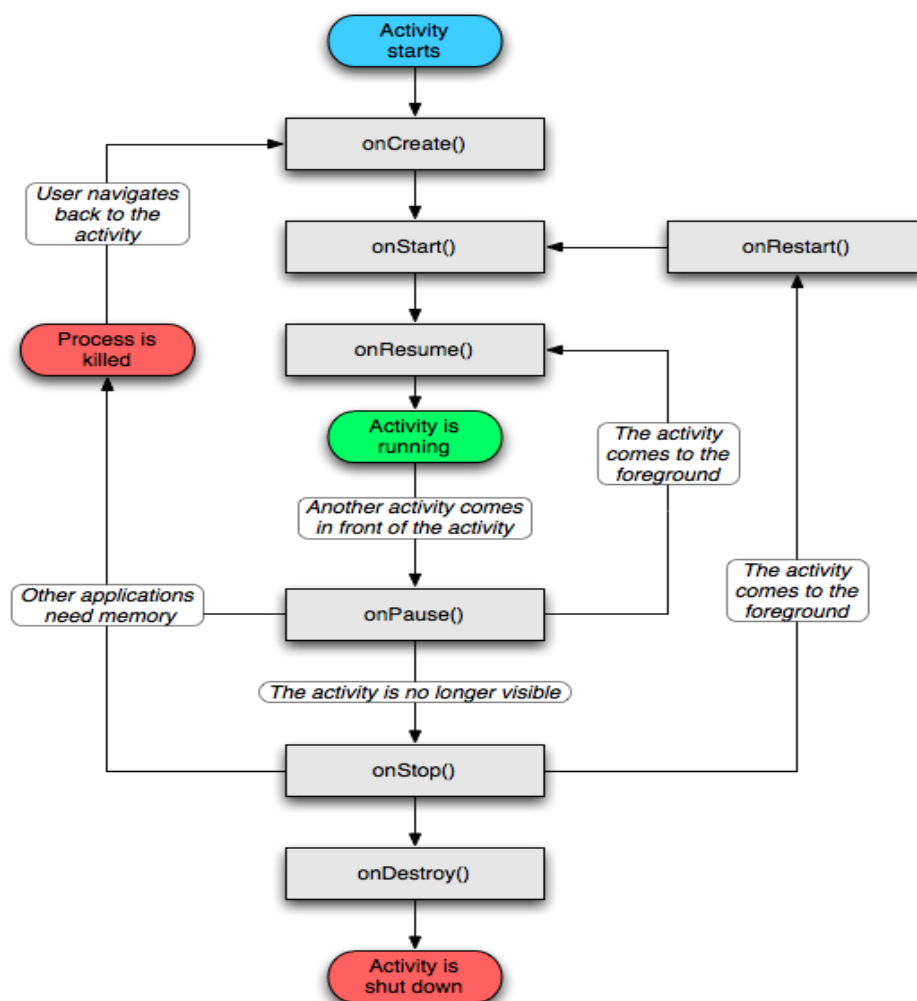


Figura 5.25. Ciclo de vida de una *Activity* en Android.

Una vez vistos los componentes que integran el módulo interfaz de usuario se adjunta de forma gráfica su estructura, tal y como puede apreciarse en la figura 5.26.

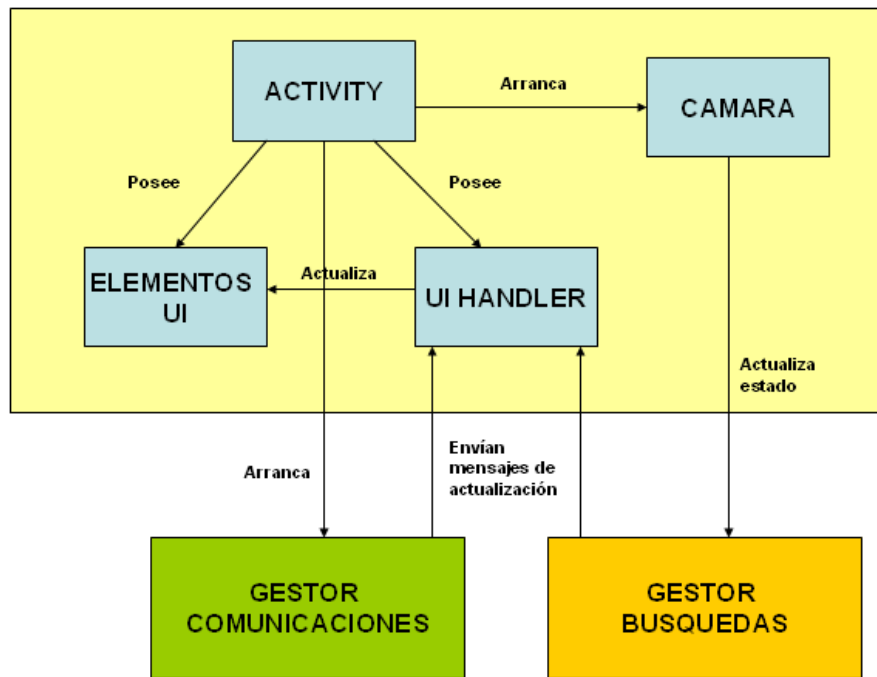


Figura 5.26. Estructura interna del módulo Interfaz de Usuario.

5.4.2 Gestión de búsquedas

Este módulo se encarga de encapsular toda la lógica relacionada con la búsqueda del marcador en el flujo de vídeo recibido desde la cámara. El algoritmo de búsqueda ha sido descrito en una sección previa, por lo que en este apartado nos centraremos en la descripción estructural de dicho módulo.

Al igual que sucedió en el diseño de todo el sistema, uno de los objetivos prioritarios era que la aplicación fuese escalable. Si bien, como acaba de mencionarse, esta tarea fue importante en el diseño del sistema, aún más si cabe dentro de este módulo que a continuación se explica. El motivo por el que es más relevante aquí se debe a la naturaleza mejorable de este módulo, ya que se quería dejar una puerta abierta ante posibles mejoras que pudieran surgir sobre el algoritmo de búsqueda.

El módulo de gestión de búsquedas está compuesto por los siguientes componentes:

- **BuscadorGeneral**. Esta clase es un *Singleton* que se encarga de recibir los eventos de la cámara de vídeo y decidir si se debe realizar o no una búsqueda del marcador.
- **HiloBúsqueda**. Este proceso se encarga de gestionar los estados de la búsqueda en cada momento. La razón por la que es un hilo es evitar el bloqueo de la interfaz de usuario, realizando la búsqueda en segundo plano.
- **BuscadorZonal**. Se encarga de la búsqueda de un marcador en una región específica de la imagen. Esta clase es invocada por *HiloBúsqueda*.
- **AjustadorEsquinas**. Esta clase está diseñada siguiendo el patrón *FactoryMethod*. Se encarga de proveer de un conjunto de funciones de comparación de patrones de las esquinas del marcador.
- **AjustadorLaberintico**. Al igual que la anterior, esta clase también utiliza el patrón *FactoryMethod*. Se encarga de realizar el ajuste de patrón laberíntico del interior de la matriz del posible marcador.

- **ControladorPintado.** Esta clase es un *Singleton* que se encarga de gestionar el proceso de preparación del mensaje de pintado pertinente hacia el *Handler* del paquete Interfaz de Usuario. Recibe el resultado de la búsqueda, preparando todos los datos necesarios para llevar a cabo el aumento sobre la escena real.
- **Marcador.** Clase que contiene la información básica relativa a un marcador, tales como su posición, tamaño y tipo. También tiene una serie de estrategias de pintado asociadas.
- **EstrategiaPintado.** Esta es una clase abstracta que define el método *seCumple()*. La idea de esta clase abstracta es definir una familia de clases que permita establecer una inteligencia adicional a la aplicación a la hora de pintar o no un marcador en caso de que en una iteración no hubiese sido localizado.
- **EstrategiaZonaCentral.** Esta estrategia extiende la definición de la clase *EstrategiaPintado*. Su funcionalidad consiste en establecer si un marcador estaba en el instante anterior en una zona central, permitiendo que vuelva a ser aumentada su información en el instante actual, ya que supone que el marcador no ha podido salirse de la escena.
- **EstrategiaTiemposConsecutivos.** Esta estrategia, al igual que la anterior, también extiende la definición de *EstrategiaPintado*. La idea de esta estrategia es evitar el parpadeo del aumento de información si en un instante no ha sido localizado pero en los anteriores sí y se prevé que en el actual deba estar también en la escena.
- **Lado.** Enumeración que permite disponer de un interfaz cómodo de manejo de las direcciones encontradas durante la búsqueda laberíntica. Sus valores son: ARRIBA, ABAJO, DERECHA, IZQUIERDA.

La figura 5.27 muestra la estructura del módulo Gestión de Búsquedas.

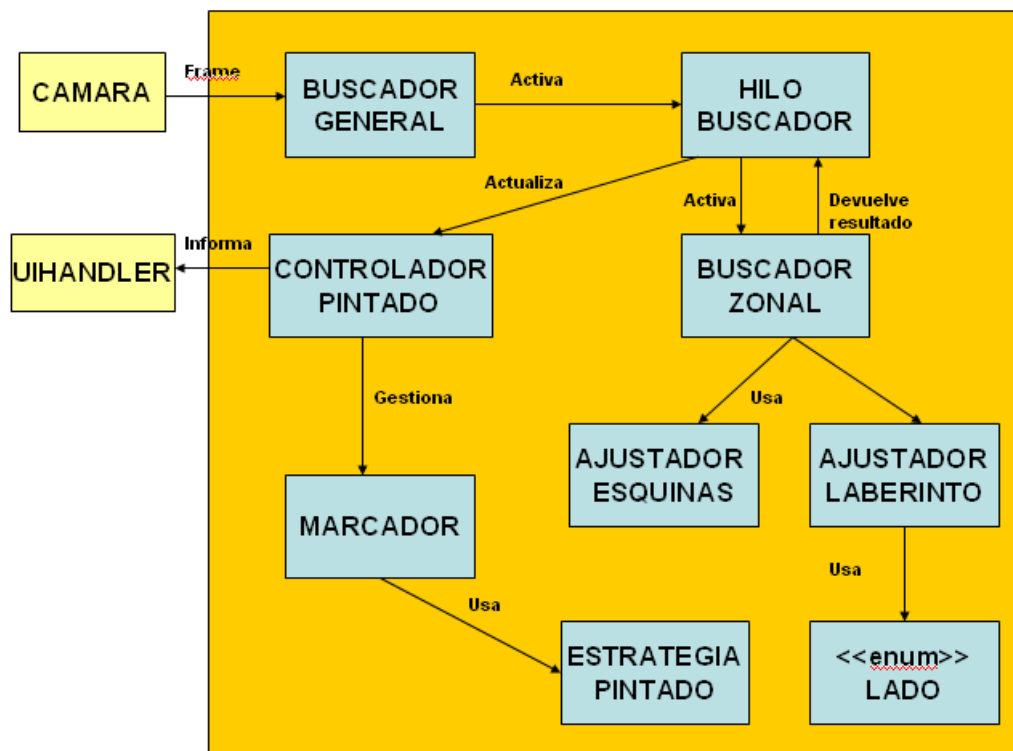


Figura 5.27. Estructura interna del módulo Gestión de Búsquedas.

5.4.3 Gestión de comunicaciones

Este módulo se encarga de solicitar al servidor la información de aumento que se mostrarán como información extra al aumento básico. Además, también se encarga de la comunicación con el servicio voz a texto de Google.

Para no sobrecargar la red, se decidió basarse en la filosofía de AJAX [ZAK 06], es decir, realizar peticiones asíncronas al servidor de forma periódica para el caso de la información extra.

En el caso del servicio voz a texto de Google, existe una clase que se encarga de escuchar las peticiones de voz y de procesar la solicitud, devolviendo la respuesta del servidor en forma textual, que será utilizada para invocar funcionalidades que se encuentren dentro de la lógica del programa, para lo cual la orden hablada debe ser una orden conocida por el sistema.

El módulo está compuesto por las siguientes clases:

- **GestorPeriódico.** Se encarga de realizar cada dos minutos la activación del Cliente para que se comunique con el servidor de aumento para preguntarle si existe información nueva.
- **ClienteAumento.** Se encarga de la comunicación con el servidor de aumento y de procesar la información recibida.
- **ListenerVoz.** Escucha eventos de la aplicación relacionados con la señal de voz de entrada del usuario. Activa la instancia de la clase `ClienteVozTexto`.
- **ClienteVozTexto.** Se encarga de comunicarse con el servicio de Google voz a texto, enviándole la señal de voz y procesando la señal de texto recibida.
- **GestorOrdenes.** Recibe la orden de entrada ya transformada a texto y localiza la acción asociada a ella. Una vez localizada informa al proceso pertinente de que la acción debe realizarse lo antes posible.

La figura 5.28 muestra la estructura interna del módulo Gestión de comunicaciones.

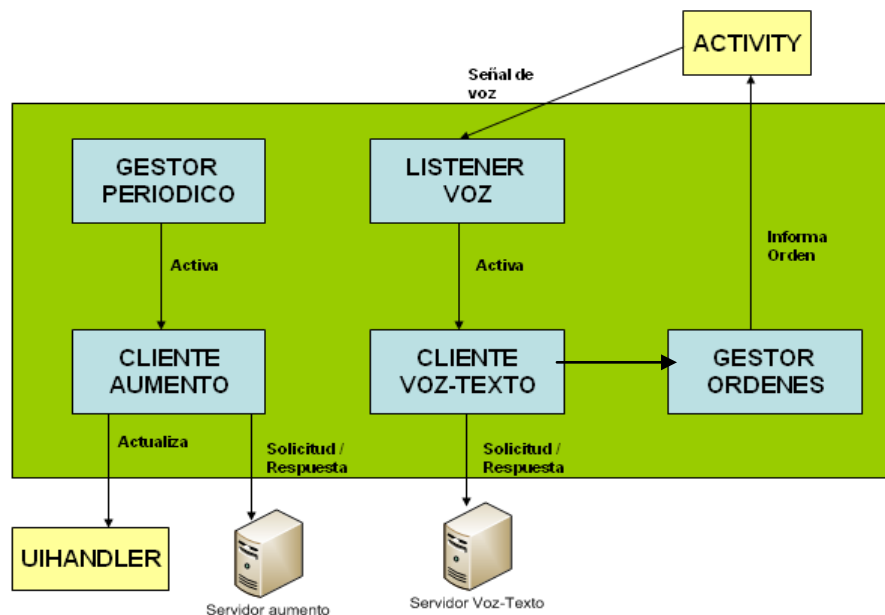


Figura 5.28. Estructura interna del módulo Gestión de Comunicaciones.

5.4.4 Paquete de utilidades

Este módulo pretende ofrecer un interfaz de trabajo para el resto de módulos del proyecto.

Las clases que lo componen son las siguientes:

- **UtilMatriz**. Interfaz que proporciona funciones de recorrido sobre matrices y transformación de éstas.
- **JSONParser**. Clase que se encarga de preparar los datos recibidos desde el servidor de aumento en formato JSON [JSON 10] en un formato adecuado para la aplicación.
- **UtilXML**. Conjunto de utilidades para recorrer y generar documentos XML. Para el recorrido se utiliza XPath [XPath 10], gracias a la potencia de su motor de búsqueda.

Destacar, por último, que todos los métodos de estas clases son del tipo *static*, de esta forma se evita tener que instanciar cada clase cada vez que se quisiese invocar a una de sus funciones.

5.5 Museo de Informática García Santesmases

Como se ha visto a lo largo de la memoria, la realidad aumentada es un campo altamente atractivo para aquellas aplicaciones y sistemas relacionados con el ámbito educativo. Un claro ejemplo del uso de esta tecnología en este campo es el proyecto “Realidad al cubo” [IPV 10], desarrollado en la Universidad Politécnica de Valencia que pretende incluir la realidad aumentada en las aulas. Este tipo de metodología educativa promete aumentar el rendimiento de los estudiantes, propiciando un estudio más atractivo e interesante gracias a la interactividad de los jóvenes con los contenidos.

Del mismo modo, el propósito del presente proyecto es incluir un sistema de realidad aumentada sobre el Museo de Informática García Santesmases de la Universidad Complutense de Madrid que permita a los visitantes obtener una cantidad de información mayor sobre cada una de las muestras de forma cómoda y sencilla. La figura 5.29 muestra algunas de las muestras del museo.



Figura 5.29. Muestras del Museo de Informática García Santesmases.

De acuerdo a estas características, el funcionamiento del sistema es el siguiente. El usuario enfoca con la cámara del dispositivo móvil sobre cualquiera de los marcadores que se encuentran colocados junto a cada muestra del museo. Una vez el sistema haya detectado la localización del marcador y a qué muestra representa, ofrece un aumento de información básico que simplemente es el nombre de la muestra enfocada. En la figura 5.30 se puede observar este proceso de aumento.

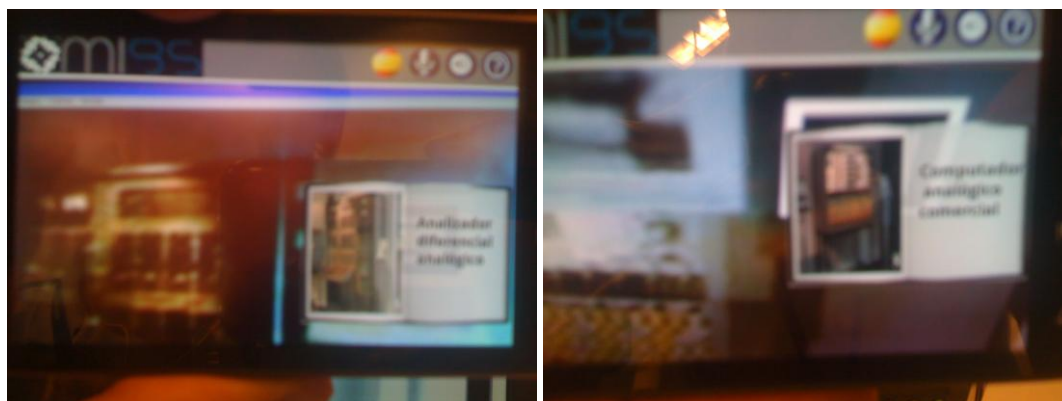


Figura 5.30. Aumento básico.

Una vez detectada la muestra, el usuario puede acceder a una información mayor mediante una simple pulsación sobre el icono de aumento básico que se encontrará encima del marcador de la muestra. Las siguientes figuras ilustran este proceso de forma clara y concisa.

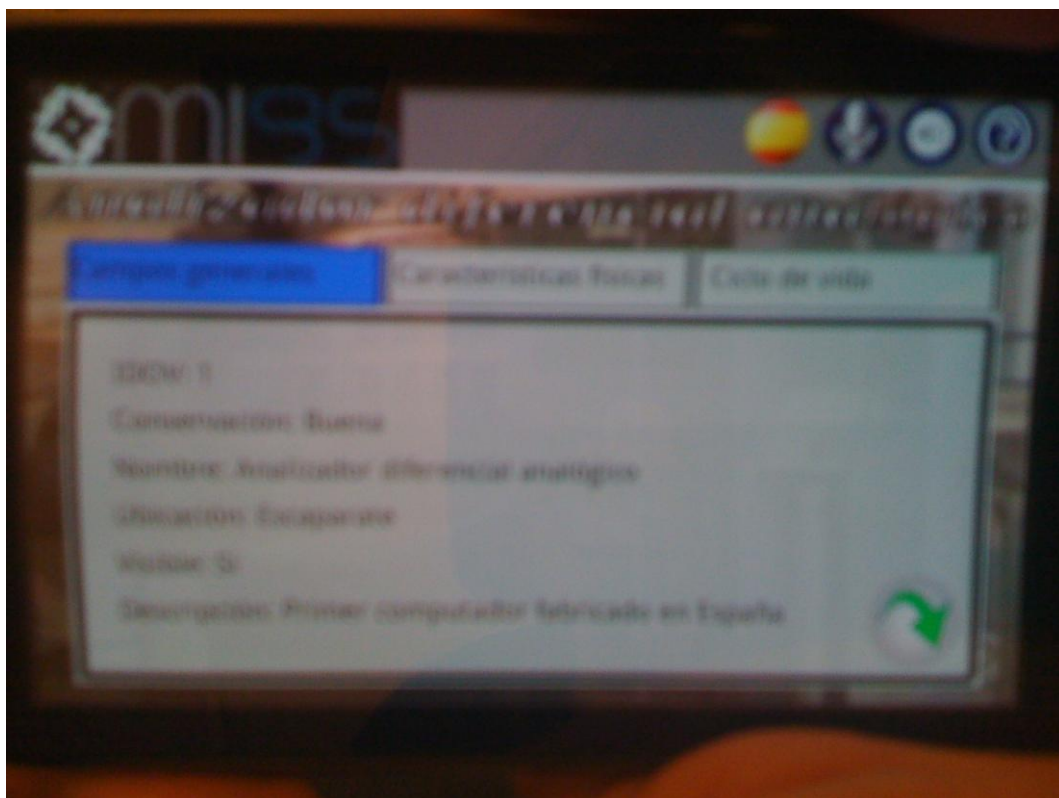


Figura 5.31. Aumento ampliado de la muestra 1.

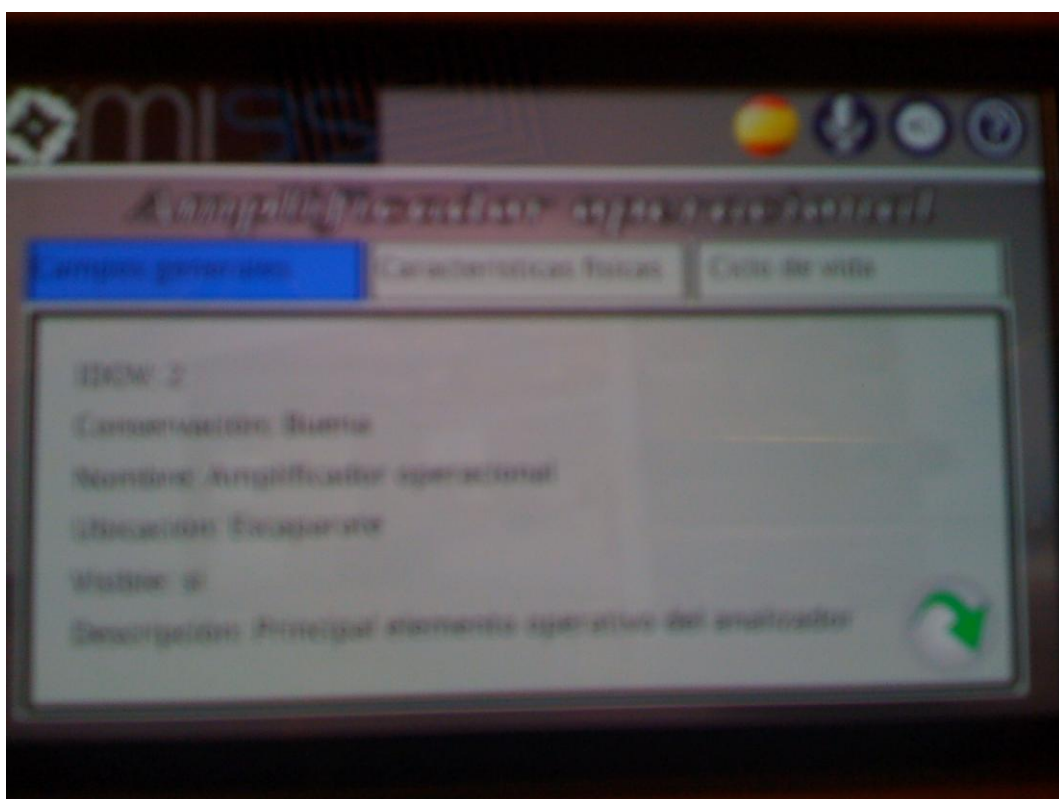


Figura 5.32. Aumento ampliado de la muestra 2.

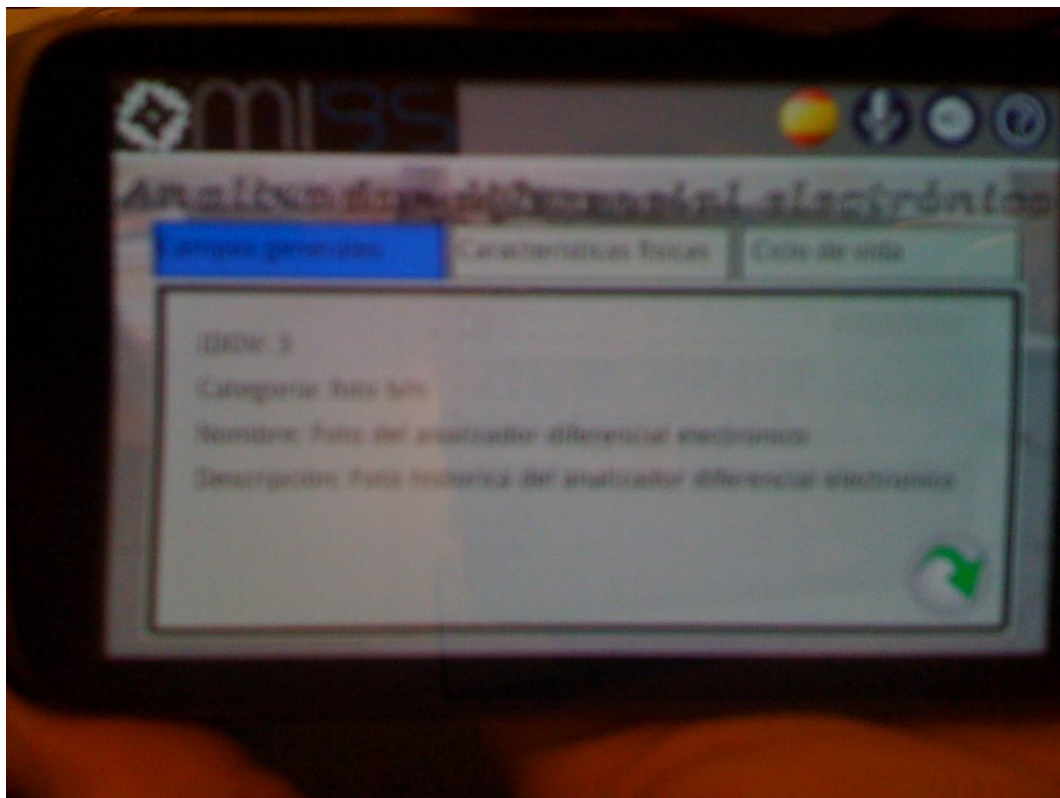


Figura 5.33. Aumento ampliado de la muestra 3.

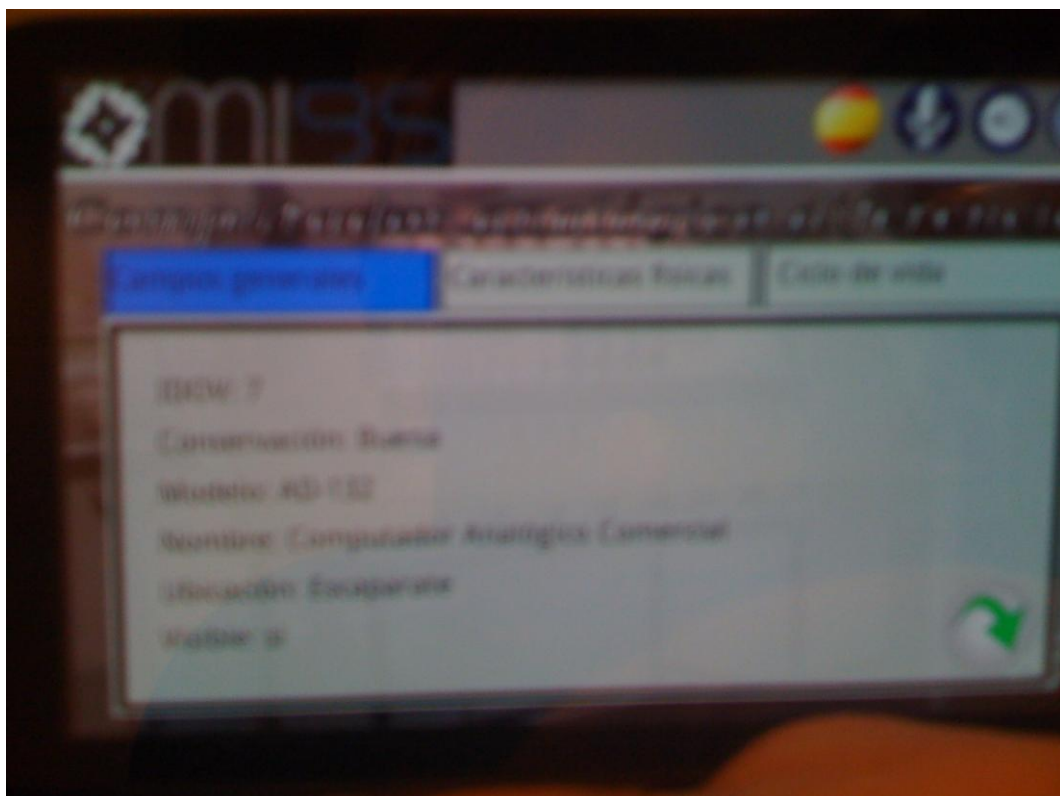


Figura 5.34. Aumento ampliado de la muestra 4.

5.6 Aplicación de domótica

Aprovechando las mejoras en los sistemas domóticos en los últimos tiempos y al interés de Telefónica I+D en este área, se desarrollo en paralelo un sistema que fuera capaz de interactuar con el hogar y que permitiese al usuario acceder a la información del mismo utilizando realidad aumentada.

Con esta idea se ha construido un prototipo de sistema que permite visualizar información simulada sobre los elementos del hogar ubicado en el edificio de Telefónica I+D en Madrid.

El escenario sobre el que se han hecho las pruebas se muestra en la figura 5.35.



Figura 5.35. Escenario de pruebas del hogar digital de Telefónica I+D en Madrid.

Como se puede observar en figura 5.x, los electrodomésticos sobre los que se ofrece aumento son: frigorífico, lavadora, horno y caldera. En la figura 5.36 se muestran los cuatro electrodomésticos, cada uno con su marcador correspondiente.

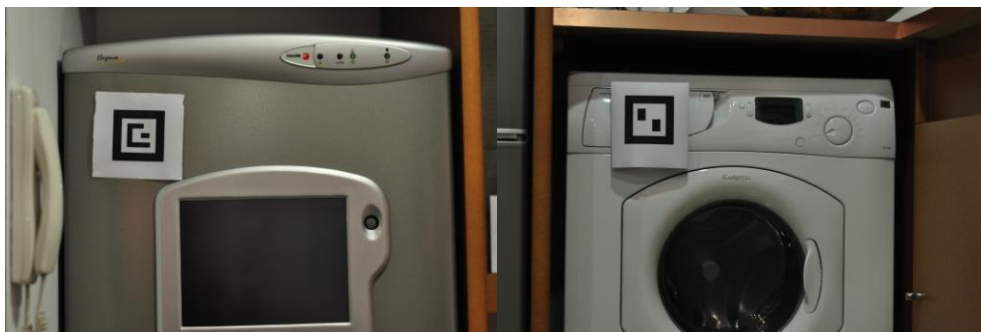




Figura 5.36. Electrodomésticos que ofrecen aumento.

A continuación se mostrará el funcionamiento del sistema con cada uno de los electrodomésticos. Las siguientes cuatro figuras muestran de qué forma el sistema ofrece el aumento básico sobre cada uno de los electrodomésticos.

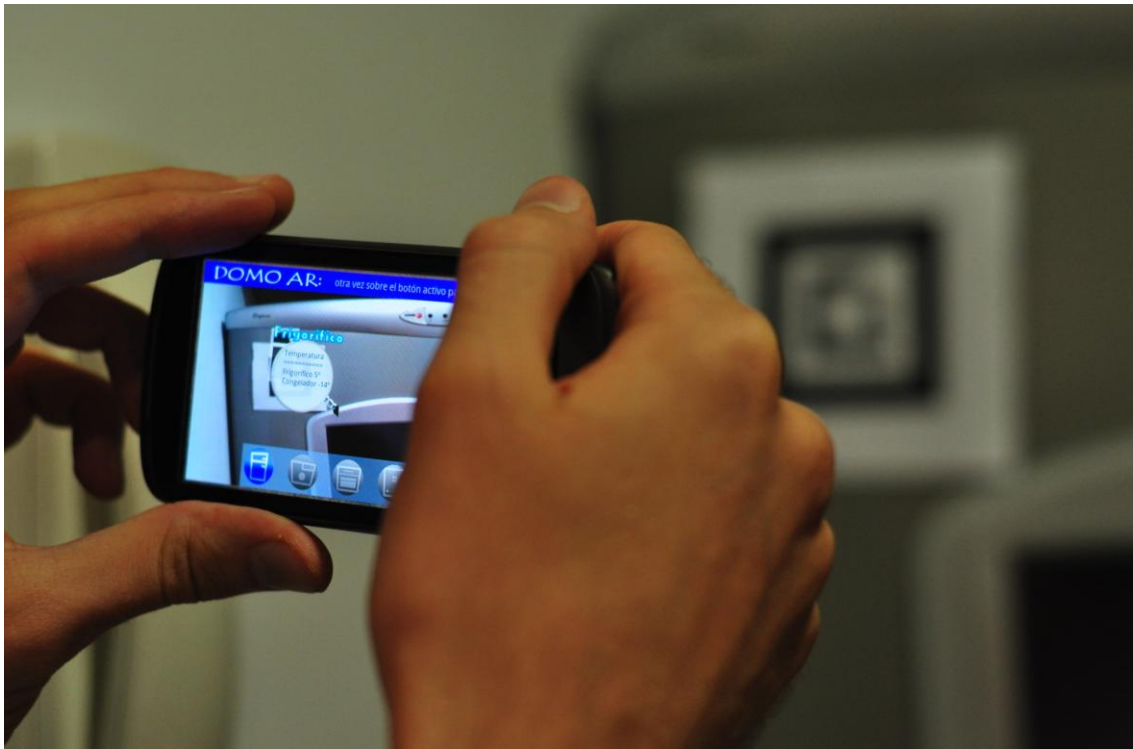


Figura 5.37. Aumento básico sobre el frigorífico.



Figura 5.38. Aumento básico sobre la lavadora.

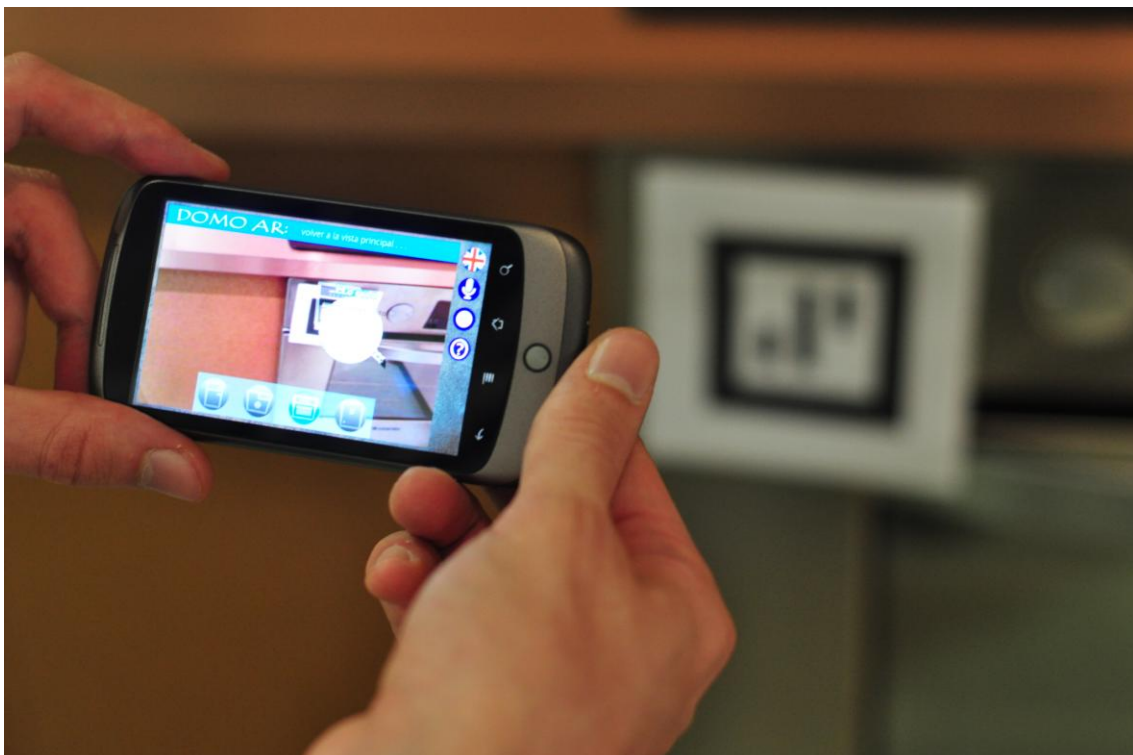


Figura 5.39. Aumento básico sobre el horno.

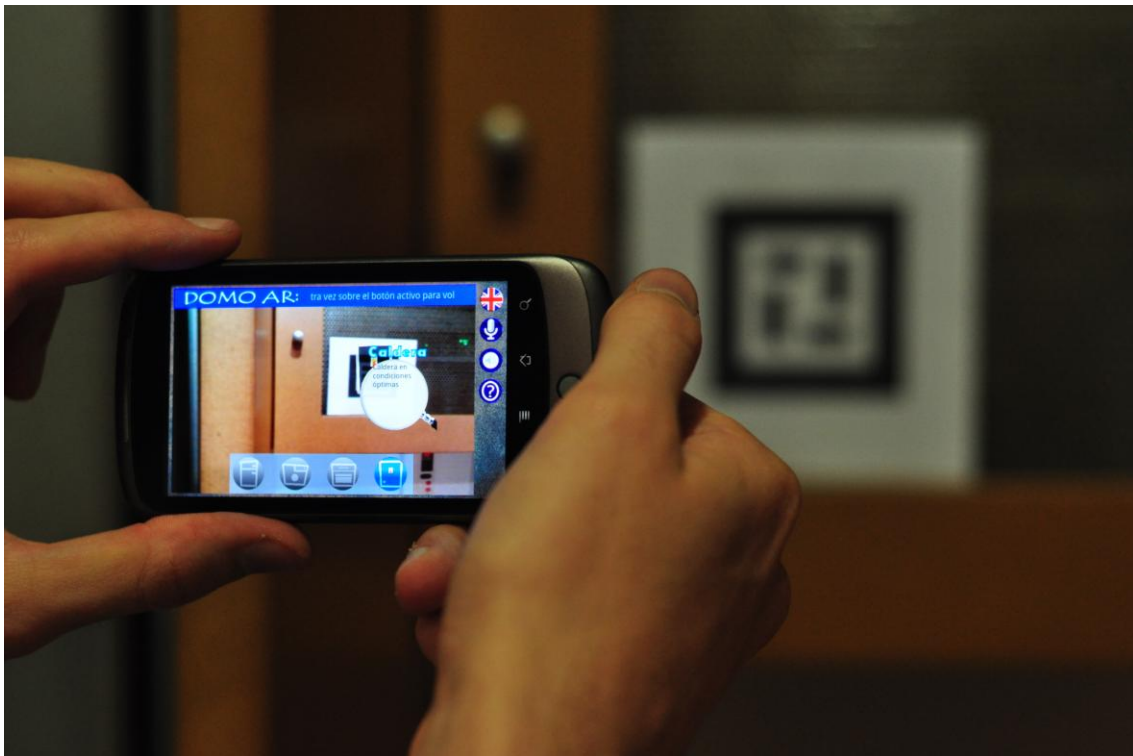


Figura 5.40. Aumento básico sobre la caldera.

Por último se muestra el funcionamiento del sistema al ofrecer el aumento más amplio de contenidos. Las siguientes figuras ilustran este proceso.

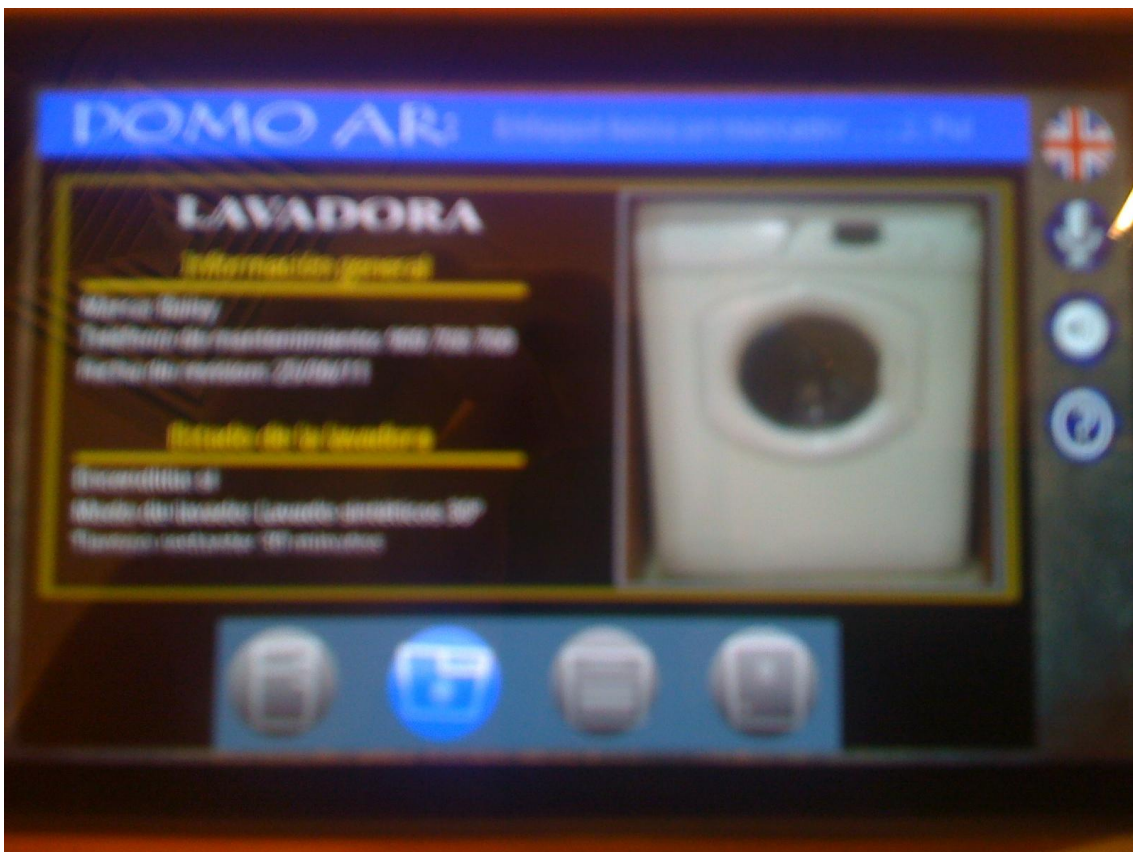


Figura 5.41. Aumento ampliado de la lavadora.

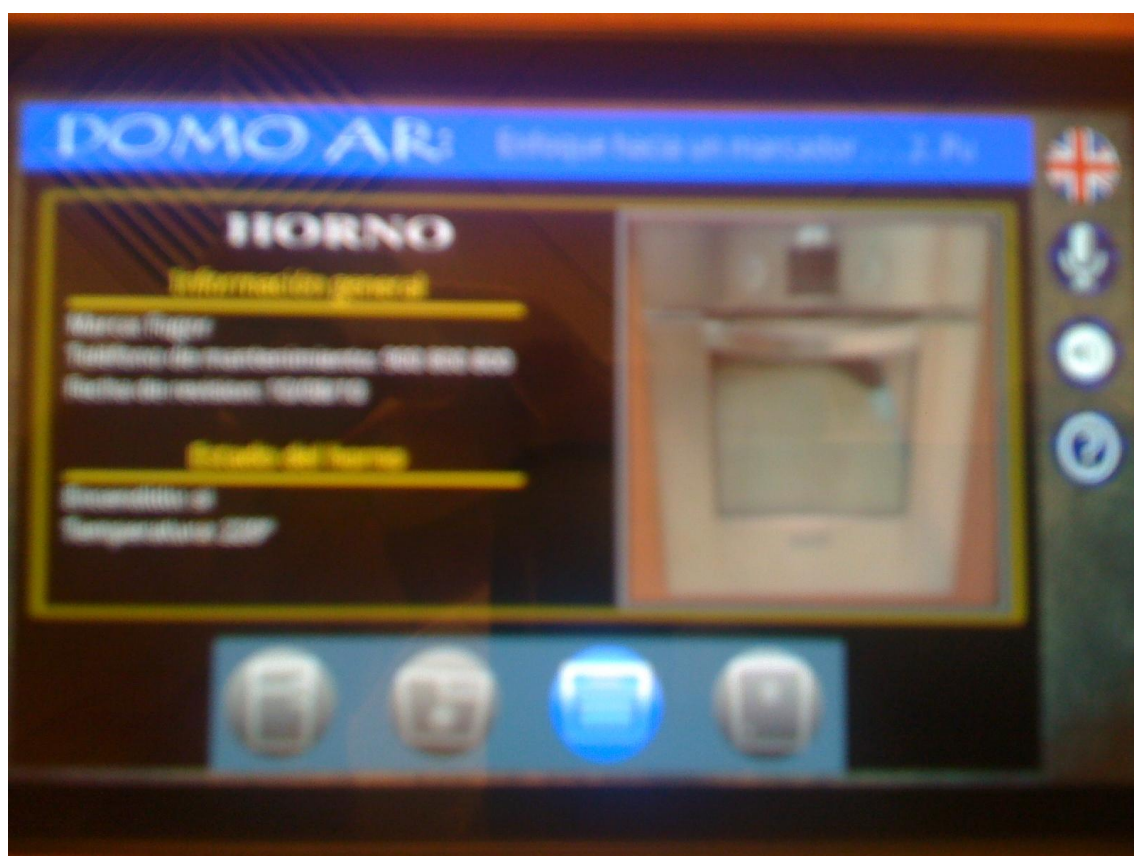


Figura 5.42. Aumento ampliado del horno.

6. CONCLUSIONES Y TRABAJO FUTURO

La realidad aumentada es un área que probablemente aún tenga mucho que decir. Aún nos encontramos en las primeras fases de aplicación de esta tecnología y múltiples campos se verían muy beneficiados de su implantación. Por ejemplo, sectores como el automovilístico ya están invirtiendo grandes sumas de dinero para implantar esta tecnología en la conducción [SEU 09].

A pesar de las grandes oportunidades que la realidad aumentada ofrece a sectores como el médico o el automovilístico, aún existe el problema de que más que ofrecer una ayuda pueda ser un peligro. El hecho de realizar tareas con una criticidad alta añadiendo información digital que no existe físicamente puede suponer que el usuario se despiste o se desoriente a la hora de realizar una tarea que está acostumbrado pero no de esa manera. Actualmente se están llevando estudios de cómo afectan los sistemas de realidad aumentada al entendimiento de éstos por el usuario y su capacidad para manejarlos de forma sencilla y clara, como es el caso de la University of South Australia [AVE 08].

Por el contrario a lo que se acaba de comentar, uno de los campos donde se ha probado la eficacia de esta tecnología es en la educación. Existen estudios que han demostrado que los niños y adolescentes aprenden mejor cuando los contenidos educativos son presentados utilizando realidad aumentada o realidad virtual.

Otro de los campos en los que la realidad aumentada aparece como una mejora importante es en los museos. La posibilidad de ofrecer animaciones relativas a los objetos de exposición puede resultar una alternativa mucho más eficaz que los actuales guías y los textos que acompañan normalmente a dichos objetos. El mayor problema que existe actualmente es el uso de dispositivos de visualización como Tablet PCs que deben utilizar los visitantes para poder contemplar la información de aumento, algo que no es muy cómodo ni agradable. En los próximos años es posible que estos dispositivos sean reemplazados por proyecciones holográficas, haciendo de la visita aumentada una experiencia más enriquecedora y agradable.

En esta memoria se han expuesto desde los diferentes componentes que los distintos sistemas de realidad aumentada suelen tener hasta las funciones que se llevan a cabo y las técnicas empleadas en ellas. También se han expuesto los dos sistemas que se han implantado, exponiendo las diferentes fases por las que ha ido pasando el proyecto desde que comenzó hasta su finalización.

Desde el punto de vista técnico, uno de los mayores problemas que conlleva el reconocimiento visual de marcadores es el manejo de las rotaciones, escalas y deformaciones de las imágenes, tales como defectos en la iluminación, brillos y sombras. Cada una de las posibles eventualidades dentro de cada uno de estos sucesos es una situación que el sistema debe contemplar. Encontrar una solución que recoja el mayor número de estos problemas es una tarea difícil y complicada.

Una de las mayores preocupaciones al principio del proyecto era el tiempo de respuesta que pudiera dar el sistema a la hora de realizar el reconocimiento visual de la escena. Este temor se acrecentó en las primeras fases al descubrir que la mayoría de técnicas de reconocimiento suponían unos tiempos de respuesta muy elevados para poder utilizarlas en un sistema en tiempo real. No obstante, posteriormente los resultados fueron bastante positivos, realizando una serie de optimizaciones y enfocando los algoritmos a lo que se quería buscar se

consiguieron unos tiempos de búsqueda bajísimos, permitiendo que el sistema no se ralentizase.

Las optimizaciones que se introdujeron al código fueron desde la forma en que se recorrían las matrices, pasando por la reducción de datos a tratar y finalizando por el uso de estructuras de datos más pequeñas o el uso de clases especiales para parámetros, con el objetivo de evitar operaciones de lectura y escritura fuera de memoria local.

Así, una de las conclusiones que se extrae de este trabajo es que en aquellos sistemas que utilicen reconocimiento visual las estrategias a seguir deben ser lo más específicas posibles. Por desgracia, a día de hoy aún no existe ningún algoritmo ni técnica general de reconocimiento visual que iguale los tiempos de respuesta del ojo humano, por lo que para que un sistema emulase esas características debería tener, como poco, unas características mucho mayores de las que actualmente existen, sin contar que no existe ningún algoritmo de simulación de la vista humana. No obstante, en la mayoría de casos no es necesario un rango tan elevado de libertad, sino que las aplicaciones conocen de antemano que deben buscar, habiendo diseñado las estrategias de la forma más óptima posible para que las búsquedas se lleven a cabo.

La opinión del autor es que la tarea más complicada a la hora de implantar un sistema de realidad aumentada es la de identificación de escenarios mediante técnicas de visión artificial, sea con o sin marcadores. La ciencia no está precisamente muy avanzada en este campo y aún existen muchos problemas a los que enfrentarse al diseñar este tipo de soluciones. Esta opinión se correspondería, al menos, con lo sucedido en el proyecto. La fase de identificación de escenarios ha llevado más de siete meses de investigación y pruebas hasta haber dado con un modelo lo suficiente robusto.

Por su parte, tras el análisis realizado de los diferentes sistemas de realidad aumentada ya implantados, posiblemente aquellos que tienen más éxito entre los usuarios son los sistemas distribuidos que utilizan técnicas de geo-posicionamiento. La razón del éxito de este tipo de sistemas se debe a que son utilizables en casi cualquier teléfono de última generación y a que suelen estar orientadas a ofrecer información sobre lugares de interés, tanto turísticos como del sector hostelero, gasolineras, etc. Por la información que manejan y la naturaleza cambiante de ésta, este tipo de sistemas no pueden ser implantados en sistemas autónomos, ya que podrían quedarse obsoletos con facilidad y la cantidad de información que deberían almacenar puede ser demasiado elevada para los dispositivos móviles.

En los sistemas desarrollados en este trabajo se podía haber elegido un sistema autónomo para los prototipos, no obstante, se prefirió implantar un sistema distribuido pensando en futuras mejoras y en mejorar la experiencia de uso en el usuario.

Dentro del trabajo futuro cabe destacar el aumentar el número de marcadores identificados por la aplicación.

Por otro lado, en las próximas versiones se pretende realizar mejoras en las técnicas de reconocimiento y tratamiento de las rotaciones y escalas en cuanto al tiempo de respuesta que actualmente tienen.

También se debería seguir investigando el desarrollo de sistemas de realidad aumentada independizándolo lo más posible de la plataforma. En este sentido cabría analizar el uso de arquitecturas dirigidas por modelos.

Por último, cabe comentar que la construcción de los dos prototipos, así como los estudios y análisis llevados a cabo en el desarrollo de este proyecto han sido bastante satisfactorios para los interesados. Si bien aún podrían mejorarse ciertos aspectos de los sistemas, y se espera seguir trabajando en ello, el resultado ha sido, desde la valoración de los interesados bastante bueno.

REFERENCIAS

- [SUT 68] Sutherland, I. 1968. "A head-mounted three-dimensional display. In Proceeding of the Fall Joint Computer Conference". AFIPS Conference Proceedings, vol. 33. AFIPS, Arlington, VA., 757- 764.
- [VRES 10] <http://www.virtualresearch.com/>
- [DOD 08] Department of Defense, United States of America, GLOBAL POSITIONING SYSTEM STANDARD POSITIONING SERVICE PERFORMANCE STANDARD, 2008.
- [GPS 10] <http://www.gps.gov/>
- [SOM 02] J. A. Somolinos Sánchez, "Avances en robótica y visión por computador", 2002.
- [CAW 08] S. Cawood, M. Fiala, "Augmented Reality: A practical guide", 2008.
- [BIM 05] O. Bimber, R. Rakar, "Spatial Augmented Reality. Merging Real and Virtual Worlds", 2005.
- [LOP 10]
- [CHO 09] O. Choudary, V. Charvillat, R. Grigoras, P. Gurdjos, "MARCH: Mobile Augmented Reality for Cultural Heritage", 2009.
- [TAT 07] K. Tateno, I. Kitahara, Y. Ohta, "A Nested Marker for Augmented Reality", 2007.
- [RGB 98] Multimedia systems and equipment - Colour measurement and management - Part 2-1: Colour management - Default RGB colour space – sRGB, 1998.
- [SHR 05] D. Shreiner, M. Woo, J. Neider, T. Davis, "OpenGL Programming Guide: The Official Guide to Learning OpenGL", 2005.
- [SHR 00] D. Shreiner, "OpenGL Reference Manual: The Official Reference Document to OpenGL", 2000.
- [GL ES 10] <http://www.khronos.org/opengles/>
- [JIA 09] H. Jiang, S. X. Yu, "Linear Solution to Scale and Rotation Invariant Object Matching ", 2009.
- [ART 10] <http://www.hitl.washington.edu/artoolkit/>
- [ART P 10] http://www.studierstube.icg.tu-graz.ac.at/handheld_ar/artoolkitplus.php/
- [JART 10] <http://www.c-lab.de/jartoolkit/>
- [OKA 96] S. Okabayashi, "Visual Optics of Head-Up Displays (HUDs) in Automotive Applications", 1996.
- [JAV 02] B. Javidi, "Image Recognition and Classification. Algorithms, Systems and Applications", 2002.
- [PLA 06] C. Platero, "Apuntes de visión artificial", 2006.
- [ESC 01] A. Escalera, "Visión por Computador: Fundamentos y Métodos", 2001.
- [PAJ 01] G. Pajares, "Visión por computador", 2001.
- [BRA 01] T. Bräunl, S. Feyrer, W. Rapf, M. Reinhardt, "Parallel Image Processing", 2001.
- [AML 08] K. Amlacher, L. Paletta, "Geo-Indexed Object Recognition for Mobile Vision Tasks", 2008.
- [DEY 00] A. K . Dey, G. D. Abowd, "Towards a Better Understanding of Context and Context-Awareness", 2000.
- [WIKTD 10] <http://www.wikitude.org/>
- [LAY 10] <http://www.layar.com/>
- [OLW 05] A. Olwal, C. Lindfors, J. Gustafsson, T. Kjellberg, L. Mattsson, "ASTOR: An Autostereoscopic Optical See-through Augmented Reality System", 2005.
- [XU 07] Xu, "GPS. Theory, Algorithms and Applications", 2007.
- [SYMB 10] <http://www.symbian.org/>
- [MCPH 06] F. McPherson, "How to Do Everything with Windows Mobile", 2006.
- [IPH DEV 10] <http://developer.apple.com/iphone/index.action>
- [SING 06] A. Singh, "Mac OS X Internals. A Systems Approach", 2006.
- [AND SDK 10] <http://developer.android.com/sdk/index.html>

[J3D 10] <http://java.sun.com/javase/technologies/desktop/java3d/>

[ULT 10] <http://www.ist-ultra.org/>

[NOK CV 10] <http://research.nokia.com/research/projects/nokiacy/>

[JRMI 10] <http://java.sun.com/docs/books/tutorial/rmi/index.html>

[OW 05] A. Owl, T. Höllerer, "POLAR: Portable, Optical see-through, Low-cost Augmented Reality", 2005.

[INVZ 10] <http://www.invizimals.com/>

[GOGG 10] <http://www.google.com/mobile/goggles>

[G SANT 10] <http://www.fdi.ucm.es/migs/presentacion.html>

[AND TTS 10] <http://developer.android.com/resources/articles/tts.html>

[HIL 95] J. R. Hilera, V. J. Martínez, "Redes neuronales artificiales. Fundamentos, modelos y aplicaciones", 1995.

[PAT 96] D. W. Patterson, "Artificial neural Networks. Theory and Applications", 1996.

[HAY 94] S. Haykin, "Neural Networks – A Comprehensive Foundation", 1994.

[MAT 10] H. Demuth, M. Beale, M. Hagan, "Neural Network Toolbox 6. User's guide", 2010.

[LEV 44] K. Levenberg, "A Method for the Solution of Certain Non-Linear Problems in Least Squares", 1944.

[ZAK 06] N. C. Zakas, J. McPeak, J. Fawcett, "Professional AJAX", 2006.

[JSON 10] <http://www.json.org/>

[XPATH 10] <http://www.w3.org/TR/xpath/>

[IPV 10] <http://www.upv.es/>

[SEU 09] S. J. Kim, A. K. Dey, "Simulated Augmented Reality Windshield Display as a Cognitive Mapping Aid for Elder Driver Navigation", 2009.

[AVE 08] S. J. Kim, A. K. Dey, "Simulated Augmented Reality Windshield Display as a Cognitive Mapping Aid for Elder Driver Navigation", 2009.

[YUV 95] RECOMMENDATION ITU-R BT.601-5, 1982-1995.

[YUV 02] RECOMMENDATION ITU-R BT.709-5, 1990-2002.

An Analysis of Augmented Reality Systems

Héctor López
Telefónica I+D
Madrid, Spain
hlp@tid.es

Antonio Navarro
Univ. Complutense de Madrid
Madrid, Spain
anavarro@sip.ucm.es

José Relaño
Telefónica I+D
Madrid, Spain
joserg@tid.es

Abstract—Augmented Reality Systems (ARSs) are a logical evolution of virtual reality systems. This new type of system enriches reality by mixing real elements with electronic information. This paper identifies the common tasks performed by ARSs, the underlying software and hardware components that support them and the most common architectures for ARSs. Subsequently, the paper reviews some ARSs, identifying their main hardware and software components. Thus, the goal of this paper is to serve as a guide for those entities that are planning to deploy an augmented reality system.

Keyword: *scene capture; scene identification; scene processing; scene visualization; architecture*

I. INTRODUCTION

Virtual Reality Systems (VRs) were developed to simulate reality in computer systems [1]. Thus, users of virtual reality systems leave reality to browse an alternative reality generated by a computer system.

Augmented Reality Systems (ARSs) blend together elements of reality with electronic elements generated by a computer system [2, 3]. Users can thus see real elements enriched with electronic information.

Applications of ARSs are almost endless: in education, students can see any real educational object [4] (e.g., monuments, pictures, engines, etc.) with the information that teachers decide to show them; in retail business, customers can visit a shopping centre and obtain information about the type of products or discounts to be found in every shop without visiting the shops; in medicine [5], surgeons can operate on patients with their vital signs in sight, without taking a look on monitors.

However, the use of ARSs is not widespread. Only a few projects and companies are currently using this type of system. This paper analyzes the main tasks that ARSs carry out, the most common architectures, and the main software and hardware components that ARSs use. In addition, several ARSs are analyzed and their software and hardware components are reviewed (as far as these components have been identified).

Thus, this paper aims to be a guide for those entities that are planning to deploy these systems. Section II, identifies the tasks that ARSs carry out. Section III identifies the software and hardware components responsible for supporting these tasks. Section IV analyzes the most common ARS architectures. Section V analyzes some

projects and specific implementations of components identified in Section III. Finally, our conclusions are presented.

II. TASKS IN AUGMENTED REALITY

In general ARSs perform four tasks: (i) scene capture; (ii) scene identification for selecting the right information for augmenting it; (iii) scene processing, enriching scenes captured with augmented information; and (iv) visualization of the augmented scene. The following subsections describe the main techniques involved in these tasks. Figure 1 depicts this process.

A. Scene capture and augmented scene visualization

Most ARSs capture images of reality in order to augment it. This can be done by using cameras or videos. However in both cases, a single photograph is processed.

In general, scene capture devices are physical components that perceive the reality that has to be augmented. These devices can be grouped into two major sets:

- *Video-through devices.* These devices capture reality with irrespective of the device responsible for visualization of the augmented reality (e.g., video cameras and mobile phones with camera).
- *See-through devices.* These devices capture reality and depict it with the augmented information (e.g., head-mounted displays [6]).

However, some ARSs do not capture any image. Instead, the real scene is static and enriched with digital information using static (i.e., non portable) devices.

In Section V both types of system are analyzed.

B. Scene identification techniques

Scene identification identifies scenarios and is a key process in reality augmentation. There are two main categories of scene identification mechanisms:

- *Marker-based.* Markers are visual tags included in the real scene that can be recognized by the ARS [7, 8, 9, 10, 11, 12]. Markers allow fast identification. They are usually used in closed environments [13], although they can also be used in mobile environments [14].

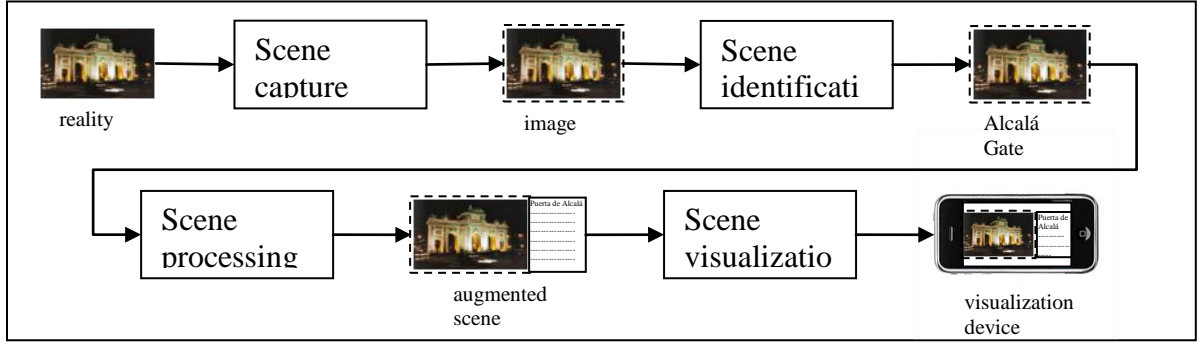


Figure 1. Tasks in augmented reality

- *Non marker-based ARSs* that do not use markers use three different mechanisms for scene identification:
 - *Image recognition*. This mechanism identifies scenes from images or video [15].
 - *Geopositioning*. This mechanism uses the position and orientation of the device capturing reality. Simpler than image recognition, it is more error prone in closed environments than an image recognition mechanism.
 - *Hybrid*. This mechanism mixes image recognition and geopositioning technologies.

C. Scene processing

When the scene is captured and identified, the process of mixing real scenes with virtual information begins. Basically, this process superimposes any type of digital information (usually audio and video) onto the real scene.

There are 2D and 3D graphical libraries that permit mixing of real and virtual information. In computer environments, open software library OpenGL [16] is one of the most notable. In mobile phone environments, graphical libraries depend on specific manufacturers, such as NokiaCV [17] for Nokia phones.

III. GENERIC HARDWARE AND SOFTWARE COMPONENTS

This section identifies generic hardware and software components necessary for deployment of ARSs (e.g., graphical library).

Table I depicts the main hardware components used in ARSs.

TABLE I. GENERIC HARDWARE COMPONENTS

Task	Hardware
<i>Video-through</i> capture	Any video camera
<i>See-through</i> capture	Embedded video camera, HMD (<i>Head Mounted Display</i>)
Geopositioning	GPS antenna, movement sensor
Hybrid	GPS antenna, movement sensor, capture image device
Image processing	CPU with, at least, 2 GHz (recommended)

Data storage	Any hard drive
Local networking	Network card, RJ45 connector, wireless antenna, access point
Mobile networking	GSM hardware
<i>Video-through</i> visualization	Screen, projector
<i>See-through</i> visualization	HMD, mobile telephone

Table II depicts the main software components used in ARSs.

TABLE II. GENERIC SOFTWARE COMPONENTS

Task	Software
Scene capture	Video camera driver
Visual recognition	Image recognition library
Geopositioning	Digital compass, GPS software
Hybrid	Digital compass, GPS software, image recognition library
Image processing	Image processing library
Data storage	Data base
Local networking	Network driver
Mobile networking	GSM driver
Visualization	Screen driver, Image processing library

Section V analyzes some ARSs and identifies the specific software and hardware components that comprise the system (as far as these components have been identified).

IV. PROCESSING ARCHITECTURES IN ARSS

There are two main architectures in ARSs: autonomous and distributed. Because response time in ARSs is a very important issue, the selection of one type of architecture has to be carefully analyzed.



Figure 2. Invizimals for PSP [18]

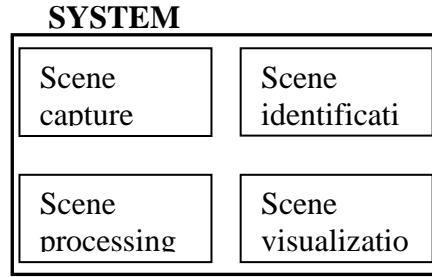


Figure 3. Architecture of autonomous systems

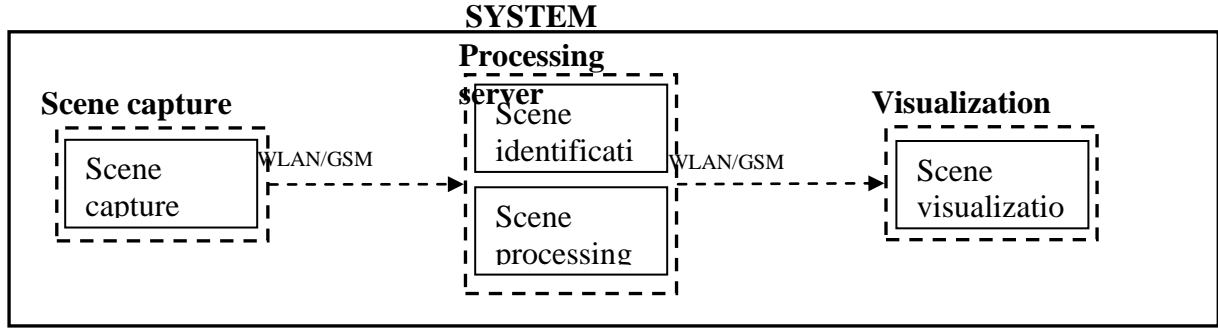


Figure 4. Architecture of distributed systems

Autonomous systems make all the operations and tasks in a single terminal (e.g., a mobile phone). Figure 2 depicts Invizimals a videogame for PSP [18]. This type of system executes simple, undemanding applications. Terminals used in autonomous systems are very heterogeneous: computers with webcams, latest generation mobile phones or visualization and processing components connected in some way can be found in autonomous systems. In general, autonomous systems are low-cost systems. Figure 3 depicts the basic architecture of autonomous systems.

Distributed systems delegate the processing in a server. This server does the majority of processing, while terminals are used as capturing and visualization devices. In some cases, terminals can do some processing before sending the image to the server (e.g., edge detection in image processing). However, most of these systems send the captured image to the server without performing any processing. If sending is performed in closed environments, W-LAN connections are usually used. However, if the terminal is used in open environments, GSM transmission and reception components are needed in both terminals and servers. Figure 4 depicts the basic architecture of distributed systems.

V. ANALYZED PROJECTS

A. ULTRA

The ULTRA project [19] aimed to develop an ultra light and compact augmented reality platform to allow programmers to develop their own systems. ULTRA was originally intended for the use of PDAs.

Because this platform was created for its use in closed industrial environments, image scene identification was made using markers. This characteristic allowed the system load to be lightened. However, terminals were responsible for most of the processing of augmented reality. Therefore PDAs with 800x600 resolution, hardware acceleration and 3D graphics were needed. OpenGL ES was the chosen graphical library. Therefore a specific implementation of EGL was used as interface between OpenGL ES and the native platform used in the PDA windows system.

In addition, ULTRA permits the creation of off-line contents from a remote server. Therefore, two additional hardware requirements arise: (i) the mobile device has to be able to perform W-LAN communications; (ii) an additional remote server is needed. Table III depicts the main components of the ULTRA project.

TABLE III. ULTRA COMPONENTS

Task	Software	Hardware
Marker recognition	OpenGL ES	Mobile telephone or PDA camera
Image processing	OpenGL ES	Mobile telephone or PDA CPU and memory
Visualization	Native device software	Mobile telephone or PDA screen

B. MARCH

The MARCH project [20] permits the visualization of digital components on rock carvings. This project was conceived to be used on mobile devices. Figure 5 depicts MARCH ARS.

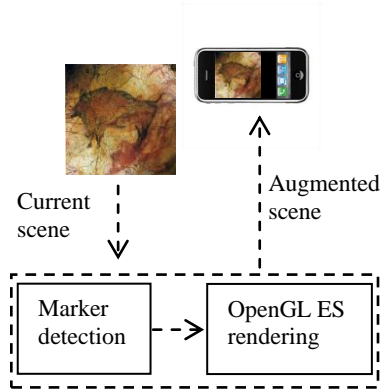


Figure 5. MARCH ARS [20]

Because the system has to be used in caves, and mobile devices have no great processing power, the project uses markers for scene identification.

The system has two modules: one off-line and the other in the mobile device. The off-line module registers the images that are augmented in the mobile device, adding the expert interpretation on the prehistoric image. Then, the interpretation is extracted on a transparent background.

Finally, because MARCH uses the graphical library NokiaCV, only mobile devices compatible with this library can be used.

TABLE IV. MARCH COMPONENTS

Task	Software	Hardware
Marker recognition	Nokia CV	Mobile telephone or PDA camera
Image processing	Nokia CV	Mobile telephone or PDA CPU and memory
Visualization	Native device software	Mobile telephone or PDA screen

C. ASTOR

ASTOR is a prototype of ARS in which digital information is depicted using projectors and *Holographic Optical Elements* (HOEs). Thus, through these transparent

screens, users do not need to use additional external visualization devices [21]. Figure 6 depicts ASTOR HOEs.

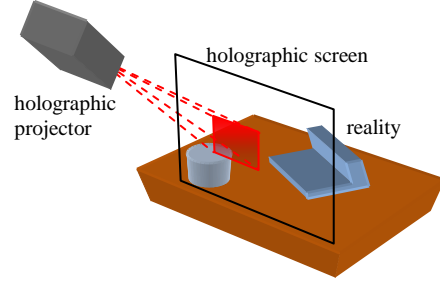


Figure 6. ASTOR ARS [21]

Therefore, in ASTOR, all the hardware and software components are localized on the server side. The graphical library used is Java 3D [22]. In addition, JAVA RMI [23] is used to deal with client-server communications. Table V depicts ASTOR components.

TABLE V. ASTOR COMPONENTS

Task	Software	Hardware
Image recognition	Java3D	Video camera
Image processing	Java3D	NVIDIA GeForce MX 440-SE, 256 MB RAM
Networking	JRMI	Client-server architecture
Visualization	Native device software	Projectors, HOE screen

Finally, ASTOR is designed to be used in small environments, although it could be used in larger environments, with a considerable increase in costs.

D. POLAR

POLAR is a low-cost ARS that superimposes augmented information using a partially silver-plated mirror [24]. By using correct positioning, this mirror reflects the augmented information on the real images. Figure 7 depicts POLAR ARS. Note that in this system a simple computer screen is able to project the augmented information onto the mirror. In addition, a specific device is needed to detect the direction of the user sight in order to position the mirror.

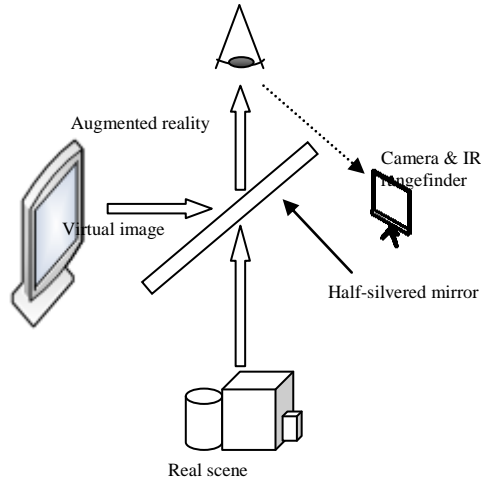


Figure 7. POLAR ARS [24]

Table VI depicts POLAR components.

TABLE VI. POLAR COMPONENTS

Task	Software	Hardware
Image recognition	OpenGL ES	WebCam, movement sensor
Image processing	OpenGL ES	PC CPU and memory
Visualization	Native device software	PC screen, half reflecting mirror

E. Layar

Layar is an Android [25] application that permits the visualization of digital information on real images captured using mobile phones [26].

The phone shows the real scene to the user using its

camera and screen. Using a GPS and a compass, the phone sends the user's location and direction to the server. Then, the server responds with a layer of digital information that is depicted on the real image that the camera is capturing. Figure 8 depicts Layar architecture.

Table VII depicts Layar components.

TABLE VII. LAYAR COMPONENTS

Task	Software	Hardware
Geopositioning	Digital compass, GPS software	Mobile telephone or PDA camera, GPS antenna
Knowledge storage	Data base software	Data base server
Visualization	Native device software	Mobile telephone or PDA screen

F. LOCUS

LOCUS is an ARS designed for mobile terminals, especially PDAs and 3G phones [27]. With LOCUS, users can see 3D augmented information about certain places using geopositioning techniques.

Table VIII depicts LOCUS components.

TABLE VIII. LOCUS COMPONENTS

Task	Software	Hardware
Geopositioning	Digital compass, GPS software	Mobile telephone or PDA camera, GPS antenna
Visualization	Native device software	Mobile telephone or PDA screen

VI. CONCLUSIONS

After analyzing different ARSs several conclusions can be drawn.

Most of the analyzed ARSs are distributed systems. Therefore, telecommunications in ARSs are a key issue.

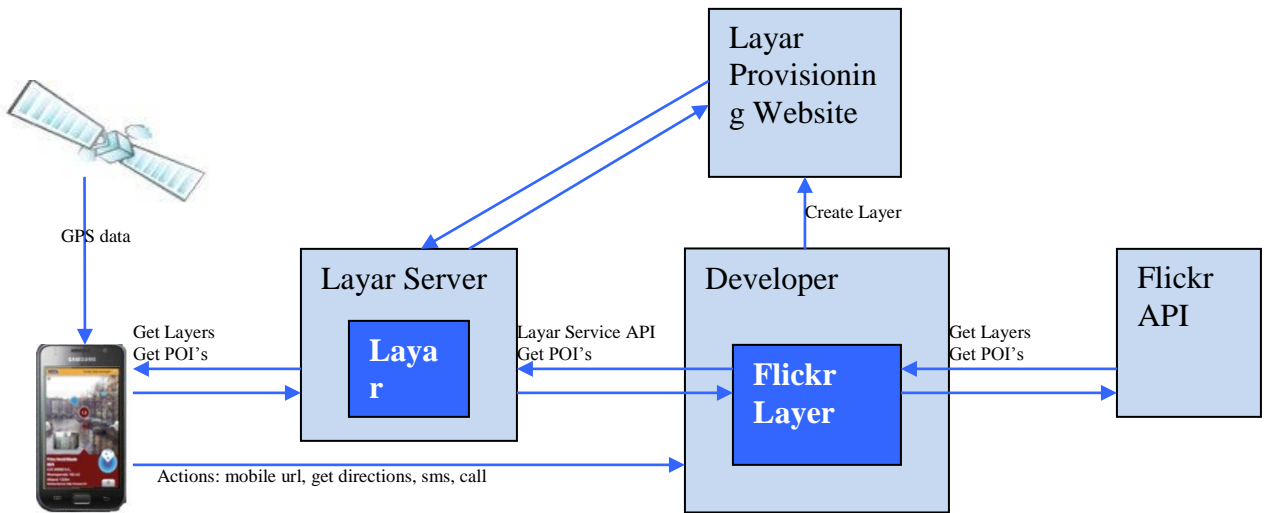


Figure 8. Layar architecture [26]

See-through devices are the most common scene capture and visualization devices. In particular, mobile phones and PDAs are the most common, in contrast to head-mounted devices.

There is not a predominant mechanism for scene identification. However marker-based identification is more common for indoor systems, while geopositioning identification is more common for outdoor systems.

In image processing, again, there is not a predominant library: Java 3D, NokiaCV and OpenGL ES are used in the analyzed systems. In addition, in general, the same libraries used for scene processing are used for scene identification.

Regarding VRSs, ARSs are both indoor or outdoor systems, while, in general, VRSs are indoor systems. This makes ARSs more versatile than VRSs.

Finally, taking into account software and hardware components used in ARSs, the deployment cost of these systems can be well controlled. The cost ultimately depends on the capabilities of the ARSs.

Future work includes analyzing more ARSs and the deployment of an ARS for a museum. The system will use a mobile phone for scene identification and visualization, scene identification through image recognition and image processing based on OpenGL ES.

ACKNOWLEDGMENT

El Ministerio de Educación y Ciencia (TIN2009-14317-C03-01), La Comunidad Autónoma de Madrid (S2009/TIC-1650) and La Universidad Complutense de Madrid (Group 921340) have supported this work.

REFERENCES

- [1] G. Kim, *Designing Virtual Reality Systems: The Structured Approach*. Springer, 2005.
- [2] S. Cawood and M. Fiala, *Augmented Reality: A practical guide*. Pragmatic Bookshelf, 2008.
- [3] O. Bimber and R. Rakar, *Spatial Augmented Reality. Merging Real and Virtual Worlds*. A. K. Peters, Ltd. 2005.
- [4] T. Miyashita, et al., "An Augmented Reality Museum Guide", Proc. 7th IEEE/ACM International Symposium on Mixed and Augmented Reality, 2008.
- [5] M. Harders, G. Bianchi, and B. Knoerlein, "Multimodal Augmented Reality in Medicine", *Universal Access in Human Computer Interaction. Ambient Interaction*, pp. 652-658, 2007.
- [6] J. E. Swan II, A. Jones, E. Kolstad, M. A. Livingston, and H. S. Smallman, "Egocentric Depth Judgments in Optical, See-Through Augmented Reality", *IEEE Transactions on Visualization and Computer Graphics*, v.13 n.3, May 2007, p.429-442.
- [7] H. Kato and M. Billinghurst, "Marker Tracking and HMD Calibration for a video-based Augmented Reality Conferencing System", Proc. 2nd International Workshop on Augmented Reality (IWAR 99), October 1999, San Francisco, USA,
- [8] J. Rekimoto and Y. Ayatsuka, "CyberCode: Designing augmented reality environments with visual tags", Proc. *Designing Augmented Reality Environments (DARE) 2000*, Apr. 2000.
- [9] X. Zhang and N. Navab, "Tracking and pose estimation for computer assisted localization in industrial environments", Proc. *IEEE Workshop on Application of Computer Vision 2000*, 2000, pp. 214-221.
- [10] S. Malik, G. Roth, and C. McDonald, "Robust 2D Tracking for Real-Time Augmented Reality", Proc. *Vision Interface (VI) 2002*, Calgary, Alberta, Canada, 2002. pp. 399-406.
- [11] M. Fiala, "ARTag Revision 1, A Fiducial Marker System Using Digital Techniques". NRC/ERB-1117. November 24, 2004.
- [12] K. Tateno, I. Kitahara, and Y. Ohta, "A nested marker for augmented reality", Proc. *ACM SIGGRAPH 2006 Sketches*, July 30-August 03, 2006.
- [13] R. Wojciechowski, K. Walczak, M. White, and W. Cellary, "Building Virtual and Augmented Reality museum exhibitions", Proc. *9th International conference on 3D Web technology*, April 05-08, 2004.
- [14] B. Avery, B.H. Thomas, and W. Piekarski, "User evaluation of see-through vision for mobile outdoor augmented reality". Proc. *7th IEEE/ACM International Symposium on Mixed and Augmented Reality*, September 15 - 18, 2008, pp. 69-72.
- [15] G. Zhu, M. Yang, K. Yu, W. Xu, and Y. Gong, "Detecting video events based on action recognition in complex scenes using spatio-temporal descriptor". Proc. *17th ACM International Conference on Multimedia*, Beijing, China, October 19 - 24, 2009.
- [16] OpenGL website (last access date June 16, 2010) <http://www.opengl.org/>
- [17] NokiaCV website (last access date June 16, 2010) <http://research.nokia.com/research/projects/nokiavc/>
- [18] Invizimals website (last access date June 16, 2010), <http://www.invizimals.com/>
- [19] Ultra website (last access date June 16, 2010), <http://www.ist-ultra.org/>
- [20] O. Choudary, V. Charvillat, R. Grigoras, and P. Gurdjos, "MARCH: mobile augmented reality for cultural heritage". Proc. *17th ACM International Conference on Multimedia*, Beijing, China, October 19 - 24, 2009.
- [21] A. Olwal, C. Lindfors, J. Gustafsson, T. Kjellberg, and L. Mattsson, "ASTOR: An Autostereoscopic Optical See-through Augmented Reality System", Proc. *4th IEEE/ACM International Symposium on Mixed and Augmented Reality*, October 05-08, 2005, pp.24-27.
- [22] Java 3D API (last access date June 16, 2010), <http://java.sun.com/javase/technologies/desktop/java3d/>
- [23] Java RMI (last access date June 16, 2010), <http://java.sun.com/docs/books/tutorial/rmi/index.html>
- [24] A. Olwal, and T. Höllerer, "POLAR: portable, optical see-through, low-cost augmented reality", Proc. *ACM Symposium on Virtual Reality Software and Technology*, Monterey, CA, USA, November 07 - 09, 2005.
- [25] Android website (last access date June 16, 2010), <http://www.android.com>
- [26] Layar website (last access date June 16, 2010), <http://www.layar.com/>
- [27] LOCUS website (last access date June 16, 2010), <http://www.locus.org.uk>

