

# **APLICACIÓN MÓVIL DE REALIDAD AUMENTADA PARA MEJORAR LA CONCIENCIA SITUACIONAL EN EL ÁMBITO MILITAR**

*Mag. Alejandro Daniel Mitaritonna*

**Directora:** Dra. María José Abásolo

**Trabajo Final presentado para obtener el grado de Especialista en  
Computación Gráfica, Imágenes y Visión por Computador**



**Facultad de Informática - Universidad Nacional de La Plata**

Febrero, 2019

# Resumen

Diversos proyectos militares han centrado sus esfuerzos en diseñar un sistema digital integrado como soporte para la toma de decisiones del personal militar, para afrontar el reto de poseer un Conciencia Situacional (CS) degradada. En este trabajo se propone un framework de Realidad Aumentada (RA) denominado RAIOM (Realidad Aumentada para la Identificación de Objetivos Militares) –entendiéndose por identificación de objetivos militares a todos los objetivos militares del enemigo circundantes a los soldados que representan una amenaza bélica- cuyo objetivo es mejorar la CS de los soldados en el campo de batalla mediante el uso de la RA. Para el desarrollo del framework se consideró como principal característica a la CS, y a la RA como tecnología de visualización e interpretación contextual. La aplicación de metodologías tales como SCRUM y Diseño Centrado en el Usuario (DCU) permitió desarrollar los despliegues correspondientes. En particular, este trabajo presenta una recopilación actualizada de diferentes sistemas utilizados en el ámbito militar y se analizan las características principales que debe poseer un framework de software basado en RA para uso militar. Se diseña un framework basado en RA que permite la visualización de objetivos tácticos tales como visualización de objetivos tácticos en un radar en 360 grados, posicionamiento y orientación espacial de los objetos y del operador, ayudas visuales mediante iconografía/simbología adaptable, uso de menús interactivos mediante reconocimiento gestual y vocal, reconocimientos de objetos edilicios y uso de cartografía en 3D mediante uso de RA. Se utiliza una arquitectura basada en capas para facilitar la integración tecnológica como también una arquitectura de sistemas distribuida donde la principal característica se centra en realizar el procesamiento de imágenes en un componente de hardware externo. Se realizan los despliegues utilizando el framework de RA propuesto en el presente trabajo y, finalmente, se arriba a las conclusiones finales y a las futuras líneas de investigación.

Palabras claves: *realidad aumentada, conciencia situacional, framework, diseño centrado en el usuario, SCRUM+DCU*

# Índice

<b>Lista de Figuras</b> .....	<b>6</b>
<b>Lista de Tablas</b> .....	<b>9</b>
<b>Tabla de Abreviaturas</b> .....	<b>10</b>
<b>I. Estado de Situación</b> .....	<b>13</b>
<b>Capítulo 1</b> .....	<b>14</b>
<b>Introducción</b> .....	<b>14</b>
<b>i. Motivación</b> .....	<b>14</b>
<b>ii. Objetivos</b> .....	<b>17</b>
<b>iii. Aportes realizados</b> .....	<b>17</b>
<b>iv. Organización del informe</b> .....	<b>18</b>
<b>Capítulo 2</b> .....	<b>20</b>
<b>Fundamentos de la Realidad Aumentada</b> .....	<b>20</b>
2.1 ¿QUÉ ES LA RA? .....	20
2.2 PARTES DE UNA APLICACIÓN DE RA.....	26
2.3 MÉTODOS DE ENTRADA DE DATOS EN UN SISTEMA DE RA .....	30
2.4 VISUALIZACIÓN.....	33
2.5 TECNOLOGÍAS PARA VISUALIZAR LA RA.....	38
2.6 TRACKING.....	40
2.6.1 TRACKING MEDIANTE DISPOSITIVOS FÍSICOS ESPECÍFICOS.....	43
2.6.2 TRACKING BASADO EN VISIÓN .....	45
2.6.3 TRACKING HÍBRIDO .....	50
2.7 RECONOCIMIENTO DE OBJETOS .....	51
2.8 ARQUITECTURA DE SOFTWARE PARA APLICACIONES DE RA .....	56
<b>Capítulo 3</b> .....	<b>59</b>
<b>Realidad Aumentada en proyectos militares</b> .....	<b>59</b>

3.1	REALIDAD AUMENTADA PARA MEJORAR LA CONCIENCIA SITUACIONAL .....	59
3.2	PRINCIPALES PROYECTOS MILITARES QUE APLICAN LA RA.....	64
3.2.1	PROYECTO EYEKON .....	65
3.2.2	PROYECTO BARS .....	66
3.2.3	PROYECTO IARM .....	68
3.2.4	PROYECTO ULTRA-Vis .....	69
3.3	COMPARATIVA DE PROYECTOS MILITARES QUE UTILIZAN RA.....	73
<b>Capítulo 4.....</b>		<b>77</b>
<b>Arquitectura y diseño de Software .....</b>		<b>77</b>
4.1	FRAMEWORK Y SU IMPORTANCIA .....	77
4.2	COMPARATIVA DE FRAMEWORKS DE RA MÁS UTILIZADOS.....	81
4.3	ARQUITECTURA DE SOFTWARE POR CAPAS Y COMPONENTES.....	84
4.4	AMBIENTE DE DESARROLLO.....	85
<b>II. Framework Propuesto .....</b>		<b>89</b>
<b>Capítulo 5.....</b>		<b>90</b>
<b>Diseño de las Arquitecturas .....</b>		<b>90</b>
5.1	INTRODUCCIÓN.....	90
5.2	METODOLOGÍA DE DESARROLLO UTILIZADA.....	90
5.3	ARQUITECTURA DEL FRAMEWORK. MIDDLEWARE .....	93
5.3.1	CAPA: MÓDULOS .....	93
5.3.2	CAPA: LIBRERÍAS.....	107
5.4	ARQUITECTURA DEL SISTEMA. COMPONENTES DE HARDWARE UTILIZADOS.....	108
5.5	PROCESO DISTRIBUIDO .....	113
<b>Capítulo 6.....</b>		<b>116</b>
<b>Despliegue de la solución .....</b>		<b>116</b>
6.1	INTRODUCCIÓN.....	116
6.2	DISPOSITIVOS UTILIZADOS .....	116
6.3	DESPLIEGUES CONSIDERADOS .....	118
6.4	MÓDULO <i>VISION</i> .....	121
6.5	INTEGRACIÓN DE TECNOLOGÍAS .....	122
<b>Capítulo 7.....</b>		<b>126</b>

<b>Conclusiones y Líneas de trabajo futuras .....</b>	<b>126</b>
7.1 CONCLUSIÓN DEL TRABAJO .....	126
7.2 LÍNEAS FUTURAS DE TRABAJO .....	128
<b>Bibliografía.....</b>	<b>130</b>

# Lista de Figuras

Figura 2.1– La RA abarca un amplio espectro de sistemas. Dentro del dominio MAR .....	21
Figura 2.2– Continuum de la virtualidad.....	22
Figura 2.3 – Diagrama de las partes de una aplicación de RA.....	27
Figura 2.4 – Modelo de referencia. Componentes de software requeridos en un sistema de RA ubicua.....	29
Figura 2.5– Arquitectura de referencia genérica de sistemas de RA.....	29
Figura 2.6 – Sistema de reconocimiento gestual .....	32
Figura 2.7 – Sistemas de Coordenadas de un sistema de RA.....	35
Figura 2.8 – Diagrama conceptual de la tecnología <i>optical see-through</i> .....	39
Figura 2.9– Diagrama conceptual de la tecnología <i>video see-through</i> .....	40
Figura 2.10 – Diferentes tipos de marcadores: a) Template; b) ID-marker; c) DataMatrix y d) QR .....	46
Figura 2.11– Marco de referencia usado para estimación de la pose .....	50
Figura 2.12 – Diferentes niveles de procesamiento de las tareas de RA. a) las tareas se ejecutan en el cliente; b) en el server se ejecuta la tarea de tracking; c) en el servidor se ejecutan las tareas de tracking y la lógica de la aplicación y d) todas las tareas se realizan en el servidor.....	58
Figura 3.1– Modelo de la CS en la toma de decisión dinámica .....	60
Figura 3.2 – Modelo de los tres niveles de la CS .....	62
Figura 3.3- Diseño conceptual del proyecto iARM.....	69
Figura 3.4 – Componentes del prototipo ULTRA-Vis.....	70
Figura 3.5 – Interfaz de usuario del prototipo ULTRA-Vis.....	71
Figura 3.6 – Información provista y componentes de hardware del proyecto ACR4 (fuente: ARA) .....	72
Figura 4.1 – Arquitectura de Android .....	86
Figura 4.2 – Secuencia de ejecución de una aplicación Android .....	88
Figura 5.1 – SCRUM + DCU .....	92
Figura 5.2 – Diseño basado en componentes de la arquitectura del framework.....	95

Figura 5.3 – Filtrado de información. Cierta conjunto de información es prioritaria. Se excluye el resto de información.....	96
Figura 5.4 – Simbología del radar. Estructura de la visualización del radar.....	97
Figura 5.5 – Mapa 360 grados del Prototipo desarrollado utilizando el framework (fuente: propia).....	98
Figura 5.6 – Simbología gestual.....	99
Figura 5.7 – Diseño de la estructura de paquetes / módulos del framework.....	100
Figura 5.8 – Pruebas iniciales de reconocimiento gestual y visualización de iconos (fuente: propia).....	102
Figura 5.9 – Representación del modelo Bag Of Words utilizado para clasificación. (Fuente: Jun Yang et al [62]).....	104
Figura 5.10 – Explicación gráfica de Bag Of Words: (i) imágenes representando una palabra visual (ii) características (iii) descriptores ubicados en el espacio (iv) agrupación en clusters (v) imagen a clasificar (vi) características (vii) correspondencias con los descriptores conocidos (viii) histograma de correspondencias.....	104
Figura 5.11 –Prueba de identificación del edificio de Electrónica del CITEDEF utilizando FAST + FREAK.....	106
Figura 5.12 – Arquitectura del Sistema de RA propuesto.....	109
Figura 5.13 – Diseño de las capacidades del framework propuesto.....	110
Figura 5.14 – Ejemplo de la Arquitectura propuesta utilizando gafas de RA (1) y ODROID-XU (2).....	111
Figura 5.15 – ODROID XU3. Solución Heterogeneous Multi-Processing (HMP) solution.....	112
Figura 5.16 – ODROID XU3.....	113
Figura 5.17 – Diagrama de despliegue.....	114
Figura 5.18 – Extrayendo características de una imagen mediante procesamiento distribuido utilizando un ODROID-XU3 y una PC que simula a un dispositivo móvil (fuente: propia).....	115
Figura 6.1 – Dispositivos móviles utilizados para en los despliegues.....	118
Figura 6.2 – <i>Glass cardbox</i> acoplado a los smartphone utilizado en los despliegues.....	119

Figura 6.3 – Opciones de arquitecturas de sistemas propuestos.....	120
Figura 6.4 – Prueba de extracción de características utilizando SIFT. Reconocimiento y Clasificación de objetos edilicios .....	121
Figura 6.5 – Visualización de iconografía y radar de 360 grados con información contextual impresa.....	122
Figura 6.6 – Activación de un mapa por medio de reconocimiento gestual .....	122
Figura 6.7 – QRCode para acceder a vídeo en el cual se muestra la integración de las funcionalidades principales del framework.....	124



# Lista de Tablas

Tabla 3.1 Elementos imprescindibles para mantener un alto nivel de CS .....	74
Tabla 3.2 Tecnologías utilizadas como soporte a la CS .....	75
Tabla 4.1 Parámetros funcionales tecnológicos .....	82
Tabla 4.2 Parámetros no funcionales.....	83
Tabla 6.1 Dispositivos móviles. Características técnicas.....	117
Tabla 6.2 Despliegues utilizados para validar la solución propuesta.....	118

# Tabla de Abreviaturas

AAM	Active Appearance Model
API	Application Programming Interface
ARA	Applied Research Associates
ARM	Advanced RISC Machines
ASCII	American Standard Code for Information Interchange
BARS	Battlefield Augmented Reality System
BoW	Bag Of Words
BRIEF	Binary Robust Independent Elementary Features
BRISK	Binary Robust Invariant Scalable Keypoints
C4I2	Comando, Control, Comunicaciones, Computación, Información, Inteligencia Command, Control, Communications, Computers, Intelligence, Surveillance and Reconnaissance
C4ISR	sance
CAR	Contact-lens augmented reality
CC	Comando y Control
CECOM	Communications Electronics Command)
CII	Centro Integrador
CITEDEF	Instituto de Investigaciones Científicas y Técnicas para la Defensa
COMFUT	Combatiente Futuro
COP	Common Operating Picture
COTS	Commercial Off-The-Shelf
CPU	Central Processing Unit
CS	Conciencia Situacional
CU	Caso de Uso
DARPA	Defense Advanced Research Projects Agency
DCU	Diseño Centrado en el Usuario
DoG	Difference of Gaussians
DoH	Determinant of Hessian
DSR	Design Science Research
DWARF	Distributed Wearable Augmented Reality
FAST	Features from Accelerated Segment Test
FELIN	Fantassin à Équipements et Liaisons Intégrés
FF.AA.	Fuerzas Armadas Argentinas
FIST	Future Integrated Soldier Technology
FREAK	Fast Retina Keypoint
GIS	Geographic Information System
GLOH	Gradient Location and Orientation Histogram
GPGPU	General-Purpose Computing on Graphics Processing Units
GPS	Global Positioning System
GPU	Graphics Processing Units

HCI	Human-Computer Interaction
HMAR	Handheld Mobile Augmented Reality
HMD	Head-Mounted Display
HMP	Heterogeneous Multi-Processing
HSV	Hue, Saturation, Value
HUD	Head-Up Displays
iARM	Intelligent Augmented Reality Model
IdZ	Infanterist der Zukunft
IHAS	Integrated Helmet Assembly Subsystem
IMU	Inertial Measurement Unit
IoC	Inversion of Control
JCa	Puesto de Mando
JIT	Just In Time
JNI	Java Native Interface
JPat	Jefe de Patrulla
LoG	Laplacian of Gaussian
LW	Land Warrior
MAR	Mobile Augmented Reality
MSER	Maximally Stable Extremal Regions
MVC	Model-View-Controller
MVD	Máquina Virtual Dalvik
MWS	Modular Weapon System
NDK	Native Development Kit
NRL	Naval Research Laboratory
OHA	Open Handset Alliance
ORB	Oriented FAST and Rotated BRIEF
Pat	Patrulla
PCA	Principal Component Analysis
PCA	Principal Component Analysis
PCBR	Principal Curvature-Based Region detector
POO	Programación Orientada a Objetos
PTAM	Parallel Tracking And Mapping
QBR	Químico, Biológico y Radioactivo
QR	Quick Response code
RA	Realidad Aumentada
RAIOM	Realidad Aumentada para la Identificación de Objetivos Militares
RFID	Radio Frequency Identification
RGP	Recognized Ground Picture
RPIDFA	Régimen del Personal de Investigación y Desarrollo de las FF.AA.
RTP	Real-time Transport Protocol
RV	Realidad Virtual
SAGAT	Situation Awareness Global Assessment Technique

SART	Situation Awareness Rating Technique
SPAM	Situation Present Awareness Method
SDK	Software Development Kit
SfM	Structure from Motion
SIFT	Scale-invariant feature transform
SIMD	Single Instruction Multiple Data
SLAM	Simultaneous Localization And Mapping)
SLAM	Simultaneous Localization And Mapping
SMS	Short Message Service
SURF	Speeded Up Robust Features
UAV	Unmanned Aerial Vehicle
ULTRA-	
Vis	Urban Leader Tactical Response, Awareness and Visualization
UML	Unified Modeling Language
UNLP	Universidad Nacional de La Plata
URL	Uniform Resource Locator
XMPP	Extensible Messaging and Presence Protocol

# **I. Estado de Situación**

# Capítulo 1

## Introducción

En este capítulo se da inicio a la primera parte del informe que se centra en detallar la motivación, objetivos, hipótesis y la metodología de investigación adoptada para organizar el trabajo, los aportes realizados tanto a la comunidad militar como civil, las pruebas efectuadas y por último la organización del informe, las partes y las secciones que conforman al presente trabajo.

### i. Motivación

Como se menciona en [1], la Conciencia Situacional (CS) es una representación mental y comprensión de los objetos, eventos, gente, estados de los sistemas, interacciones, condiciones ambientales y cualquier otro tipo de factores de una situación específica que puedan afectar al desarrollo de las tareas humanas, bien sean complejas o dinámicas. Logrando mantener la CS se potencia la adquisición, la representación, la interpretación y la utilización de cualquier información relevante con el objeto de poner sentido a los eventos que ocurren, pudiéndose anticipar a los acontecimientos futuros, dando la capacidad de poder tomar decisiones inteligentes y de poder mantener el control. La CS es crucial para la mayoría de las operaciones militares, pero tiene diferentes requisitos en los diferentes niveles de las operaciones militares y las diferentes funciones.

El equipamiento del futuro soldado debe consistir en un conjunto integrado de componentes electrónicos modulares para mejorar, de manera equilibrada, sus habilidades de comunicación, observación, letalidad, movilidad y protección. La interacción del soldado con el medio ambiente también significa que debe ser capaz de procesar la información contextual sin que afecte su capacidad cognitiva. Esto dio lugar a una variedad de programas de Sistemas de "Future Soldier" lanzados por diferentes ejércitos. Estos programas tienen un alto grado de diferencia de sofisticación tecnológica y desarrollo. En general, todos estos

programas tienen como objetivo mejorar sustancialmente la eficacia de combate y supervivencia del soldado. Una regla de los programas de modernización de los soldados es que el soldado no opera solo, sino como parte de un equipo para llevar a cabo una determinada tarea en un entorno determinado. Adicionalmente en [1] se menciona que el fratricidio sigue siendo una amenaza muy real en los campos de batalla actuales. Para hacer frente a esto, los militares han puesto mucho esfuerzo en el desarrollo de tecnologías de identificación de combate para mejorar la capacidad de los soldados en identificar al enemigo con precisión. Debido a la naturaleza asimétrica de los conflictos recientes, estos sistemas utilizan cada vez más los dispositivos móviles sensibles al contexto para mejorar esta capacidad.

Como se detalla en [2], muchas de las operaciones militares se desarrollan en entornos desconocidos. Estos complejos campos de batalla son muy exigentes e introducen desafíos para el combatiente, entre ellos el hecho de tener una CS tridimensional amplia del terreno es vital para que la operación sea un éxito, minimizando los efectos colaterales. Un reto importante en el desarrollo de herramientas eficaces de apoyo de decisiones radica en la forma en que se presenta la información, siendo uno de los principales objetivos reducir la carga cognitiva en momentos de estrés para ello las soluciones tecnológicas apoyadas en mejorar la CS permiten a los soldados hacer un uso efectivo de la información variada en un contexto de batalla. La nueva tecnología ofrece métodos innovadores de obtener información contextual y representarla visualmente de una manera natural y no invasiva sin afectar el proceso cognitivo del combatiente.

Por su parte, las aplicaciones de Realidad Aumentada (RA) son cada vez más populares debido a los notables progresos en los dispositivos de computación móviles, como celulares inteligentes, asistentes digitales personales y computadoras portátiles ligeras [3]. La posibilidad de acceso a Internet en todas partes, incluso desde los dispositivos móviles motiva el desarrollo con la disponibilidad de los dispositivos electrónicos que combinan las tecnologías necesarias en una sola unidad integrada. Generalmente las aplicaciones de RA se ejecutan en un ambiente externo, y el tracking del usuario se basa en utilizar sistema de posicionamiento global (GPS), brújula electrónica y/o acelerómetros, así como también análisis del vídeo de la escena real capturada en tiempo real. Asimismo, puede requerir el acceso directo a información geo-referenciada. La aplicación también debe permitir visualizar la realidad con la superposición de objetos 3D generados. El punto principal dentro del desarrollo de la RA es

un sistema de seguimiento de movimiento o *Tracking System*. Desde el principio hasta ahora la RA se apoya en “Marcadores” o un arreglo de Marcadores dentro del campo de visión de las cámaras para que el dispositivo tenga un punto de referencia sobre el cual superponer las imágenes. Recién en los últimos años el desarrollo de RA sin marcadores (en inglés, markerless) está madurando, pues la gran capacidad computacional de los nuevos dispositivos tecnológicos permite reconocer formas geométricas, pudiendo el sistema de seguimiento captar un objeto y superponer información o una imagen [4].

De acuerdo a lo que mencionan Livingston et al [5], en lo que respecta al ámbito militar, recientemente se está utilizando la RA como soporte para mejorar la CS en la toma de decisiones en particular en operaciones militares. Muchas de las operaciones militares se desarrollan en entornos desconocidos. Estos complejos campos de batalla en 3D son muy exigentes e introducen muchos desafíos para el combatiente. Estos incluyen visibilidad limitada, falta de familiaridad con el medio ambiente, amenazas de francotirador, ocultamiento de fuerzas enemigas, mala comunicación y un problema de la localización e identificación de los enemigos y de las fuerzas aliadas. Para ello tener una CS amplia del terreno es vital para que la operación sea un éxito minimizando los efectos colaterales. De acuerdo a la investigación previa efectuada, el único centro de investigación en América que está llevando adelante proyectos de este tipo es el DARPA (Defense Advanced Research Projects Agency) pues considera a la RA como una tecnología que ayudará a realizar con éxito las operaciones militares [2].

En este contexto se evidencia la falta, en el ámbito de la Defensa, en Argentina, de especialistas que puedan aportar los conocimientos necesarios para el desarrollo de un software que implemente la RA para ayudar a mejorar el desempeño militar en operaciones militares. Por esto, la motivación principal del informe es contribuir a desarrollar un framework de software que utilice la RA a fin de desarrollar aplicaciones que se utilicen en el ámbito militar, considerando requisitos importantes en este contexto, como son la CS, las facilidades de uso y la interoperabilidad y portabilidad de los artefactos desarrollados.



## ii. Objetivos

El objetivo del informe es aportar los conocimientos necesarios sobre RA en el ámbito de la Defensa para que sean implementados en soluciones concretas como apoyo a las actividades militares. Para lograr lo antes mencionado se propone desarrollar un framework centrado en el desarrollo de aplicaciones móviles con una arquitectura distribuida e integrador de tecnologías. Por lo tanto, esto último trae aparejado los siguientes objetivos específicos:

- Explorar y analizar la bibliografía existente sobre RA y CS en el ámbito militar
- Estudiar y analizar las características funcionales de la RA para que sean aplicadas sobre dispositivos móviles utilizados por personal militar en el proceso de obtención de información circundante en un terreno desconocido y hostil.
- Diseñar e implementar un framework de software específico que tome como funcionalidades principales lo aportado por personal militar cumpliendo con las necesidades operativas propuestas.
- Desarrollar y probar artefactos desarrollados teniendo en cuenta el framework propuesto.
- Generar el conocimiento necesario en el tema para posteriormente transferir al personal militar las experiencias adquiridas.

## iii. Aportes realizados

En cuanto al aporte del presente informe se pueden listar las principales contribuciones:

- a) Recopilación de aplicaciones utilizadas en el ámbito militar para apoyar la CS y análisis de frameworks de RA disponibles, así como definición de características deseables de un software basado en RA como soporte a la CS para la toma de decisiones en el sector militar aprovechando el avance del software y de los dispositivos móviles actuales;
- b) La utilización Diseño del software antedicho utilizando una metodología combinada SCRUM-DCU para la identificación y clasificación de requisitos funcionales y para la construcción de los despliegues;

- c) Implementación del software antedicho considerando una arquitectura basada en capas a fin facilitar la integración tecnológica, el cual es un sistema distribuido utilizando como servidor componentes de hardware externo (ODROID-XU3) en donde se ejecuta el tracking, procesamiento de imágenes y reconocimiento de objetos mientras que en el cliente se ejecutan las tareas de captura de video, rendering y visualización (display). En el servidor se ejecutan las tareas de tracking y de procesamiento de imágenes mientras que en el cliente se ejecutan las tareas de captura de video, rendering y visualización (display).
- d) Pruebas de los despliegues ejecutados en dos arquitecturas de componentes de hardware, por un lado, gafas de RA + ODROID-XU3, y por el otro, el uso de glass cardboard y smartphone + ODROID-XU3.

Cabe aclarar que el trabajo se realizó en el laboratorio del LIDI de la UNLP en conjunto con el laboratorio de investigación del CITEDEF, con el apoyo de las FF.AA para la concreción del presente trabajo.

## **iv. Organización del informe**

El desarrollo del presente trabajo fue dividido en etapas con el fin de lograr el objetivo final propuesto. A continuación, se describen cada una de estas etapas que pueden verse reflejadas a lo largo de los distintos capítulos.

La parte I del informe está conformada por el Estado de Situación del trabajo de investigación en donde se encuentra el Capítulo 1 y se detallan la motivación del tema seleccionado, los objetivos, los aportes realizados, la hipótesis, y, finalmente, la organización del presente informe.

En el Capítulo 2 se investiga en profundidad los fundamentos de la RA. Dentro de los fundamentos se detallan algunas definiciones de RA, los métodos de entrada de un sistema de RA, las partes que componen a un sistema de RA, los sistemas de coordenadas utilizados en un sistema de RA, el proceso de tracking, el proceso de reconocimiento de objetos y gestual, el modelo de referencia y la arquitectura de software utilizadas para el desarrollo de aplicaciones de RA.

En el Capítulo 3 se realiza un estudio de la aplicación de la RA en proyectos militares y se realiza una comparación de algunos proyectos militares que utilizan la RA como tecnología principal.

En el Capítulo 4 se detalla la importancia de la arquitectura de software y el diseño de un framework basado en capas y componentes.

En la parte II del informe se detalla el Framework Propuesto.

En el Capítulo 5 se detalla el diseño de la arquitectura del framework, la metodología de desarrollo utilizada, los componentes de hardware empleados en los despliegues y la implementación del proceso distribuido.

En el Capítulo 6 se detalla el despliegue de la solución, los dispositivos utilizados en las pruebas y la integración tecnológica.

En el Capítulo 7 se mencionan las conclusiones finales y las futuras líneas de investigación.

Al final del documento se encuentra la bibliografía referenciada a lo largo del informe.

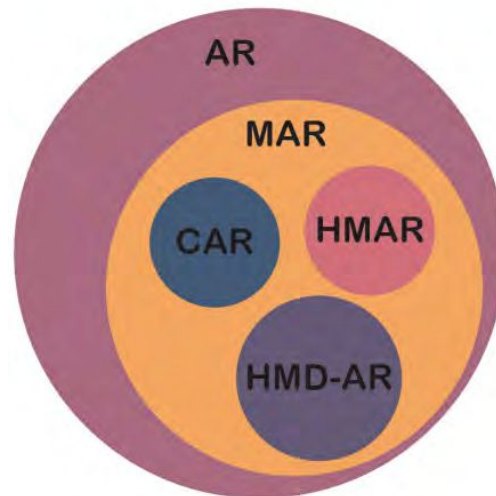
# Capítulo 2

## Fundamentos de la Realidad Aumentada

En este Capítulo se hace una introducción a los aspectos técnicos relacionados con la RA. Se menciona que es la RA, las partes de una aplicación de RA como también los métodos de entrada de datos, visualización de información contextual y las tecnologías actuales para lograr este objetivo. Adicionalmente, se detalla qué es el tracking, el reconocimiento gestual y se detalla la arquitectura de software para aplicaciones e RA.

### 2.1 ¿Qué es la RA?

De acuerdo a [6], la Realidad Aumentada (RA) proporciona al usuario información superpuesta que se puede visualizar en el mundo real, es decir, complementa al mundo real con información virtual. La RA mejora la percepción del mundo natural mediante el agregado de información a los sentidos, ya sean visuales, sonidos, olores o sensaciones táctiles. La RA se refiere a la mezcla de las señales virtuales a partir del entorno tridimensional real en la percepción del usuario. Denota la fusión 3D de imágenes sintéticas en la visión natural del usuario del mundo circundante utilizando gafas, dispositivos móviles o un HMD (en inglés, Head-Mounted Display). Antes de continuar detallando en qué consiste la RA, es importante mencionar el uso de dispositivos móviles que implementan técnicas de RA. De acuerdo a [7] existen diferentes tipos de implementación de RA en dispositivos móviles. En este caso MAR (en inglés, Mobile Augmented Reality) puede ser entendido como cualquier dispositivo móvil que utiliza RA. Estos dispositivos móviles puede ser HDM, HMAR (en inglés, Handheld Mobile Augmented Reality), CAR (en inglés, Contact-lens augmented reality) o cualquier tipo de tecnología móvil que utilice RA para visualizar el mundo aumentado. Cabe destacar que HMAR es un subconjunto de MAR específicamente referido al uso de dispositivos handheld que pueden visualizar RA tal cual se muestra en la siguiente figura:



**Figura 2.1– La RA abarca un amplio espectro de sistemas. Dentro del dominio MAR  
Se encuentran sistemas tales como CAR, HMAR y HMD basadas en RA  
(fuente: Gjosaeter, T. [7])**

A través de la capacidad de presentar la información superpuesta, integrados en el entorno del usuario, la RA tiene el potencial de proporcionar beneficios significativos en muchas áreas de aplicación. Muchos de estos beneficios surgen del hecho de que las señales virtuales presentadas por un sistema de RA pueden ir más allá de lo que es físicamente visible.

La información espacial puede estar alineada directamente con el medio ambiente. A través de la capacidad de presentar la información superpuesta, integrada en el entorno del usuario, la RA tiene el potencial de proporcionar beneficios significativos en muchas áreas de aplicación. Muchos de estos beneficios surgen del hecho de que las señales virtuales presentadas por un sistema de estas características pueden ir más allá de lo que es físicamente visible.

Un sistema de RA es un tipo de sistema de visión sintética que mezcla gráficos generados por una computadora (anotaciones) con el mundo real. Las anotaciones proporcionan información destinada a ayudar a la toma de decisiones. El sistema de RA debe decidir qué anotaciones mostrar y cómo mostrar para asegurar que la información sea intuitiva y sin ambigüedades.

Un sistema de RA móvil consiste en un dispositivo móvil, un sistema de seguimiento (tracking) y un HMD, handheld o smartphone. El sistema hace tracking de la posición y orientación del usuario y superpone, dentro del campo de visión del usuario, gráficos y anotaciones que están alineados con los objetos en el medio ambiente. Este enfoque tiene muchas ventajas. La información puede ser presentada de una manera intuitiva y directamente integrada con el medio ambiente [8].

Cuando se implementa correctamente, la RA proporciona al usuario una experiencia de inmersión y permite la interacción entre el mundo virtual y el mundo real. Azuma [9] pone restricciones en la definición de la RA con la adición de tres criterios: combina lo real y lo virtual, interactiva en tiempo real, y están registrados y alineados en 3D.

En [10] se define a la RA como una herramienta moderna que están explorando los investigadores es la tecnología de aumentación o ampliación de la percepción humana conocida como Realidad Aumentada. Esta tecnología combina datos virtuales con el ambiente real observado por el usuario. Se produce la registración de información virtual con la escena real desde el punto de vista del usuario.

En [11] se introduce un término más genérico, realidad mixta, y lo definen como un continuum en donde uno de los extremos persiste el ambiente real y en el extremo opuesto persiste el ambiente virtual. La Realidad Aumentada y la Virtualidad Aumentada constituyen la realidad mixta o intermedia entre estos dos extremos.



**Figura 2.2– Continuum de la virtualidad**  
(fuente: Carmigniani, J. et al [4])

Azuma define tres requerimientos para las aplicaciones de RA:

- 1) Se deben combinar los aspectos real y virtual
- 2) La aplicación debe ser en tiempo real e interactiva, y

### 3) Los objetos deben estar registrados en el mundo tridimensional (3D)

Aunque el término RA deriva de la industria aeronáutica, otras industrias y aplicaciones rápidamente adoptaron el término. Los simuladores de vuelo se citan como ejemplos de RA. No obstante, se considera a la RA a aquellas aplicaciones que combinan en una pantalla los escenarios reales y virtuales. Un buen ejemplo de representación visual de RA son los HMD. La información es proyectada de tal manera que un operador puede observar el mundo real directamente.

En el año 1992 Tom Caudell crea el término RA para describir una pantalla que usarían los técnicos electricistas de Boeing que mezclaba gráficos virtuales con la realidad física. La RA agrega información sintética a la realidad. La diferencia principal entre RV (Realidad Virtual) y RA es que por una parte RV implica inmersión del participante en un mundo totalmente virtual y por otra parte la RA implica mantenerse en el mundo real con agregados virtuales.

A diferencia de las aplicaciones de RV las aplicaciones de RA generalmente necesitan la movilidad del usuario, incluso hacia ambientes externos- en inglés se denominan aplicaciones outdoor. En dichas aplicaciones de realidad aumentada puede ser necesaria conocer la posición global del participante utilizando dispositivos como GPS y brújulas digitales.

Dependiendo del dispositivo de visualización usado las aplicaciones de realidad aumentada pueden basarse en:

- Gafas de *video see-through*
- Gafas de *optical see-through*
- Proyector
- Monitor
- Dispositivos móviles o HandHeld

En [12] se detalla que el ambiente real enriquecido o guiado con objetos virtuales puede resultar de mayor utilidad a los usuarios. Esta tecnología tiene diversas aplicaciones para la industria, la educación, el arte, etc. La principal diferencia con la realidad virtual es que no sustituye la realidad física, sino que superpone datos informáticos al mundo real.

El resultado final obtenido, permite al usuario el poder reconocer un objeto que este siendo apuntado en ese momento con la cámara del dispositivo y mostrar sobre éste, mediante realidad aumentada, información sobre el objeto. Específicamente la información extraída del escenario consiste de puntos característicos de la escena real detectados utilizando alguna técnica de tracking. En términos de tracking, existen dos categorías de registros:

- Basados en sensores (mecánicos, magnéticos, ultrasónicos, etc.)
- Basados en visión por computador.

En los basados en sensores se deben calibrar los mismos y son pesados y caros y los niveles de precisión son pobres.

En cambio, los basados en visión por computador pueden evitar la calibración de sensores externos y ofrecen un muy nivel de precisión sin ser equipos pesados y costosos. De acuerdo a la calibración de la cámara se puede mencionar:

- 1) No requiere calibración de los parámetros de la cámara (ya sean parámetros intrínsecos o extrínsecos), el cual involucra el uso de un objeto conocido de calibración 3D. Un método para resolver este problema es calcular los parámetros de la cámara y estructuras 3D en un espacio Euclidiano directamente de la información contenida en las secuencias de imágenes de entrada desde la cámara. Otra alternativa es ejecutar la tarea de RA en un espacio proyectivo o afin en vez de en un espacio Euclidiano.
- 2) Asume que los parámetros intrínsecos de la cámara están pre calibrados.

Bajo esta aproximación pragmática actual, para componer un servicio de RA son necesarios 4 ingredientes básicos [3]:

Para diseñar un servicio de RA se necesitan 4 elementos básicos:

- Por un lado, un elemento que capture las imágenes de la realidad que están viendo los usuarios.
- Por otro, un elemento sobre el que proyectar la mezcla de las imágenes reales con las imágenes sintetizadas.
- En tercer lugar, es preciso tener un elemento de procesamiento, o varios de ellos que trabajen conjuntamente.
- Finalmente se necesita un elemento al que podríamos denominar «activador de realidad aumentada». En un mundo ideal el activador sería la imagen que están



visualizando los usuarios, ya que a partir de ella el sistema debería reaccionar. Pero, dada la complejidad técnica que este proceso requiere, en la actualidad se utilizan otros elementos que los sustituyen. Se trata entonces de elementos de localización como los GPS que en la actualidad van integrados en gran parte de los Smartphone, así como las brújulas y acelerómetros que permiten identificar la posición y orientación de dichos dispositivos, así como las etiquetas o marcadores del tipo RFID o códigos bidimensionales, o en general cualquier otro elemento que sea capaz de suministrar una información equivalente a la que proporcionaría lo que ve el usuario, como por ejemplo sensores.

De acuerdo a [13] hasta hace muy poco tiempo las aplicaciones de RA se ejecutaban en poderosas computadoras de escritorio. La introducción de RA en los smartphones permitió nuevas experiencias de RA a los usuarios. La RA se ha transformado en una comodidad ubicua para el ocio y para el aprendizaje. Con esta capacidad de ubicuidad, los móviles de RA permiten idear y diseñar escenarios de aprendizaje innovadores en entornos del mundo real.

Por otro lado se puede definir a la RA como un sistema que mejora los sentidos primarios de las personas, es decir, visión, audición y táctil. Estos sentidos son mejorados o aumentados con información digital visible para el usuario. En este contexto, la información virtual incluye información como geo-localización, sobreimpresión visual/audio o información visual en 3D, entre otras.

Cabe destacar que al igual que los sistemas sensibles al contexto, las aplicaciones de RA permiten filtrar información y presentar información superpuesta relativa al contexto actual del usuario. La información del contexto puede ser filtrada de acuerdo a la ubicación, camino a recorrer, objeto en foco, período de tiempo o cualquier otra meta información tales como intereses o el perfil personal de la persona.

Por lo tanto se define a la RA móvil como una especialización de los sistemas sensibles al contexto ya que existe una relación muy cercana entre los sentidos del humano y la percepción con la información digital presentada en RA.

De acuerdo a [14], los componentes que forman parte de un sistema de RA móvil deben contener:

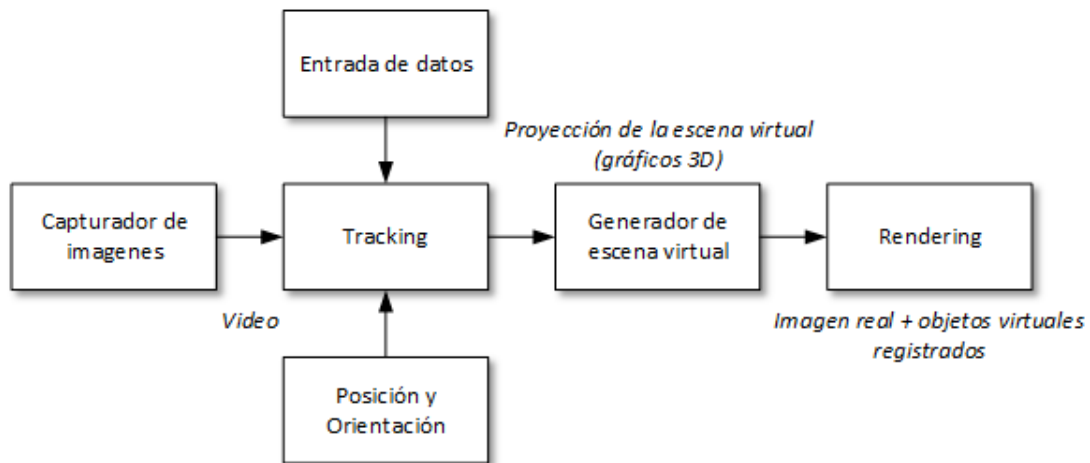
- Sistemas de pantalla flexible, incluyendo un sistema HMD, cámaras en los Smartphone, y proyectores de mano.
- Sistemas de sensores en los dispositivos móviles tales como GPS, brújulas electrónicas, micrófono inclusive sistemas de tracking.
- Conexión a las redes de datos mediante tecnología inalámbrica.
- Poder computacional alto para el procesamiento de imágenes y visualización de objetos en 3D y sobreimpresión de información.
- Tecnología de tracking con 6 grados de libertad, tracking multi marcador y sistemas de tracking híbrido.
- Vinculación de la información de RA basada en localización
- Integración de navegadores de RA con los medios sociales.

La RA ha sido ampliamente demostrada y aplicada en dispositivos móviles, ejecutándose diversas aplicaciones tales como juegos, navegación, referencias, etc. Otro aspecto que está influyendo en las aplicaciones de RA es la tendencia a dirigirse hacia entornos donde se requiera la movilidad del usuario. Estas nuevas aplicaciones basadas en la computación móvil requieren acceder a servicios independientemente del lugar o del tiempo.

## **2.2 Partes de una aplicación de RA**

De acuerdo a [15] una aplicación de realidad aumentada tiene las siguientes partes:

- Captura de la escena real: El video capturado puede utilizarse para tracking basado en visión, es decir basado en el análisis de la imagen mediante algoritmos de visión
- Tracking del usuario: puede realizarse mediante dispositivos específicos o puede realizarse tracking basado en visión para lo cual es necesaria la captura de la escena real.
- Generación de la escena virtual: se tiene un mundo virtual, con la información de la posición y orientación del participante se genera una vista acorde del mismo
- Rendering: se combinan las imágenes del mundo real con los objetos virtuales. Los objetos virtuales se renderizan y se proyectan en el dispositivo móvil.



**Figura 2.3 – Diagrama de las partes de una aplicación de RA  
(fuente: propia)**

Adicionalmente se agrega un componente más:

- Entrada de datos: se captura el video y se toma como entrada de datos el reconocimiento gestual o el reconocimiento por voz para ejecutar un comando determinado

Por otro lado en [9] especifica las partes o componentes principales de un Sistema de RA:

1. Generador de Escena: En los sistemas de RA las imágenes virtuales complementan al mundo real a diferencia de los entornos de RV que necesitan mayor requerimientos para las imágenes ya que reemplazan completamente al mundo real. En el caso de los sistemas de RA las anotaciones, textos, dibujos en 3D podría ser suficiente para representar cierta información.
2. Dispositivo de visualización: En los sistemas de RA los dispositivos de visualización puede ser menos rigurosos que los utilizados en RV ya que no reemplazan el mundo real sino que lo complementa. Por ejemplo, pantallas monocromáticas puede ser útiles para algunos sistemas de RA, mientras que los sistemas de RV deben utilizar pantallas a color. Otro ejemplo son los HMD de visión óptica (*HMD optical see-through*), incluso pueden ser satisfactorios debido que el usuario puede ver el mundo real con su visión periférica. También se puede mencionar los dispositivos móviles tales HMAR, CAR, smartphone, entre otros.

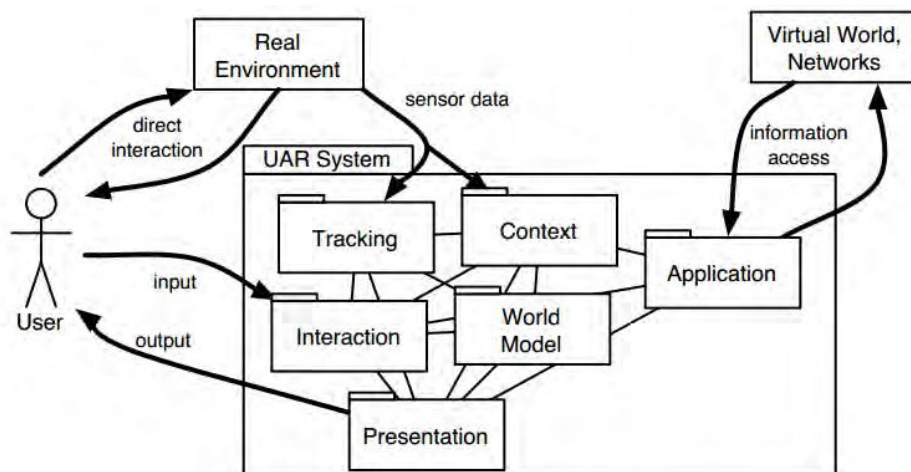
3. Tracking (seguimiento) y detección: En este caso el tracking y la detección juega un rol importante en los sistemas de RA. En esta área los requerimientos para los sistemas de RA son muchos más estrictos que los sistemas de RV. La principal razón es el problema de registración de la información.

### Modelo de referencia

Un Modelo de Referencia en ingeniería de sistemas y en ingeniería de software es un modelo de algo que contiene un objetivo o idea básica de algo, y que se puede establecer como una referencia para múltiples propósitos.

También se puede definir como un marco de referencia abstracto para entender el significado de las relaciones entre entidades de algún ambiente. Permite el desarrollo de referencias específicas o de arquitecturas por medio del uso de estándares o especificaciones que soportan el ambiente en cuestión. Un Modelo de Referencia consiste de un conjunto mínimo de conceptos, axiomas y relaciones propios de un dominio particular de problema, y es independiente de estándares específicos, tecnologías, implementaciones, o de cualquier otro detalle concreto.

Un estudio de varios sistemas de RA reveló que a pesar de ser bastante diferentes en los detalles, la mayoría de los sistemas de RA existentes comparten una estructura arquitectónica en común. En base a esto, se diseñó una arquitectura de referencia [16] como la que se muestra en la siguiente figura:

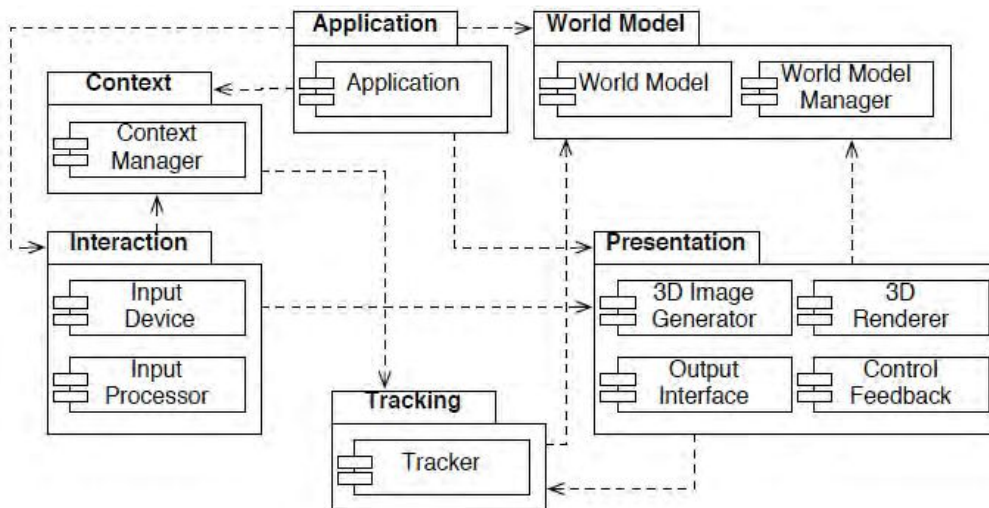


**Figura 2.4 – Modelo de referencia. Componentes de software requeridos en un sistema de RA ubicua**

(fuente: MacWilliams, A. [16])

El Modelo de Referencia se organiza en seis subsistemas lógicos:

- *Subsistema de seguimiento (tracking)*: Responsable de responder a los cambios de posición y orientación del usuario.
- *Aplicación*: Responsable del control de flujo, la lógica de la aplicación y la coordinación.
- *Modelo del mundo*: provee acceso a la representación digital del mundo.
- *Presentación*: Sistema responsable de la presentación de los resultados. Incluye renderizado 2D/3D y salida de audio o táctil.
- *Contexto*: provee a la aplicación con contexto tanto del estado como de la situación del usuario.
- *Interacción*: representa la interacción del usuario con el sistema de RA.



**Figura 2.5– Arquitectura de referencia genérica de sistemas de RA**

(fuente: MacWilliams, A. [16])

Cada subsistema está formado por varios componentes. Los subsistemas se muestran como un paquete en UML lo cuales están compuestos por componentes. Por otro lado,

cada componente a su vez puede ser obtenido por medio de otros componentes. Hay relaciones de dependencia entre los subsistemas. Una dependencia muestra que un subsistema se basa en las interfaces de otro subsistema.

El modelo de referencia describe a nivel abstracto los componentes generales y la estructura de los sistemas de realidad aumentada. Sin embargo, dependiendo de los requisitos funcionales de un sistema particular, algunos de los componentes pueden dejarse afuera [17].

## **2.3 Métodos de entrada de datos en un sistema de RA**

Resulta importante entender los métodos de entrada de datos en un sistema de RA. Esta importancia se denota ya que poder ingresar comandos para una determinada acción sin utilizar, por ejemplo, el teclado se transforma en una posibilidad a tener en cuenta sobre todo cuando es primordial tener las manos libres de periféricos para el ingreso de información. En los sistemas de RA se pueden mencionar los siguientes mecanismos de toma de datos:

De acuerdo a [15] se establece los siguientes mecanismos de entrada:

- Controles físicos
- Control del habla
- Control por gestos

En un sistema de RA podemos encontrar dentro de los controles físicos a los dispositivos móviles, sean estos smartphones, tablets, etc. Por otro lado el control del habla es muy adecuado cuando se tienen las manos ocupadas. Mientras que el control por gestos es un método natural de comunicar la información en el que se basan nuevas metáforas de interacción.

Requiere de dispositivos especiales para el tracking o en su lugar de la aplicación de técnicas de visión por computador.

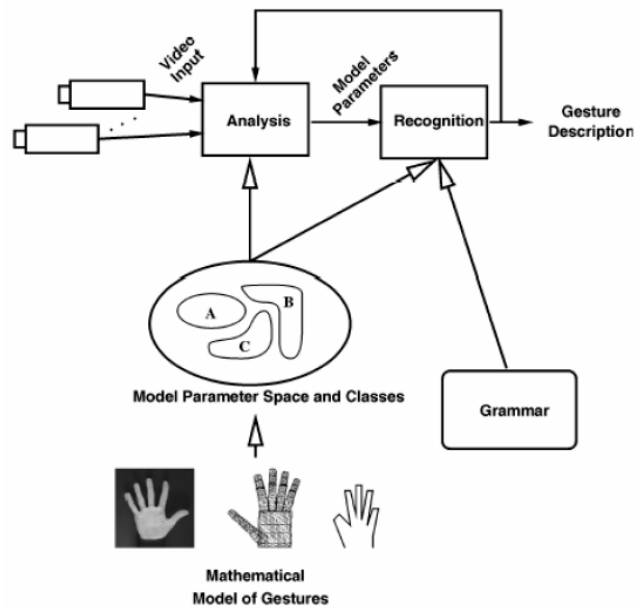
Kölsch et al [18] mencionan la existencia de las interfaces multimodales donde se combinan, por ejemplo, reconocimiento de voz y gestos. Por lo tanto se puede llegar a la siguiente conclusión:

- Aquellas tareas que requieren parámetros adimensionales, es decir, no se debe indicar nada en el espacio, pueden indicarse adecuadamente mediante el habla
- Las tareas que requieren parámetros espaciales de una dimensión como un desplazamiento en un eje (por ejemplo, indicar un zoom)
- Pueden indicarse adecuadamente mediante un dispositivo como un trackball unidireccional.
- Las tareas que requieren parámetros espaciales de dos dimensiones y tres dimensiones (posicionamiento, escalado y orientación de objetos en el mundo virtual) requieren utilizar diferentes tipos de entrada combinando gestos, movimientos de cabeza y trackball unidireccional.

Hay muchos tipos de dispositivos de entrada los sistemas de RA. Algunos sistemas utilizan guantes para ingresar datos, otros utilizan algún sistema inalámbrico. En el caso de los smartphones pueden ser utilizados como dispositivos de orientación o posicionamiento como es el caso de Google Sky Map para el sistema operativo Android [4].

La interacción humana con la computadora ha sido por años una manera de comunicación centrada en las máquinas. El reconocimiento gestual es una forma natural de expresión humana, por lo humano-computadora (en inglés, Human-Computer Interaction –HCI-). Una solución es utilizar técnicas de visión para el reconocimiento gestual. La mayoría de las estrategias de reconocimiento gestual basadas en visión por computador se centran en reconocimiento de gestos estáticos conocido como posturas. Sin embargo, se ha argumentado que el movimiento de los gestos transmite tanto significado como la propia postura. Los movimientos incluyen, por ejemplo, movimiento global de la mano y análisis del movimiento del dedo [10]. La interpretación de los gestos puede dividirse en tres fases: modelado, análisis y reconocimiento.

- *Modelado*: El modelado de los gestos involucra la descripción esquemática de un sistema de gestos que da cuenta de sus propiedades conocidas o inferidas.
- *Análisis*: el análisis del gesto involucra el cálculo de los parámetros del modelo basado en las características de la imagen detectada por la cámara.
- *Reconocimiento*: esta fase involucra la clasificación de los gestos basado en los parámetros del modelo procesado.



**Figura 2.6 – Sistema de reconocimiento gestual**  
**(fuente: McDonald, C. [10])**

Un acercamiento muy interesante es el que se menciona en [19] donde se detalla una interacción más precisa en el reconocimiento gestual en lo que se refiere al tracking para la interacción de la interfaz del usuario en RA dónde se evalúa la usabilidad y la precisión. Esta solución combina algoritmos para la extracción del contorno de la mano para aplicarlo en una aplicación de RA. De esta manera, se ocupa de los siguientes requisitos:

- Selección de los ítem del menú utilizando reconocimiento gestual
- Robustez y precisión en la detección de la mano y tracking en condiciones de variación de la iluminación con oclusión limitada
- Uso de la mano como un puntero en 3D

La facilidad de uso y la independencia de los gestos se abordan mediante la definición de cuatro posturas intuitivas de las manos que sirven para la navegación de los menús y la selección. Las tareas de navegación y selección son soportadas por detección automática de objetos de la mano y extracción de formas de la mano.



Una de las técnicas más populares y robustas utilizadas para la detección de la mano es la propuesta por Viola & Jones. Este método utiliza Adaboost un método supervisado que ejecuta clasificación binaria. Durante el entrenamiento, se genera iterativamente un clasificador denominado fuerte agregando un clasificador débil cuya función es adivinar al azar. Cada clasificador débil es entrenado en un conjunto de datos generado donde se aplica muestreo o ponderación en los casos del conjunto de datos original de la mano.

Active Appearance Model (AAM) es un modelo estadístico que maneja variaciones de formas y texturas de la apariencia foto realística. AAM utiliza un conocimiento previo de la apariencia en escalas de grises, estructura de la forma inclusive sus relaciones, con el objetivo de construir modelos generativos para un análisis global de una clase específica de objetos. Esta técnica se utiliza para recuperar formas en 2D de la postura de la palma y puño de la mano en un sistema de RA, siguiendo una serie de análisis locales, alrededor del área rectangular de la mano utilizando la técnica de Viola & Jones. En RA cuando se utiliza el detector Viola & Jones de la palma de la mano para estimar la ubicación de la punta del dedo índice, y AAM para permitir un puntero basado en el dedo.

## 2.4 Visualización

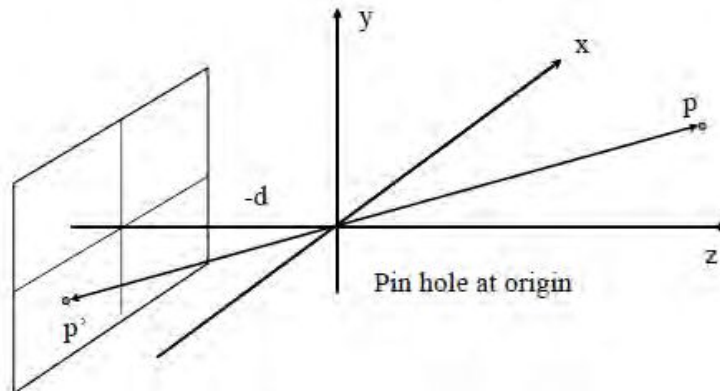
### Sistemas de Coordenadas

El principal desafío para los sistemas de RA es cómo combinar el mundo real y el mundo virtual en un único entorno.

Como menciona [15] existen dos tipos de proyecciones más conocidas y utilizadas en informática gráfica: proyección ortogonal y proyección perspectiva.

La proyección perspectiva puede describirse por un modelo matemático sencillo denominado “modelo de Pinhole”. Este modelo de cámara es utilizado en computación gráfica y permite modelar la transformación proyectiva de una escena de tres dimensiones en un plano de dos dimensiones. En este modelo de cámara, el lente de la cámara (pin-hole) se encuentra en el origen y un punto  $p$  se proyecta sobre la película en el punto  $p'$

La distancia entre la película fotográfica y el lente se conoce como longitud focal  $d$  [10]



Usando este modelo, se puede definir la relación entre las coordenadas 3D en la escena virtual,  $x$  e  $y$ , y el resultado de las coordenadas de la imagen en 2D,  $x'$  e  $y'$

$$x' = d \frac{x}{z} \quad \text{and} \quad y' = d \frac{y}{z}$$

En su forma general, esta relación puede ser representada a través de la siguiente transformación homogénea:

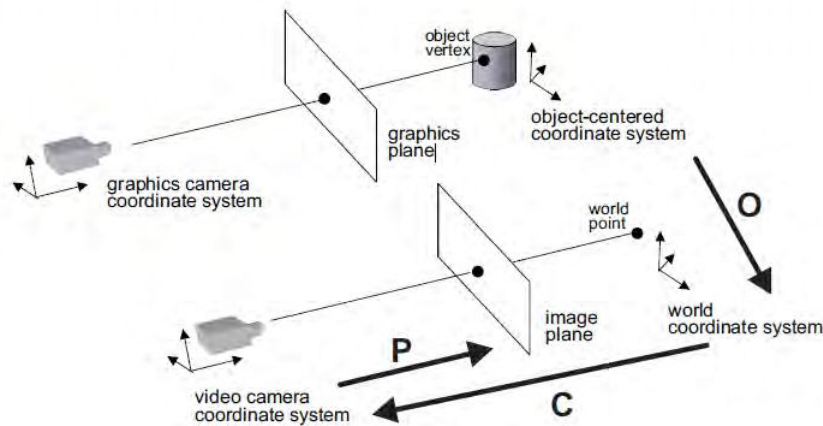
$$p' = Mp,$$

donde  $p$  y  $p'$  son puntos homogéneos y  $M$  es la matriz de proyección 4x4:

$$\begin{bmatrix} x' \\ y' \\ z' \\ w \end{bmatrix} = \begin{bmatrix} d/z & 0 & 0 & 0 \\ 0 & d/z & 0 & 0 \\ 0 & 0 & d/z & 0 \\ 0 & 0 & 1/z & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Para obtener esta matriz de proyección para una posición arbitraria de la cámara en el espacio, los parámetros intrínsecos y extrínsecos de la cámara se deben extraer independientemente.

Para mantener esta ilusión al usuario, de que los objetos virtuales en realidad son parte del mundo real, requiere una fusión coherente entre el mundo virtual con el mundo real.



**Figura 2.7 – Sistemas de Coordenadas de un sistema de RA**  
**(fuente: McDonald, C. [10])**

El principal requisito es determinar las relaciones entre ambos mundos. Estas relaciones son del sistema de coordenadas objeto a mundo  $O$ , de mundo a cámara  $C$ , y de cámara a una imagen plana  $P$ . De objeto a mundo se refiere a la posición y orientación de un objeto virtual con respecto al sistema de coordenadas del mundo que define la escena real. La de mundo a cámara define la posición de la videocámara que apunta hacia la escena real. Por último, de cámara a una imagen plana especifica la proyección que la cámara realiza para crear una imagen 2D a partir de la escena real en 3D.

La creación de una imagen tridimensional de un objeto virtual en consonancia con la vista actual del usuario en el mundo real y la colocación del objeto que está viendo en dicho mundo, requiere la definición de las relaciones geométricas entre los objetos virtuales y los físicos.

Como también se especifica en [10] el sistema de coordenadas del mundo es el punto de referencia inicial. Luego, el sistema de coordenadas del video cámara se debe determinar utilizando alguna estrategia de visión por computador. La transformación del sistema de coordenadas del mundo al sistema de coordenadas de la cámara está representado por  $C$ .

La transformación proyectiva definida por medio del modelo de cámara está representada por  $P$ . La transformación final necesaria para llevar a cabo la registración de manera adecuada es la transformación del sistema de coordenadas centrado en el objeto al sistema

de coordenadas del mundo, O. Las coordenadas 3D del objeto virtual son asignadas a priori, por lo que esta transformación se construye en ese momento.

Con las dos cámaras alineadas, la fusión de los componentes reales y sintéticos de la escena será debidamente registrada.

De acuerdo a [15], en el proceso de formación de imágenes, reales o sintéticas, existe:

- una *escena* compuesta por diversos objetos (en diferentes posiciones y orientaciones)
- una *cámara*, con una cierta posición y orientación en la escena
- una *imagen* resultante de proyectar la escena de acuerdo a la cámara

#### *Sistema de coordenadas local*

Es un sistema de coordenadas 3D utilizado para referenciar los puntos de un objeto. Si se modela un árbol seguramente lo haremos situando un sistema de coordenadas centrado en el árbol o en la base del mismo, con un eje alineado con el tronco.

#### *Sistema de coordenadas mundo*

La posición y orientación de los objetos de una escena suele expresarse en relación a un sistema de coordenadas 3D situado en algún lugar del mundo. Todos los objetos que integran una escena se posicionan, orientan y escalan en relación al mismo sistema de coordenadas mundo.

#### *Sistema de coordenadas cámara*

El sistema de coordenadas cámara 3D tiene el origen en el centro óptico de la cámara. El eje de proyección de la cámara pasa por el centro óptico y es perpendicular al plano de formación de la imagen. El eje Z del sistema de coordenadas cámara está alineado con el eje de proyección, con Z+ hacia donde se ubica la escena. El eje Y tiene dirección vertical y el eje X dirección horizontal.

El sistema de coordenadas mundo 3D puede expresarse en relación al sistema de coordenadas cámara 3D mediante transformaciones geométricas. De forma equivalente, el sistema de coordenadas cámara puede expresarse en relación al sistema de coordenadas mundo.

### *Sistema de coordenadas de la imagen*

Los puntos de la imagen se expresan en relación a un sistema de coordenadas 2D con los ejes alineados con los bordes de la imagen.

Es importante destacar lo que menciona [20] en lo que se refiere a los sistemas de coordenadas cuando se manipulan objetos 3D. Los autores evalúan el impacto de los tres sistemas de coordenadas: cámara, objeto y mundo, sobre la rotación y la traslación de un objeto 3D en un escenario de RA. Los ejes del Sistema de coordenadas de la cámara dependen de la pose actual del dispositivo móvil, también conocido como la vista del usuario. La vista del usuario se utiliza frecuentemente como sistema de referencia en los dispositivos móviles de RA. Cada objeto en el espacio tiene su pose 3D específica. La característica del sistema de coordenadas del objeto es la dependencia de los ejes entre la traslación y la rotación. El Sistema de coordenadas del mundo es el único sistema de coordenadas estático y representa el espacio en el mundo real.

Por otro lado en [4] se mencionan algunos métodos de visión por computador utilizados en RA. La registración de la imagen en RA utiliza diferentes métodos de visión por computador generalmente relacionado con el tracking de video. Estos métodos usualmente consisten de dos etapas: tracking y reconstrucción/reconocimiento. Primero, los marcadores fiduciaros, imágenes ópticas o puntos de interés son detectados en las imágenes de la cámara. El tracking puede utilizar métodos de procesamiento de imágenes tales como detección de características, detección de contornos u otros métodos a fin de interpretar las imágenes de la cámara.

En visión por computador la mayoría de las técnicas de tracking pueden separarse en dos clases: basados en características y basados en modelos. El primer método consiste de descubrir la conexión entre las características de la imagen en 2D y su las coordenadas del mundo en 3D. El segundo método utiliza modelos de los objetos a trackear tales como modelos CAD o plantillas en 2D.

Una vez que se realiza la conexión entre la imagen en 2D y el frame del mundo en 3D, es posible encontrar la pose de la cámara proyectando las coordenadas 3D de la característica en las coordenadas de la imagen en 2D y minimizando las distancias a sus características correspondientes en 2D.

Las limitaciones de la estimación de la pose de la cámara son frecuentemente determinadas utilizando las características de los puntos.

La etapa de reconstrucción/reconocimiento utiliza los datos obtenidos de la primera etapa para reconstruir el sistema de coordenadas del mundo real.

### **Registro de información**

La RA se ha transformado en una tecnología prometedora para soportar tareas en 3D. La registración es una de las claves más importantes de esta tecnología. El proceso de registro es la superposición de objetos virtuales en una escena real utilizando información extraída de la escena [21].

Como menciona [15] la información virtual tiene que estar vinculada espacialmente al mundo real de manera coherente, lo que se denomina registro de imágenes (“registration” en inglés). Por esto se necesita saber en todo momento la posición del usuario, tracking, con respecto al mundo real.

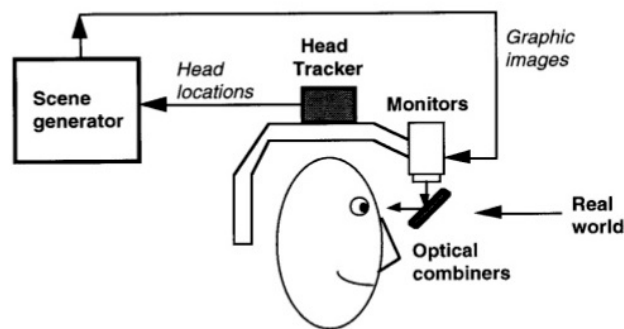
Cuando se habla de posición y orientación de un objeto o de una cámara debe existir un sistema de referencia en base al cual se expresan. En RA, al hablar de imágenes registradas significa que tanto las imágenes sintéticas como el mundo real estén en referencia al mismo sistema de coordenadas. Es importante destacar la registración geométrica de los objetos sintéticos en la escena real. Para ello [22] detalla que la registración geométrica del contenido virtual dentro del ambiente real del usuario es la base para una aplicación de RA. Una correcta registración geométrica le permite al observador aceptar el contenido virtual como una mejora del ambiente real más que una separada o superpuesta capa de información. La escena virtual se registra geoméricamente midiendo o estimando la pose actual del punto de vista del usuario o la cámara con respecto al sistema de coordenadas del contenido virtual.

## **2.5 Tecnologías para visualizar la RA**

Azuma, R [9] menciona para construir un sistema de RA es importante determinar qué tecnología utilizar para combinar el mundo real con el digital. Para ello existen dos tipos de tecnologías: óptica y video. Un HMD *see-through* es un dispositivo que se utiliza para para combinar el mundo real con el digital. Este tipo de dispositivo permite ver el mundo real

sobreimprimiendo objetos digitales a través del uso de la tecnología óptica o por video. La tecnología óptica trabaja combinando objetos digitales en frente del campo visual del usuario. Generalmente se utiliza un combinador óptico transparente del tipo Head-Up Displays (HUDs) como se utiliza en los cascos de los pilotos de aviones de combate. El combinador óptico, usualmente reduce la cantidad de luz que usuario visualiza en el mundo real para lograr una alta absorción de la luz se utilizan proyectores del tipo monocromáticos o gafas oscuras donde directamente reducen la onda de luz proveniente del mundo real.

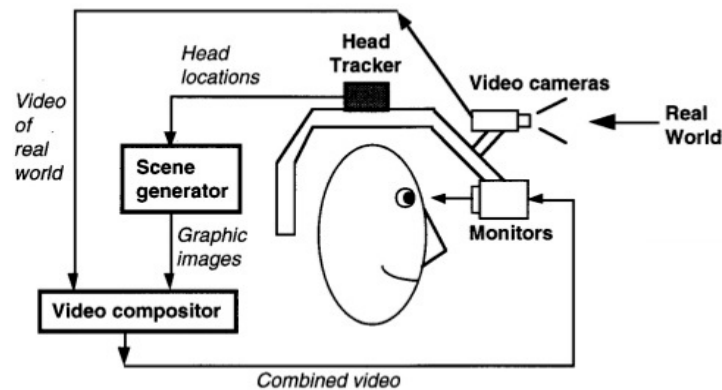
En la siguiente figura se muestra el diagrama conceptual del uso de la tecnología *optical see-through* utilizando un HMD:



**Figura 2.8 – Diagrama conceptual de la tecnología *optical see-through***  
(fuente: Azuma, R [9])

Por otro lado la tecnología *video see-through* trabaja combinando, en un HDM cerrado, una o más video cámaras (monocular o binocular, respectivamente) montadas en el casco. Las videos cámaras proveen al usuario la vista del mundo real. El video de estas cámaras se combina con las imágenes gráficas creadas por el generador de escenas, mezclando el mundo real con el digital. El resultado se envía a los monitores (o pantallas) en frente de los ojos del usuario.

En la siguiente figura se observa el diagrama conceptual de la tecnología *video see-through*:



**Figura 2.9– Diagrama conceptual de la tecnología *video see-through***  
(fuente: Azuma, R [9])

Cabe aclarar que si bien la técnica de proyección recomendada es *optical see-through*, a través del uso de gafas de RA ya que el principal desafío es que el operador no pierda la percepción de la realidad pues al ejecutarse operaciones militares de alto riesgo como premisa principal es no perder la conciencia y percepción del entorno, adicionalmente, y a fin de evaluar la versatilidad tecnológica, se realizaron pruebas mediante *video see-through* utilizando las *cardboard box*.

## 2.6 Tracking

En [15] define al tracking (seguimiento) como el proceso de seguimiento de un objeto en movimiento, es decir, la estimación de la posición y orientación (pose) del mismo en cada instante.

En una aplicación de RA se necesita el tracking del participante para conocer la matriz de transformaciones geométricas y así realizar el registro de imágenes sintéticas y reales.

En la mayoría de aplicaciones de RA se trata de realizar un tracking de la cámara que captura la escena. Aunque también puede tratarse del tracking de la cabeza del usuario o de algún objeto manipulado por este.

Los autores también mencionan que el tracking en una aplicación de realidad aumentada puede hacerse:

- mediante dispositivos físicos específicos



- tracking basado en visión
- tracking híbrido, que combina las salidas de dispositivos físicos con el análisis de la imagen

Por otro lado [4] menciona que aunque el tracking visual tiene la habilidad para reconocer y seguir varias cosas, mayormente utiliza tecnologías tales como GPS y acelerómetros.

También en [23] menciona métodos de visión por computador se utilizan para el tracking de RA. A diferencia de otras tecnologías, los métodos de visión pueden estimar la pose directamente de la misma imagen que el usuario observa.

En [4] detalla que los métodos de tracking en RA dependen mayormente del tipo de ambiente que el dispositivo móvil se encuentre inclusive el tipo de sistema de RA. El ambiente puede ser interior, exterior o una combinación de ambos. De la misma manera, el sistema puede ser móvil o estático (con una posición fija). Si el dispositivo de RA es móvil y se diseñó para ambientes exteriores. El tracking será mucho más duro y dependiendo de las técnicas implementadas ofrecerán algunas o desventajas.

El tracking por visión por computador es eficiente para calcular la pose de la cámara en relación a la escena objetivo. Si bien el poder de hardware de los dispositivos móviles aún no es del todo poderoso se han explorado técnicas de tracking mediante características naturales.

En [24] menciona que el tracking de objetos mediante visión por computador es uno de los tópicos de investigación más activos en visión por computador por su importancia en teleconferencias, vigilancias, seguridad, interacción humano computadora, comprensión de video y edición de video, etc.

La tarea de encontrar correspondencias entre las imágenes de una misma escena o el objeto es esencial en las aplicaciones de visión por computador. Esto puede ser implementado utilizando tres pasos: detección, descripción y comparación (matching)

En el paso de detección se seleccionan los puntos de interés de distintos lugares de la imagen tales como esquinas y bordes. Estos puntos de interés deben ser distintivos y repetibles, es decir, deben poder detectarse bajo diferentes condiciones visuales.

En el paso de descripción, los puntos vecinos de cada punto de interés se representan por medio de un vector de características. Este proceso debe ser robusto a ruidos, detección de errores y deformaciones geométricas y fotométricas.

Por último, en la etapa de comparación, los vectores de características de las diferentes imágenes son comparadas. Esto se hace usualmente calculando la distancia Euclidiana entre los vectores de características.

En [25] recientes técnicas de comparación (matching) de imágenes tal como BRIEF reducen el tiempo omitiendo el cálculo del factor invariante de la escala, y en cambio, se considera variaciones multiescala de la imagen destino en el pre-procesamiento.

En [26] determina que el tracking de objetos es pertinente en las tareas de:

- Reconocimiento basado en movimiento, es decir, identificación humana basada en el paso, detección automática de objeto, etc.:
- Vigilancia automatizada, es decir, monitorear una escena para detectar actividades sospechosas;
- Indexación de video, es decir, anotación automática y recupero de los videos desde una base multimedia;
- Interacción humano computadora, es decir, reconocimiento gestual, seguimiento del iris ocular para ingresar datos a la computadora, etc.
- Monitoreo de tráfico, es decir, obtener en tiempo real las estadísticas de tráfico vehicular con el objetivo de redirigir el flujo de tráfico.
- Navegación vehicular, es decir, planificación del camino basado en video con capacidades de evitar los obstáculos.

Tracking puede ser definido como el problema de estimar la trayectoria y como se mueve un objeto en la imagen alrededor de una escena. Adicionalmente, un tracker puede proveer información del objeto como ser la orientación, área, forma, etc., de ese objeto. El tracking de objetos puede ser complejo debido a:

- Pérdida de información causada por la proyección del mundo 3D en una imagen 2D,
- Ruido en las imágenes,
- Complejidad en el movimiento del objeto,
- Oclusión total o parcial del objeto,

- Complejidad en las formas del objeto,
- Cambios en la iluminación de la escena, y
- Requerimientos de procesamiento en tiempo real.

Se puede simplificar el tracking mediante la imposición de restricciones en el movimiento y/o la apariencia de los objetos. Por ejemplo se puede aplicar una restricción en el movimiento del objeto para que sea constante basado en información previa. En conclusión información preliminar de los objetos, como el tamaño, la apariencia y la forma puede ser utilizado para simplificar el problema.

Se han propuesto diferentes soluciones para el tracking de objetos. Para llegar a una solución adecuada las diferentes propuestas se basan al menos en las siguientes preguntas:

- ¿Cuál representación del objeto es adecuado para tracking?
- ¿Cuáles características de la imagen debe ser utilizada?
- ¿Cómo debería ser modelado el movimiento, apariencia y forma del objeto?

Las respuestas a estas preguntas dependen del contexto/entorno en el que se realiza el seguimiento y el uso final para el que se solicita la información del tracking.

Como menciona [15] según la aplicación de RA que se desarrolle el tracking puede estar implementado en función de lo que se interesa conocer:

- La posición y orientación de la cámara en un sistema de coordenadas global
- La posición y orientación de la cámara en relación a un objeto de la escena real en cuestión
- La posición y orientación de la cámara en relación a la posición y orientación del cuadro de video anterior

## **2.6.1 Tracking mediante dispositivos físicos específicos**

Los sensores proveen otra interesante oportunidad o soporte a los desarrolladores de aplicaciones de visión por computador. Muchos dispositivos cuentan con sensores tales como:

acelerómetros, giróscopos, brújulas y GPS. Estos sensores son muy útiles a la hora de reconstruir la orientación de la cámara o la trayectoria en 3D.

El problema de extraer el movimiento de la cámara a partir de un conjunto de frames es desafiante, en términos de performance y precisión. Por ejemplo, SLAM (simultaneous localization and mapping), SfM (structure from motion) y otras soluciones pueden calcular la posición de la cámara e incluso la forma de los objetos que ve la cámara, pero estos métodos no son fáciles de implementar, calibrar y optimizar y además necesitan mucho poder de procesamiento.

Los sensores, no obstante, pueden proporcionar una estimación bastante precisa de la orientación del dispositivo. Para obtener resultados precisos el sensor debe ser utilizado sólo como punto de partida, para ser refinado mediante técnicas de visión por computador [27].

Por otro lado [28] detalla la fusión de tracking utilizando métodos visuales e inerciales y menciona como superar algunas limitaciones fundamentales en el tracking visual utilizando la combinación de estas dos técnicas.

- *Gran poder de cómputo*: el procesamiento de imagen usado para generar mediciones de la cámara es computacionalmente intensivo, sobre todo sin una secuencia de imagen continua y una predicción razonable de la pose de la cámara. Mediante la explotación de la información proporcionada por los sensores inerciales se puede obtener una predicción más exacta de la cámara. La predicción mejorada permite reducir el costo computacional del procesamiento de imágenes.
- *Movimientos lentos de la cámara*: los sistemas de tracking basados puramente en visión usualmente pueden manejar únicamente movimientos lentos de la cámara. Esto es debido a la demanda computacional y también debido a las características inherentes de la cámara tales como el desenfoque de movimiento. Los sensores inerciales pueden ser muestreados a una tasa mucho más alta que las cámaras y los algoritmos son computacionalmente menos intensivos.
- *Oclusión*: si muchas características están ocluidas, la pose de la cámara no puede ser estimada utilizando técnicas de visión por computador. Los sensores inerciales, en general, brindan mediciones. Por lo tanto la pose de la cámara puede ser actualizada cada tanto.

- *Jitter*: las técnicas de visión por computador tienden a introducir saltos (jitter). La razón es que la estimación actual de la pose por lo general se basa sólo en el conjunto de características visible. La cantidad de *jitters* aumenta cuando el conjunto de características cambia significativamente entre los frames.

## 2.6.2 Tracking basado en visión

Existen diferentes técnicas para realizar tracking basado en visión. Las mismas pueden categorizarse en dos grandes ramas:

- Tracking de marcadores
- Tracking basado en características naturales

### Tracking de marcadores

Un marcador en el contexto de RA se refiere a lo que se conoce en idioma inglés como “fiducial markers”. Es tipo de maracador es cualquier objeto que se puede colocar en una escena para proporcionar un punto fijo de referencia. En RA, estos marcadores pueden proporcionar una interfaz entre el mundo físico y el contenido virtual que proporciona la RA, tales como modelos 3D o vídeos. Estos marcadores permiten al dispositivo que está generando el contenido virtual calcular la posición y la orientación, de su cámara. Cuando este cálculo se realiza en tiempo real, este proceso se conoce como tracking. El tracking de marcadores se puede lograr utilizando sólo información visual. En la práctica esto significa que el tracking de marcadores se puede lograr con sólo una cámara y una PC de escritorio; no es necesario ningún sensor adicional.

De acuerdo a [29] menciona que la información que proviene de la imagen de entrada se compone básicamente de características de la imagen que son fáciles de extraer de la escena. Las técnicas de tracking basadas en marcadores y las de características naturales (sin marcadores) se basan en este principio, con la diferencia de que el primero añade algunos marcadores o plantillas que originalmente no pertenecen a la escena con el fin de permitir el tracking.

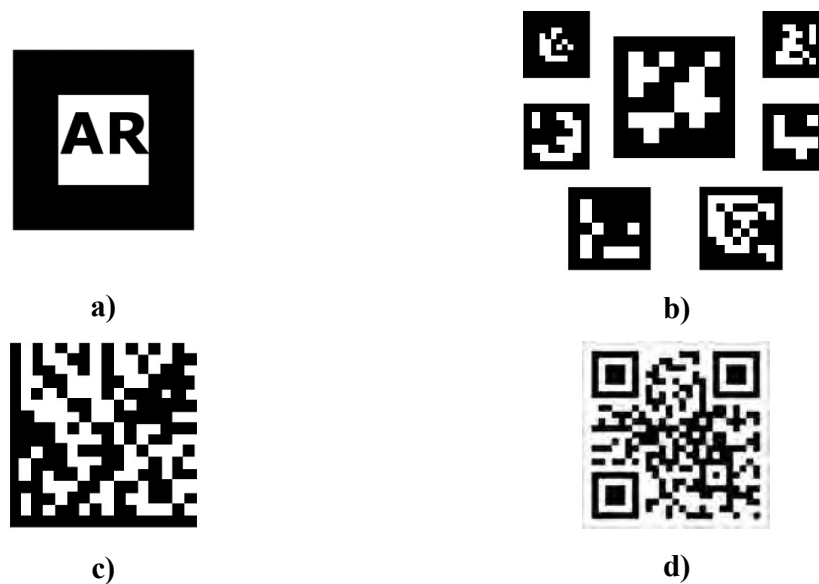
En [15] se señala que el algoritmo de tracking basado en marcadores está formado por dos pasos principales:

- Detección e identificación de marcadores
- Estimación de la matriz de transformaciones geométricas de cada marcador detectado

En cada cuadro capturado se aplica un algoritmo de detección de marcadores seguido de una identificación de cada marcador.

Adicionalmente en [15] se detalla los tipos de marcadores utilizados: Template, ID-marker, DataMatrix y QR. Los marcadores ID-marker codifican un número de 9-bits (hasta 512 diferentes) en un patrón de 6 x 6, repitiendo los 9 bits 4 veces completando los 36 bits. Los marcadores DataMatrix y marcadores QR no fueron diseñados específicamente para RA, sino que su propósito inicial es codificar una serie de caracteres ASCII, como por ejemplo, una dirección URL.

A continuación se muestran los tipos de marcadores más utilizados:



**Figura 2.10 – Diferentes tipos de marcadores: a) Template; b) ID-marker; c) DataMatrix y d) QR  
(fuente: varias)**

#### **Tracking de características naturales**

Las aplicaciones de visión por computador, incluido el reconocimiento de objetos, recuperación de imagen, odometría visual y RA, se basan, en gran medida, en el establecimiento

de correspondencias entre puntos del mundo real similares que aparecen en las diversas imágenes. El método más aceptado y reconocido para lograr esto se basa en el reconocimiento de puntos claves que comprende dos etapas fundamentales:

- El primer paso corresponde a la detección de puntos clave (keypoints) en una imagen que contiene el objeto de interés.
- El segundo paso corresponde a la extracción de características cuya función es calcular una característica para describir la región que abarca el punto clave. Por lo tanto se pueden encontrar correspondencias comparando las características de una imagen fuente y una imagen de consulta.

Ambos pasos han recibido una considerable cantidad de atención en las últimas décadas, con algoritmos que han sido diseñados para que sean más robustos a las variaciones minimizando el tiempo de procesamiento. En general, los descriptores de características pueden ser categorizados en dos clases:

- Los basados en descriptor
- Los basados en clasificación

*Basados en Descriptor:* se basa en el diseño de la característica sea robusta contra específicos tipos de deformaciones. Por lo general se centran en la escala o tamaño, rotación, corrección y normalización de la intensidad. Algunos enfoques basados en descriptores son: SIFT, GLOH y SURF. Las grandes necesidades computacionales de estos descriptores generalmente limitan su aplicabilidad a las aplicaciones en tiempo real. Los recientes intentos se han hecho para reducir el tiempo de procesamiento: ORB, BRISK, y FREAK.

*Basado en Clasificación:* a diferencia del anterior, requiere una etapa de entrenamiento off-line para aprender un conjunto de características provenientes de imágenes (del objeto de origen) lo que permite lograr insensibilidad a determinados tipos de deformaciones. Estos enfoques están diseñados para transferir gran parte de la carga computacional a la fase de entrenamiento con el fin de reducir el costo de comparación (matching) ejecutado de manera on-line al tiempo que aumenta su robustez.

Es bien sabido que lograr el reconocimiento de objetos invariantes ha sido uno de los mayores desafíos en visión por computador. Recientemente, ha habido un progreso significativo en el uso e implementación de algoritmos hacia la detección de características invariantes en imágenes más complejas.

Como menciona [30] es importante saber que los algoritmos basados en características han sido ampliamente utilizados como detectores de puntos debido a que las esquinas y bordes corresponden a las ubicaciones de imagen con alto contenido de información, lo que significa esto que pueden ser comparados entre las imágenes. Los detectores basados en características son precisos cuando los objetos a ser comparados tienen una esquina o borde distinguibles. En otras palabras, los detectores basados en características tienden a ser más adecuados sobre superficies planas y objetos dentro de una imagen dada. Además, los algoritmos basados en características no se desempeñan tan bien cuando las imágenes están sujetas a variaciones en la escala, la iluminación, la rotación o transformación afín. Para superar estas limitaciones, una nueva clase de algoritmo de comparación de imagen fue desarrollado simultáneamente. Estos algoritmos son conocidos como algoritmos basados en la textura debido a su capacidad para comparar características entre diferentes imágenes a pesar de la presencia de fondos texturizados y bordes bien definidos.

En [22] menciona que los trackers sin marcadores están basados en puntos de características naturales (frecuentemente llamado puntos de interés o puntos clave) visibles en el ambiente del usuario. A fin de obtener una precisión adecuada en la determinación de la pose, dichos puntos de características naturales deben reunir los siguientes requerimientos:

*Procesamiento veloz*

Debe ser posible para calcular grandes cantidades de características de los puntos y los descriptores asociados en tiempo real a fin de calcular la estimación de la pose en un tiempo aceptable.

*Robustez con respecto a los cambios en las condiciones de luminosidad e imagen borrosa*

El conjunto de características de puntos inclusive los descriptores calculados no deben variar significativamente bajo condiciones diferentes de luz o imagen borrosa. Ambos efectos son bastante comunes en particular en ambientes abiertos o exteriores.

*Robustez frente a la observación de diferentes ángulos de visión.*



### *Invariancia en la escala*

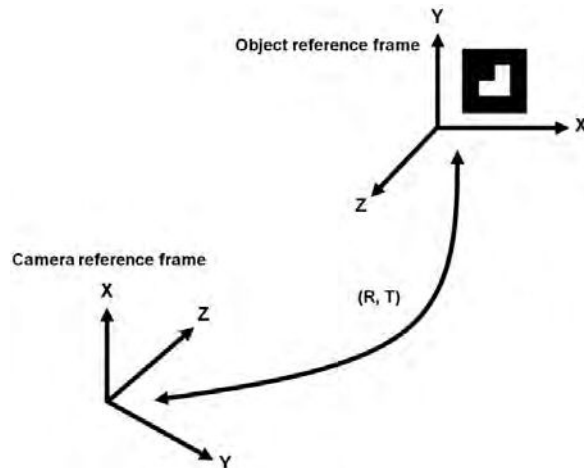
En RA, los objetos que proporcionan los puntos característicos necesarios a menudo se observan desde diferentes distancias. La invariancia de escala se refiere al hecho de que los puntos característicos visibles desde una distancia bastante grande no desaparecerán.

Se debe considerar una buena característica local cuando cumple con las siguientes condiciones:

- *Localidad*: regiones pequeñas son menos sensibles a las deformaciones de la imagen deformaciones, y otras partes del objeto pueden ser ocluidas.
- *Pose invariancia*: El detector características de puntos puede seleccionar una posición canónica, la escala y la orientación para comparación subsecuente.
- *Distinción*: Los descriptores de características deberían permitir una alta tasa de detección (0,7 a 0,9) y una baja tasa de falsos positivos.
- *Repetitividad*: Debemos ser capaces de detectar los mismos puntos a pesar de los cambios en las condiciones de visión

El proceso de tracking en aplicaciones de RA consiste en comparar (matching) puntos 2D-3D y estimar la pose de la cámara. El matching determina una relación entre pares de puntos 2D-3D para establecer una restricción geométrica entre la imagen y los frames de referencia del objeto 3D. Diferentes algoritmos de estimación de la pose se desarrollaron, sin embargo, hay dos clases de algoritmos: analítico e iterativo. Los métodos analíticos son algoritmos que admiten un número finito de soluciones. Mientras que los métodos iterativos se basan en la minimización de un cierto criterio de error. La estimación de la pose de la cámara se basa en la extracción geométrica de primitivas lo que permite hacer el matching de los puntos 2D extraídos de la imagen con los puntos 3D conocidos del objeto. Para determinar la pose, los pares de puntos 2D-3D deben ser conocidos y la cámara se supone calibrada.

La calibración de la cámara determina el modelo geométrico de un objeto y el sistema de formación de la imagen que corresponde [31].



**Figura 2.11– Marco de referencia usado para estimación de la pose**  
**(fuente: Maidi, M. et al [31])**

### 2.6.3 Tracking híbrido

Como menciona [32], los sistemas que utilizaban el tracking híbrido se hicieron disponibles durante la década de los noventa y utilizan por ejemplo compases electromagnéticos (magnetómetros), sensores de inclinación gravitatoria (inclinómetros) y giroscopios (mecánicos y ópticos) para el seguimiento de orientación y el seguimiento de posición ultrasónica, magnética y óptica, respectivamente. La utilización de técnicas de tracking híbridos son actualmente la manera más prometedora de lidiar con las dificultades que plantean los entornos de MRA en interiores y exteriores en general.

Por otro lado en [15] mencionan que en algunas aplicaciones de RA las técnicas de visión por computador no brindan una solución de tracking adecuada y por esta razón se han desarrollado métodos de tracking híbrido que consisten en combinar las salidas de dispositivos físicos y el análisis de video. Este tipo de tracking resulta efectivo para aplicaciones que requieren estimar la posición de la cámara con respecto a una escena estática, pero no se aplica al tracking de objetos en movimiento con una cámara estática.

De acuerdo a [33], las tecnologías de tracking pueden agruparse en tres categorías: *active-target*, *pasive-target* e inercial. Los sistemas de *active-target* incorporan emisores de señal y sensores puestos en un entorno preparado y calibrado. Ejemplos de tales sistemas utilizan señales magnéticas, ópticas, radio y acústicas.

Los sistemas *passive-target* utilizan señales ambientales o de origen natural. Ejemplos incluyen brújulas que detectan el campo de la tierra y sistemas de visión que detectan marcadores fiduciarios intencionalmente colocados (por ejemplo, círculos, cuadrados) o características naturales. Los sistemas *inerciales* son aquellos que detectan fenómenos físicos creados por aceleración lineal y movimiento angular. Los sistemas híbridos intentan compensar las deficiencias de cada tecnología usando mediciones múltiples para producir resultados robustos combinando estas tecnologías con tracking basado en visión.

## 2.7 Reconocimiento de Objetos

De acuerdo a [10] algunos sistemas utilizan objetivos para ayudar en el tracking. Este tipo de tracking se conoce como tracking de landmark. Se conoce la posición Euclidiana de cada objetivo en el ambiente y esta información puede ser utilizada para inferir la posición de la cámara. Esta técnica requiere que estén visible dos o más características del objetivo proporcionando un registro preciso.

El número de características del objetivo depende del número de grados de libertad del punto de vista. El foco de los sistemas es determinar la posición de los objetos en la escena con respecto a la cámara.

Utilizar las características naturales del ambiente en lugar de objetivos elimina la restricción del movimiento de la cámara. Sin embargo, la detección de características naturales normalmente agrega suficiente complejidad computacional. En ambos casos, es decir, sistemas de tracking por características naturales o por objetivo, las características deben ser encontradas antes que sean rastreadas (tracked).

El enfoque basado en objetivos tiene ventajas y desventajas. Una desventaja es que el ambiente observado debe tener un mínimo número de objetivos sin obstrucción. También, la estabilidad de la pose disminuye con menos características visibles. La razón para el uso de las características naturales es eliminar el requisito de colocar objetivos en el medio ambiente. Para reconocer un objetivo es primordial obtener las características principales o atributos más relevantes de un objeto específico y compararlo contra una base de características de los objetos almacenados a fin de determinar la similitud alcanzada.

Como menciona [12] la extracción de puntos característicos es un paso usado frecuentemente para llegar a definir el objeto en sí dentro de una imagen. El objetivo principal es la obtención de los puntos característicos del objeto que nos permitan identificarlo nuevamente a partir de estos puntos.

Para la extracción de puntos característicos, existen una serie de pasos:

- *Detección*: consiste en encontrar el lugar de la imagen donde existen puntos característicos. La detección es el proceso en el cual se localizan las regiones de la imagen que contienen unas características determinadas, debido a su forma o textura. Los retos actuales de los detectores son encontrar regiones características de distintos tamaños y con cualquier orientación. De esta forma se consigue ser invariante a tales efectos, lo cual permite detectar las mismas regiones en escenas donde varía el tamaño de los objetos y donde puedan estar orientados de cualquier forma. Asimismo es importante que la detección de puntos no se vea afectada por cambios de iluminación en la escena.
- *Descripción*: se realiza tras la detección. Una vez localizado el lugar, se procede a describir mediante una o más características la región situada alrededor del punto característico detectado. Estas características se almacenan frecuentemente en un vector para su uso posterior. Este proceso se lleva a cabo mediante el cómputo de la región característica que rodea a cada punto, concepto conocido como *descripción local*.
- *Comparación (Matching)*: para obtener resultados fiables en el proceso de comparación (matching) entre un par de imágenes es la tarea que algunos de los métodos de matching más populares están tratando de lograr. Pero ninguno de ellos ha sido universalmente aceptado. Es de destacar que la selección del método adecuado para completar una tarea de matching depende en gran medida del tipo de imagen y en las variaciones dentro de una imagen.

Algunos descriptores de características se han convertido en populares para el uso de navegación de robots en el interior y exterior de construcciones, tracking de objetos y reconocimiento, y algunas otras tareas que incluyen la comparación y registro de imágenes o

secciones de imágenes. Es importante mencionar que los smartphones también utilizan reconocimiento de objetos en aplicaciones de, por ejemplo, RA, seguridad en basada en reconocimiento facial, e interfaz de usuario basados en reconocimiento gestual. Entre otros métodos y por su precisión se podría utilizar el método SIFT como extractor de puntos de interés de los objetos y los describe de manera robusta contra cambios de traslación, escala y rotación. A este método se lo considera robusto con respecto a cambios geométrico gracias a su detector de región invariante y a su descriptor basado en la distribución de orientación. Sin embargo SIFT consume mucho poder de procesamiento debido a la generación del descriptor y al proceso de matching. La detección e identificación de objetos sigue siendo un desafío en las aplicaciones de visión por computador debido al cambio de condiciones del ambiente.

Adicionalmente hay que considerar al reconocimiento de patrones como técnica para identificar y clasificar un objeto. En [12] se define al Reconocimiento de Patrones como la ciencia que se ocupa de los procesos sobre ingeniería, computación y matemáticas relacionados con objetos físicos o abstractos, con el propósito de extraer información que permita establecer propiedades de entre conjuntos de dichos objetos. Los patrones se obtienen a partir de los procesos de segmentación, extracción de características y descripción, donde cada objeto queda representado por una colección de descriptores. El sistema de reconocimiento debe asignar a cada objeto su categoría o clase (conjunto de entidades que comparten alguna característica que las diferencia del resto). Por lo tanto una definición formal de Reconocimiento de objetos englobado dentro del reconocimiento de Patrones es la siguiente: es la categorización de datos de entrada en clases identificadas, por medio de la extracción de características significativas o atributos de los datos extraídos de un medio ambiente que contiene detalles irrelevantes.

En un problema de clasificación de patrones tradicional, se tienen varias clases que se caracterizan por sus estadísticas. Es decir, se asume que se ha observado en el pasado más de un ejemplo de cada clase y por lo tanto se puede clasificar cada clase por algunas propiedades del conjunto de observaciones pasadas (llamado un conjunto de entrenamiento). Un conjunto de entrenamiento puede estar caracterizado por un modelo paramétrico, en la que todo el conjunto está representado por un pequeño conjunto de parámetros tales como un vector de media y matriz de covarianza.

## **Descriptores de características**

Una de las características del framework propuesto, es reconocer e identificar objetos, tales como construcciones edilicias y vehículos de combate. Para lograr este objetivo una tarea importante es codificar la estructura local en un conjunto de puntos de una imagen con el fin de comparar entre las imágenes a pesar de los cambios visuales tales como cambios de escala, orientación, contraste, etc.

Los descriptores locales para cálculo de puntos de interés han demostrado ser muy exitosos en aplicaciones tales como reconocimiento de objetos, reconocimiento de texturas, ubicación de robots, minería de datos de video, reconocimiento de categorías de objetos, entre otros. Son distintivos, robustos a oclusión y no requieren segmentación. Recientes trabajos se han concentrado en hacer de estos descriptores invariantes a la transformación de imágenes. La idea es detectar áreas de imagen covariantes a una clase de transformaciones, que luego se utilizan como zonas de soporte para calcular descriptores invariantes [34].

Considerando a los detectores de regiones invariantes, la pregunta que resta hacer es cuál descriptor es el más apropiado para distinguir las regiones y si la selección del descriptor depende del detector de región. Hay un gran número de posibles descriptores y medidas de distancia asociados que hacen hincapié en diferentes propiedades de la imagen como intensidades de los píxeles, color, textura, bordes, etc.

Una característica local es un patrón en una imagen que difiere de su vecino inmediato. Está usualmente asociado con un cambio de la propiedad de la imagen o diversas propiedades simultáneamente, aunque no esté necesariamente localizado exactamente en este cambio. Las propiedades de la imagen comúnmente consideradas con la intensidad, color y la textura. Características locales pueden ser puntos, bordes o pequeñas porciones de una imagen. La invariancia en las características locales es una poderosa herramienta que ha sido aplicado exitosamente en un amplio rango de sistemas y aplicaciones [35].

Existen tres clases de categorías de detectores de características:

- 1) un tipo específico de características locales, que puede tener una interpretación semántica específica en el contexto de una determinada aplicación. Por ejemplo, bordes detectados en imágenes aéreas a menudo corresponden a las carreteras; detección de regiones se puede utilizar para identificar impurezas en algunas tareas de inspección;

etc. Estas fueron las primeras aplicaciones para las que han sido propuestos detectores de características locales.

- 2) lo que las características representan realmente no es realmente relevante, siempre y cuando su ubicación puede determinarse con precisión y de manera estable en el tiempo. Esto es, por ejemplo, la situación en la mayoría de las aplicaciones de matching o de tracking, y en especial para la calibración de la cámara o reconstrucción 3D.
- 3) un conjunto de características locales puede ser utilizado como una representación robusta de la imagen, permite reconocer objetos o escenas sin la necesidad de segmentar.

Las 3 categorías anteriores impone sus propias limitaciones, y una buena característica de una aplicación puede ser inútil en el contexto de un problema diferente. Estas categorías pueden ser consideradas cuando se busca un detector de características adecuado para una aplicación determinada. Tradicionalmente, el término detector ha sido utilizado para referirse a la herramienta que extrae las características de una imagen, por ejemplo, una esquina, una región o un detector de bordes. La evaluación de los descriptores se ejecuta en el contexto de matching y reconocimiento de misma escena u objeto observado bajo diferentes condiciones de vista.

Muchas características son seleccionadas manualmente por el usuario dependiendo del dominio de aplicación. Sin embargo, el problema de la selección automática de características ha recibido atención significativa en la comunidad de reconocimiento de patrones. Los métodos de selección automática de características pueden dividirse en métodos de filtro y métodos de contenedor. El método de filtro intenta seleccionar las características basadas en un criterio general, por ejemplo, las características deben ser no correlacionadas [26].

Los métodos de contenedor seleccionan las características basadas en la utilidad de las características en un dominio específico del problema, por ejemplo, la performance en la clasificación utilizando un subconjunto de características.

PCA es un ejemplo de métodos de filtro para la reducción de características. PCA implica la transformación de un número de variables correlacionadas en un número (pequeño) de variables no correlacionadas llamada componentes principales. El primer componente

principal representa la mayor cantidad de variabilidad en los datos como sea posible y cada componente subsiguiente representa como gran parte de la variabilidad restante como sea posible.

Un método de contenedor selecciona las características de manera discriminatoria para seguir (tracking) una clase particular de objetos es el caso del algoritmo Adaboost.

Adaboost es un método para encontrar un clasificador fuerte basado en una combinación de clasificadores débiles moderadamente inexactos. Teniendo en cuenta un amplio conjunto de características, un clasificador puede ser entrenado para cada característica. Adaboost, descubrirá una combinación ponderada de los clasificadores (características que representan) que maximizan el rendimiento de la clasificación del algoritmo.

## **2.8 Arquitectura de software para aplicaciones de RA**

De acuerdo a [17] un sistema de RA requiere de un conjunto de componentes para cumplimentar con sus objetivos básicos: el reconocimiento en tiempo real del contexto del mundo físico, la registración de los objetos físicos de destino con sus objetos virtuales correspondientes, visualización del contenido de RA y el manejo de la interacción del usuario. También, hay que considerar que la arquitectura global de una aplicación de RA es responsable por:

- Procesar el contenido incluyendo contenido adicional multimedia
- Procesar las entradas de los usuarios
- Procesar el contexto provisto por los sensores
- Administrar la presentación del resultado final (auditivo, visual, háptico, etc)
- Administrar la comunicación con los servicios adicionales

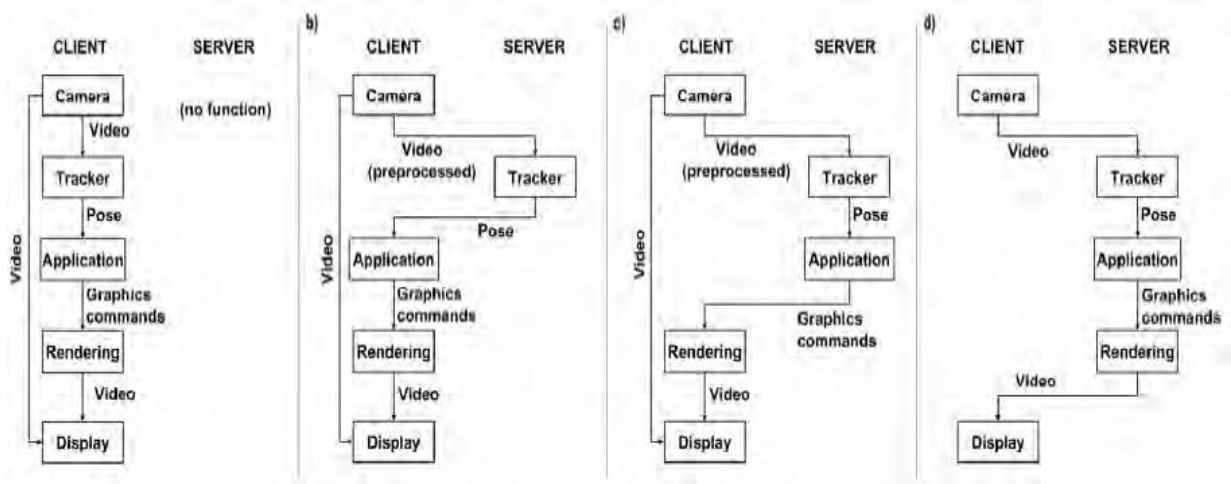
En cuanto a los procesos o tareas centrales que debe realizar una aplicación de RA se pueden ubicar la carga principal del lado del cliente, del lado del servidor o una distribución de carga equilibrada. Si consideramos una aplicación típica de RA que utiliza una sola fuente



de vídeo, tanto para el tracking como para la proyección visual, el “pipeline” de procesamiento se debería componer de las siguientes tareas principales: adquisición de vídeo, tracking, cómputo de aplicaciones, renderizado y visualización. La descarga de algunas de estas tareas a un servidor permite ejecutar aplicaciones distribuidas pero para ello se debe tener cautela en el uso del ancho de banda de red disponible. En [14] se detalla que existen 4 niveles diferentes de externalizar las tareas al servidor:

1. Toda la carga es procesada por el cliente independizándose del servidor y de la infraestructura que lo soporta.
2. Se externaliza al servidor únicamente la tarea de tracking. En este caso se enviará una imagen pre-procesada o un video comprimido para calcular la posición y orientación correspondiente. La ventaja de este tipo de externalización es que la tarea computacional es cargada al servidor mientras que los otros detalles de la aplicación son manejados por el cliente. Las dependencias entre el cliente y el servidor son mínimas. Por ejemplo, mientras que el tracking de un marcador puede ser ejecutado en tiempo real desde el cliente a través de un dispositivo móvil, el tracking de características naturales se puede beneficiar si se ejecuta desde el servidor por su gran capacidad de cómputo. No obstante los procesadores de los nuevos dispositivos móviles son muy poderosos y esta tendencia podría cambiar drásticamente.
3. Se externaliza al servidor las tareas de tracking y de la aplicación. En este caso la aplicación envía al cliente un flujo de comandos gráficos o de información gráfica para que sea, posteriormente, renderizado el objeto virtual en el lado del cliente
4. Las tareas de tracking, el cómputo de la aplicación y el renderizado es enviado al servidor. Estas 3 tareas resultan las de mayor procesamiento de cómputo y el objetivo es enviar toda la carga al servidor.

En la siguiente figura se grafica los cuatro niveles de tareas para el procesamiento de RA entre el cliente y el servidor:



**Figura 2.12 – Diferentes niveles de procesamiento de las tareas de RA. a) las tareas se ejecutan en el cliente; b) en el server se ejecuta la tarea de tracking; c) en el servidor se ejecutan las tareas de tracking y la lógica de la aplicación y d) todas las tareas se realizan en el servidor  
(fuente: Wagner, D. [14])**

# Capítulo 3

## Realidad Aumentada en proyectos militares

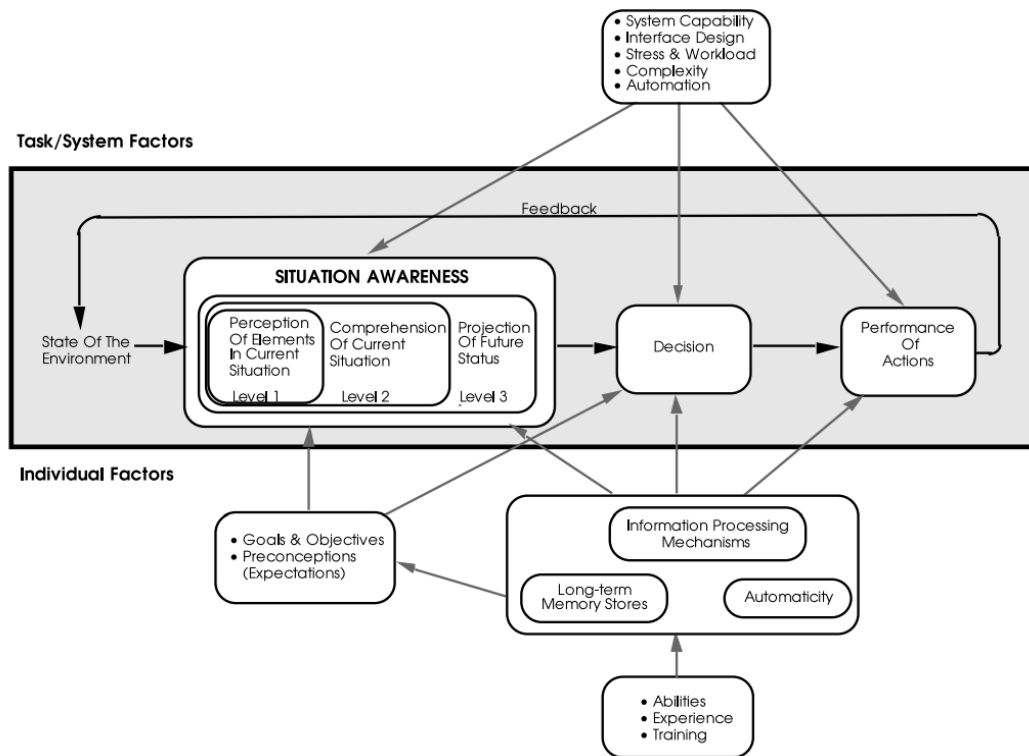
En este Capítulo se detallan algunos proyectos militares que utilizan la RA como tecnología principal para mejorar la CS. Se explica que es la CS y cómo impacta la RA en la CS. Puntualmente se estudian diferentes tecnologías militares que aplican RA para mejorar la CS. Se realiza una tabla comparativa entre los diferentes proyectos militares que utilizan RA contrastada con la tecnología propuesta (framework de software y los componentes tecnológicos que le dan soporte) en el informe.

### 3.1 Realidad Aumentada para mejorar la Conciencia Situacional

Brown, David Wm [1] menciona que la CS (en inglés, situation awareness o también situational awareness) se refiere a la percepción, la comprensión, y la previsión de los elementos dentro de un entorno operacional requerido para actuar con eficacia dentro de ese ambiente.

Tremblay, Sébastien et al [36] definen que la CS es un requisito previo para la oportuna y correcta toma de decisiones en el rápido y altamente estresante contexto de los entornos operativos de infantería. Se espera que la introducción de las tecnologías de soporte electrónico en el campo de batalla mejore la CS, proporcionando la información correcta, en el momento adecuado y en el formato correcto.

En la siguiente figura se grafica el modelo de la CS en la toma de decisión dinámica.



**Figura 3.1– Modelo de la CS en la toma de decisión dinámica  
(fuente: Masoud, G. et al [37])**

Endsley, M. R. et al [38] determinan que uno de los factores más importantes que subyacen en el desarrollo de una adecuada CS es la presencia de los modelos mentales y esquemas de situaciones prototípicas. Proporcionan una construcción mental fundamental para dirigir la forma de asignar la atención y destacar los temas críticos.

Los principales atributos de la CS son:

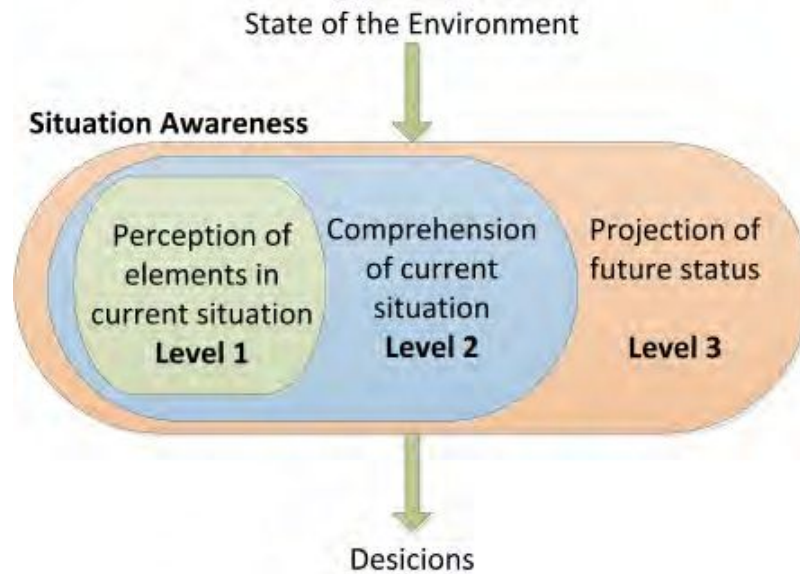
- Visión global en todo momento con la posibilidad de obtener detalles en cualquier momento
- El uso de interfaz estándar para el acceso y el intercambio de datos
- Los datos organizados para alcanzar los objetivos
- Fácil transición entre los objetivos

Endsley, M. R [39] [40] menciona que la CS es la percepción de los elementos en el medio ambiente dentro de un volumen de tiempo y espacio, la comprensión de su significado

y la proyección de su situación en un futuro próximo. Es importante detallar los tres niveles de la CS propuestos. El modelo sugerido se estructura en tres niveles jerárquicos de la CS, cada uno de los cuales es necesario, pero no suficiente, precursor del nivel siguiente y superior. El mismo se basa en el establecimiento de una cadena de procesamiento de la información que, dividida en tres niveles, comienza con la percepción de la misma, prosigue con la interpretación y finaliza con la predicción.

Los procesos están formados por tres niveles aplicables a la CS, estos son:

- Nivel 1: **percepción de los elementos del entorno**. Se refiere a la percepción de la información recibida a través de la instrumentación disponible, del comportamiento del entorno, de la información asociada a otros sujetos del entorno, a amenazas circundantes, en definitiva, al entendimiento del ambiente. En este estadio no se interpretan los datos, sólo se almacenan en forma “bruta”, sin elaborar. Algún dato puede servir para confirmar el estado de una variable, pero sin ser integrado todavía por el individuo.
- Nivel 2: **comprensión de la situación actual**. Se alcanza a través de la percepción de los elementos del entorno. En este estadio no es necesario que se produzca una comprensión relevante para el desarrollo de su tarea por parte del operador. Es el nivel en el que se produce el proceso de integración de los datos. La CS va más allá de la percepción, contemplando también el proceso mediante el cual los sujetos combinan, interpretan, almacenan y retienen información. De esta manera la CS incluye no solo la percepción de la información del entorno, sino también la integración de múltiples piezas de información y la determinación de su relevancia en función de los objetivos del sujeto.
- Nivel 3: **predicción del estatus futuro**. Este es el nivel más elevado de la CS, que se asocia con la habilidad para proyectar el estatus futuro de los elementos del entorno. La precisión en la predicción depende enormemente de la información obtenida en los niveles 1 y 2. La anticipación proporciona al sujeto el tiempo suficiente para resolver conflictos y planificar su actuación para alcanzar sus objetivos.



**Figura 3.2 – Modelo de los tres niveles de la CS**  
**(fuente: Endsley, M. R. [39] [40])**

Por otro lado, los problemas más frecuentes asociados con la mala CS están dados por la ausencia en la recolección de la información necesaria, los problemas de percibir la información clave en la situación y la falta de comunicación. Las soluciones de CS permiten a los soldados hacer un uso efectivo de la información variada en un contexto de batalla.

De acuerdo a [41], la nueva tecnología ofrece métodos innovadores de obtener información contextual y representarla visualmente de una manera natural y no invasiva sin afectar el proceso cognitivo del combatiente.

En [42] el concepto de Soldado del Futuro fue desarrollado para identificar las capacidades de un soldado y llevarlas al campo de batalla. Es un concepto de cómo podría estar equipado el Soldado del Futuro. El Soldado del Futuro estará adaptado a cada área tecnológica teniendo especial énfasis en el rendimiento cognitivo para mejorar la eficacia del soldado y un aumento en el ritmo operativo.

Hay siete áreas principales aplicable al soldado:

- Rendimiento Humano y Entrenamiento
- Protección
- Letalidad
- Movilidad y Logística
- Redes

- Sensores
- Potencia y energía

Es importante resaltar el área de Rendimiento Humano ya que está relacionado con el desempeño de los soldados en el campo de batalla. Tener un correcto desempeño en una operación militar depende en gran medida en tener una correcta CS. La CS incluye la pérdida o percepción incompleta o cambio a elementos presentes en el entorno operacional del soldado. Este factor puede estar relacionado con las percepciones limitadas de un individuo de su entorno operacional así como su capacidad de mantener un nivel alto de vigilancia de una serie de elementos.

Las futuras operaciones militares se apoyarán en herramientas de Comando, Control, Comunicaciones, Computación, Información, Inteligencia (C4I2) para un rendimiento óptimo en sus tareas asignadas en ambientes versátiles y hostiles. Los soldados portarán diminutas y potentes unidades centrales de procesamiento donde procesarán toda la información, capaz de controlar desde los parámetros fisiológicos, pasando por la gestión de las comunicaciones hasta la navegación o los datos de los subsistemas de armas y demás sensores asociados. En [43] se menciona que las soluciones críticas están centradas en mejorar:

- Comunicación
- Conciencia Situacional (CS)
- Comando y Control (C2)

Los ambientes virtuales y aumentados estarán en todas partes (ubicuidad) y apoyarán casi todas las facetas de la guerra incluyendo comunicaciones, visualización de datos, el control del sistema y capacitación.

El incremento en el uso de sensores de vehículos tanto en vehículos no tripulados como en lugares estratégicos se combinan para aumentar los requisitos de información a fin de proveer a los soldados datos que les sirva para cumplimentar su misión en el campo de batalla. Las unidades de combate se vuelven más dispersas y especializadas por lo que adquirir y mantener la Conciencia Situacional se vuelve más difícil. Por ejemplo, el aspecto predictivo de la Conciencia Situacional se hace especialmente difícil en las operaciones urbanas, donde la línea de contacto visual con las fuerzas aliadas o amigas se hace difícil mantenerla durante un largo período de tiempo. En principio, algunas de estas dificultades se

pueden superar mediante el uso de una tecnología que puede organizar y presentar la información al usuario de forma automática. Un enfoque prometedor es la RA [44].

Por otro lado se puede mejorar la CS por medio del envío de información a través de los sensores dispersos geográficamente para luego visualizar dicha información utilizando como técnica la RA [45].

Es importante entender que la CS juega un papel crítico en las operaciones militares en cualquier terreno donde se desarrolle una misión. El comando y control dependen en gran medida de la CS, ya que las decisiones tienen que hacerse sobre la base de la situación actual.

La literatura apoya el potencial de la RA para mejorar la CS. Con el uso de la RA podría producir mejoras dramáticas en el rendimiento del soldado y proporcionar una gran ventaja en el combate.

## **3.2 Principales proyectos militares que aplican la RA**

En lo que respecta al ámbito militar, recientemente se está utilizando la RA como soporte para mejorar la CS en la toma de decisiones en particular en operaciones militares. Muchas de las operaciones militares se desarrollan en entornos desconocidos. Estos complejos campos de batalla en 3D son muy exigentes e introducen muchos desafíos para el combatiente. Estos incluyen visibilidad limitada, falta de familiaridad con el medio ambiente, amenazas de francotirador, ocultamiento de fuerzas enemigas, mala comunicación y un problema de la localización e identificación de los enemigos y de las fuerzas aliadas. Para ello tener una conciencia situacional amplia del terreno es vital para que la operación sea un éxito minimizando los efectos colaterales.

Livingston [5] [44] pasó a especificar diez observaciones importantes para el desarrollo de un exitoso sistema digital:

- No existen modelos maduros para la comunicación en ambientes urbanos
- Ningún modelo coherentemente integra la información del terreno, el clima, los océanos y el espacio para las operaciones urbanas conjuntas.
- Las unidades necesitan poder comunicarse entre sí y con unidades cercanas



- El seguimiento (tracking) de fuerzas aliadas en ambientes urbanos es importante
- Personalizar el cuadro de operaciones común (en inglés, common operating picture –COP-)
- La obtención de información debe ser automatizado de acuerdo a las prioridades y los roles del individuo.
- Las unidades deben actualizar los mapas y las bases de datos
- Existe la necesidad de información en 3D dentro de los edificios y bajo tierra
- La información debe estar rotulada
- Ayudar al usuario encontrar objetos de interés dentro del desorden visual

Una serie de programas de investigación han explorado los medios por los cuales la información de navegación y coordinación se pueden enviar al soldado utilizando técnicas de RA. Los programas del Soldado del Futuro han incursionado en la investigación y el desarrollo de componentes electrónicos aplicando la RA como tecnología para visualizar información del medio ambiente donde se produce el combate y de esta manera facilitar la toma de decisión ante situaciones hostiles.

### **3.2.1 Proyecto Eyekon**

Hicks, Jeffrey et al [46] definen al proyecto EyeKon como un sistema de soporte a la toma de decisiones basado en agentes inteligentes instalados en una computadora portátil que transporta el soldado. El soldado visualiza información de objetivos y otra información en su armamento. El soldado no sólo ve objetos y posiciones sino también los medios para designar nuevos objetos como posibles amenazas y coordinar las acciones del equipo en respuesta a una situación que cambia dinámicamente. El proyecto tiene como objetivo desarrollar iconos inteligentes y descripciones que se superponen en el video del arma del soldado.

Los sistemas de armas deben ser optimizados no sólo para reducir al mínimo el peso total transportado por el soldado, sino también para reducir al mínimo la sobrecarga cognitiva. Demasiados datos en su pantalla y el soldado rápidamente se siente abrumado con demasiada información afectando su conciencia situacional del medio que lo rodea.

Las funciones básicas se encuentran en una computadora portátil conectada, a través de una red inalámbrica segura, a una base de datos local y remota. Incorpora sensores que brindan información en tiempo real (ej. Inercial, GPS, IR, etc). El software está compuesto por agentes que realizan consultas a la red a fin de monitorear información de amenazas y otros tipo de información estratégica para el soldado (ej. misión, estados, etc). Sobre la visión del arma se superimprime información utilizando técnicas de RA o a través del uso de una gafa monocular.

EyeKon implica el diseño de una interfaz avanzada de usuario y es un subconjunto de RA de mejoras visuales aplicadas al concepto de Land Warrior.

Las características conceptuales son:

- Superponer iconos de situación en el armamento de un soldado
- Mejorar la conciencia situacional analizando el rango de amenazas
- Maximizar la capacidad de respuesta y minimizar los requisitos de peso y potencia

### **3.2.2 Proyecto BARS**

El Naval Research Laboratory (NRL) desarrolló un sistema prototipo de realidad aumentada conocido como BARS (en inglés, Battlefield Augmented Reality System) [2]. Este sistema conecta a múltiples usuarios móviles junto con un centro de mando. BARS se centró en el desarrollo de un sistema digital para ayudar a resolver el creciente énfasis en las operaciones militares en terreno urbano. BARS realiza el seguimiento (tracking) de la posición y orientación de la cabeza del usuario y superpone gráficos junto con anotaciones que se alinean con los objetos reales en el campo visual del usuario. Varias unidades compartían una base de datos común, donde las personas podían optar por unirse a un canal determinado para acceder a los datos gráficos.

Una característica importante es que se utilizaron componentes comerciales con el objetivo de acceder a las mejoras o actualización de hardware apenas estuviesen disponibles en el mercado.

De acuerdo a [47] el proyecto se centró en la investigación de los aspectos relativos al Factor Humano e Interfaz de Usuario, haciendo principal énfasis en las siguientes características:

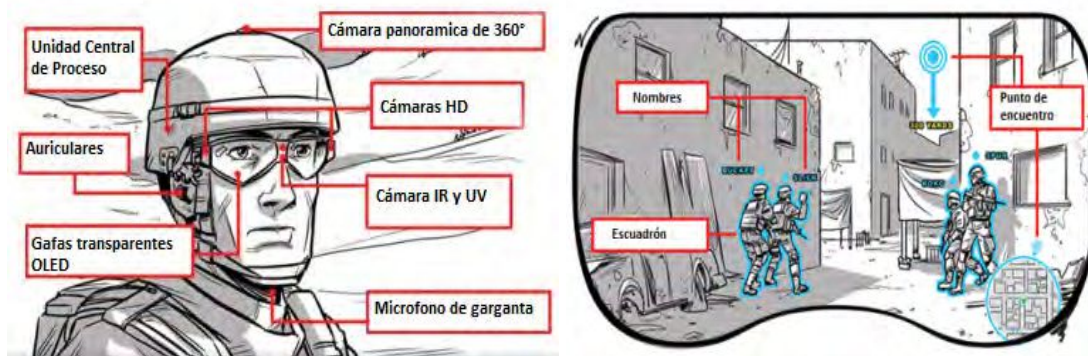
- Visión “rayos X” y percepción de la profundidad: una ventaja del proyecto era saber dónde se encontraban las tropas distribuidas en un terreno urbano; como también establecer los puntos de interés en el entorno de los miembros del equipo. Ambos objetivos requiere que el sistema de Realidad Aumentada identifique objetos que se encuentran ocultos a la visión del usuario. Para lograr se utilizó como principio la metáfora “visión de rayos X de superman”. La investigación se centró en el estudio de la profundidad de objetos gráficos.
- Percepción Básica: con el objetivo de mejorar la visión que se obtenía por medio del video provisto por las gafas, se investigó la posibilidad de perfeccionar la percepción del entorno. Para ello se analizaron aspectos como sensibilidad al contraste, percepción de los colores y percepción circundante.
- Filtrado de la Información: la sobre carga de información puede resultar un factor negativo a la hora de tomar una decisión. La cantidad de información puede llegar a ser demasiada para procesar en el ritmo dinámico de las operaciones militares, hasta el punto en que inhibe la capacidad del personal militar para completar las tareas asignadas. Se han desarrollado algoritmos para restringir la información que se muestra a los usuarios. El algoritmo de filtrado evolucionó a partir de un filtro basado en regiones a uno híbrido que se centra en el modelo de interacción espacial, un filtro basado en reglas y el concepto militar de un área de operaciones.
- Selección de Objetos: Para consultar, manipular, o actuar sobre los objetos, el usuario debe primero seleccionar estos objetos. El proyecto BARS permite a un usuario seleccionar objetos mediante la combinación de visión (utilizando el tracking de la cabeza) y señalización relativa dentro del campo visual utilizando tracking ocular o mouse 2D o 3D. La compleja naturaleza de la operación de selección hace que sea susceptible a errores. Para mitigar estos errores, se diseñó un

algoritmo de selección probabilística multimodal (voz y gesto). Este algoritmo incorpora una jerarquía de objetos, diversos algoritmos de reconocimiento gestual y reconocimiento por voz.

- **Colaboración:** El sistema incluye una base de datos de información, que puede ser actualizado por otros usuarios. El intercambio de información a través del área de operación es un componente crítico del equipo para mejorar la Conciencia Situacional. Se diseñó un sistema de distribución de información de modo que estas actualizaciones se envían a través de la red de acuerdo con un esquema de prioridades. Por otro lado, se utilizó Google Earth como API ya que ofrecía una adecuada visualización de edificios en 3D y, adicionalmente, permitió la rápida creación de prototipos.
- **Entrenamiento:** el entrenamiento requiere que los participantes se desempeñen en terrenos urbanos reales. El objetivo es realizar los entrenamientos en sitios reales con la superposición de objetos sintéticos (edificios, armamento, enemigos, etc). Un desafío en el entrenamiento utilizando Realidad Aumentada es que las fuerzas simuladas tienen que aparecer en la pantalla del usuario para dar la ilusión de que existen en el mundo real.

### **3.2.3 Proyecto iARM**

En el año 2009, el Defense Advanced Research Projects Agency (DARPA) contrató a la empresa Tanagram Partners para desarrollar el proyecto Intelligent Augmented Reality Model (iARM) [48]. Básicamente, el objetivo de iARM era desarrollar un sistema digital integrado que podría mejorar significativamente la toma de decisiones del personal militar en entornos complejos a través de un sistema operativo integrado, un modelo de servicios de datos, y un HMD mejorado. El objetivo era que todos estos componentes trabajen juntos de una manera transparente permitiendo a los soldados percibir, comprender y, lo más importante, proyectar el mejor curso de acción para un mayor rendimiento para alcanzar los objetivos tácticos. El proyecto iARM abarcaba muchos de los atributos de la inteligencia artificial. En la siguiente figura se muestra el diseño conceptual del HMD y la visión del soldado a través de las gafas.



**Figura 3.3- Diseño conceptual del proyecto iARM**  
(fuente: Juhnke, J. et al [48])

### 3.2.4 Proyecto ULTRA-Vis

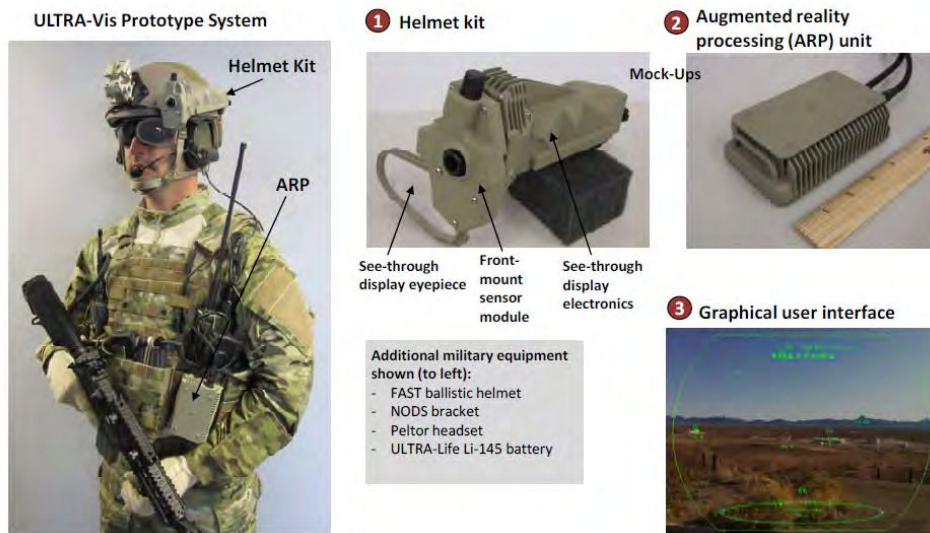
En [49] se detalla que el programa Urban Leader Tactical Response, Awareness and Visualization (ULTRA-Vis), soportado por el DARPA en su fase 1, ha desarrollado un prototipo de un sistema de RA para los soldados en el campo de batalla. El sistema ULTRA-Vis superpone iconografía gráfica a todo color en la escena local observada por el soldado. El programa desarrolló e integró un sistema de poco peso, una pantalla *see-through* holográfica de bajo consumo con un sistema de visión de tracking de posición y orientación. Usando el sistema ULTRA-Vis, un soldado puede visualizar la ubicación de otras fuerzas, los vehículos, los peligros y las aeronaves en el medio ambiente local, incluso cuando éstos no son visibles para el soldado. Además, el sistema puede ser utilizado para comunicar al soldado de una variedad de información tácticamente significativa incluyendo imágenes, rutas de navegación y alertas. El prototipo estará dotado para el reconocimiento gestual mediante el uso de un guante. Permitirá sobreimprimir símbolos en el campo de batalla en 3D, localizar objetivos enemigos y ubicar a las fuerzas aliadas. ULTRA-Vis provee a los escuadrones una ventaja táctica muy clara permitiendo la colaboración entre los integrantes del escuadrón. Posibilitará una alta conciencia situacional y la habilidad de tomar decisiones mientras se está en movimiento en el campo de operaciones.

En cuanto al equipamiento que forma parte del proyecto se destaca:

- Casco de combate con tecnología *optical see-through* con pantalla color
- Un unidad de procesamiento con arquitectura ARM

El sistema ULTRA-Vis incorpora un peso adicional adecuado al conjunto de componentes militares que lleva el combatiente. El prototipo procesa información obtenida de la red militar de datos obtenida por radio y superpone información georegistrada por medio de iconos sobre el campo de visión del usuario.

En la siguiente figura se detalla los componentes del prototipo.



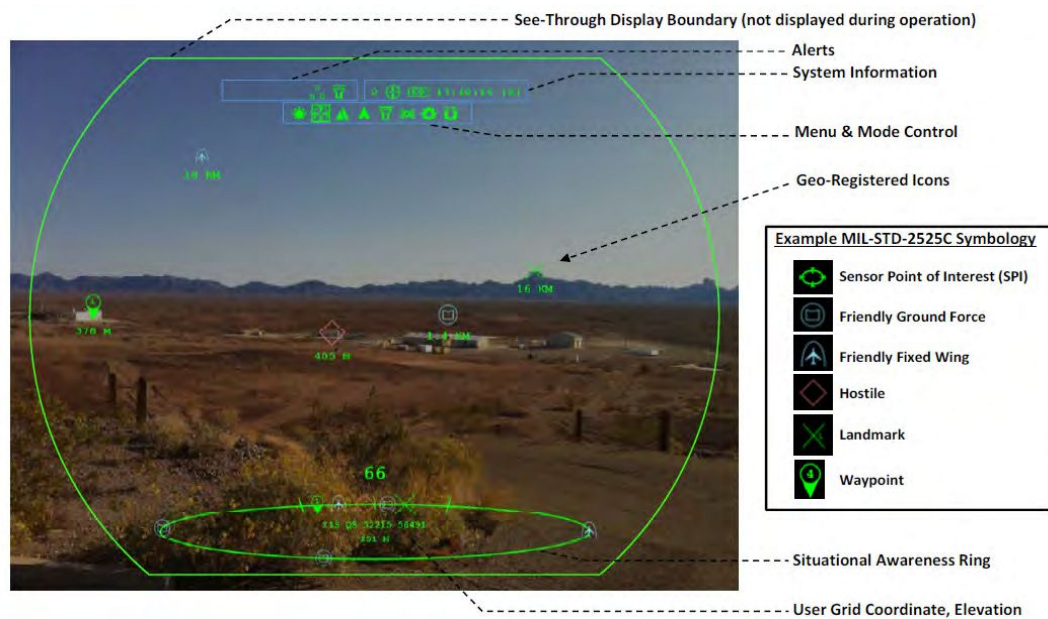
**Figura 3.4 – Componentes del prototipo ULTRA-Vis**  
(fuente: Roberts, R. et al [50])

De acuerdo a [50] ULTRA-Vis posee una avanzada interfaz de usuario, que consta de las siguientes características:

- Alertas operacionales: informa al soldado sobre la inoperatividad de la radio; disturbios magnéticos que afectan a los componentes; GPS dañado, entre otros.
- Sistema de información de estado de los componentes: informa sobre el nivel de batería de la unidad de procesamiento; tiempo operacional; tiempo regional. El sistema de información es activado a demanda del usuario por medio de un botón que se encuentra en el casco.
- Acceso a los menús del sistema: los usuarios pueden acceder a los diferentes menús a fin de cambiar, por ejemplo, configuraciones del sistema.

- Visualización de iconos de puntos de interés geo-referenciados: los iconos o simbología militar son representados gráficamente dentro del campo de visión de la pantalla del usuario
- Anillo de Conciencia Situacional: es una herramienta gráfica intuitiva que permite al soldado tener un entendimiento en tiempo real de los aspectos circundantes en 360°. Estos aspectos están relacionados con el conocimiento de la ubicación de fuerzas aliadas, fuerzas enemigas y la ubicación de otros puntos de interés. Los iconos que se encuentran alrededor del anillo corresponden a la ubicación del soldado y la rotación del mismo.

En el siguiente se encuentra representada la interfaz de usuario del proyecto ULTRA-Vis



**Figura 3.5 – Interfaz de usuario del prototipo ULTRA-Vis  
(fuente: Roberts, R. et al [50])**

La evolución del proyecto ULTRA-Vis se denomina *ARC4*. Este proyecto, desarrollado por la empresa ARA<sup>1</sup>, se caracteriza por mostrar íconos geo-registrados e información

<sup>1</sup> ARA: Applied Research Associates, Inc. <http://www.ara.com/>



estratégica superponiéndose en el campo de visión del usuario. Se trata de un sistema de software que toma información provista por sensores externos, pudiéndose visualizar a través del HMD (Head-Mounted Display) del soldado. Una ventaja del software es que permite ejecutarse sobre dispositivos que soporten visión nocturna.



**Figura 3.6 – Información provista y componentes de hardware del proyecto ACR4 (fuente: ARA)**

Las características principales del software ACR4 son las siguientes:

- Visualización agnóstica: permite *renderizar* información en cualquier dispositivo móvil de visión transparente –incluyendo dispositivos de visión nocturna–
- Head Tracking: implementa algoritmos de alta precisión para calcular la pose del usuario en ambientes externos



- Visualización a través de visores (Heads-up): toda la información como iconos, anotaciones, etc., son visualizadas por medio de visores especiales.
- Coordinación: el software permite proyectar, entre los usuarios, la información como los iconos, las anotaciones y los mensajes en el campo de visión correspondiente.

### 3.3 Comparativa de proyectos militares que utilizan RA

Luego de haber analizado las tecnologías digitales que utilizan a la RA como soporte tecnológico para mejorar la CS en operaciones militares se construyó una tabla comparativa a fin de determinar la necesidad concreta de construir un framework propio de RA que sea aplicado en la construcción de aplicaciones y que sean ejecutables en tecnologías digitales provistas por el Ejército Argentino. Cabe destacar que los proyectos que se encuentran en las tablas comparativas forman parte de tecnologías integradas, muchas de ellas son parte de programas de Soldado del Futuro explicado en la sección *¡Error! No se encuentra el origen de la referencia. ¡Error! No se encuentra el origen de la referencia.* El uso de RA para mejorar la CS, en alguno de los proyectos, es solo una pieza importante del conjunto.

A continuación se confeccionó una lista de elementos que hacen a la CS. Estos elementos deben ser tratados para poder tener un mejor control y entendimiento del contexto del campo de operaciones con el objetivo de mejorar la CS en operaciones militares.

Estos elementos son los siguientes:

- *Reconocimiento del lugar de operaciones*: es de vital importancia tener una referencia precisa del lugar de operaciones. Conocer el terreno de operaciones aumenta la confianza y la percepción del entorno.
- *Identificación de infraestructura*: en un terreno hostil y poco conocido es de importancia identificar infraestructura con el objeto de utilizarlo ya sea como referencia de ubicación o para ejecutar una misión determinada.

- *Identificación de elementos del ambiente geográfico*: además de la identificación de infraestructura es importante obtener información contextual adicional para ello se debe identificar objetos, tales como referencias geográficas, piezas de artillería, u otro.
- *Alerta de amenazas mediante el uso de simbología descriptiva*: comprender el ambiente y sus amenazas circundantes le permite al operador tomar decisiones en el momento justo. Para ello se debe contar con una simbología militar, clara y precisa de las amenazas y alertas dentro de un área de acción establecida.
- *Localización de miembros aliados*: conocer la ubicación del resto de los operadores es de vital importancia desde un punto de vista estratégico y de seguridad para el conjunto de operadores
- *Tracking del trayecto de operaciones*: como método de reconocimiento del terreno y de puntos de chequeo operativos una opción es marcar en un mapa digital el trayecto que realizaron los operadores.
- *Comunicación entre el mando superior y el JPat*: la comunicación entre el mando central y el JPat forma parte de las tareas de comando y control. La comunicación debe hecha directamente con el jefe del grupo ya que será quién tomará las decisiones operativas principales.
- *Prioridad (filtrado) de información*: para no abrumar la capacidad de procesar la información contextual a la que debe someterse el operador, debe existir una primer capa de información, tal como, el radar e información contextual como las alertas. Las capas restantes deben visualizarse a pedido del operador a través de la introducción de gestos o de un comando vocal.

En la siguiente Tabla se compara las funcionalidades implementadas en el resto de los frameworks militares:

Tabla 3.1 Elementos imprescindibles para mantener un alto nivel de CS

<b>Elementos de la Conciencia Situacional (requerimientos funcionales)</b>	<b>EyeKon</b>	<b>BARS</b>	<b>iARM</b>	<b>ULTRA-Vis</b>	<b>RAIOM</b>
Reconocimiento del lugar de operaciones	X	X	X	X	X
Identificación de infraestructura			X	X	X
Identificación de elementos del ambiente geográfico	X		X	X	X
Alerta de amenazas mediante el uso de simbología descriptiva	X		X	X	X
Localización de miembros aliados			X	X	X
Tracking del trayecto de operaciones					X
Comunicación entre miembros y superiores		X	X	X	X
Prioridad (filtrado) de información	X	X	X	X	X

A continuación se listan la tecnología que deben utilizarse para dar soporte técnico a los elementos que se detallaron precedentemente:

Tabla 3.2 Tecnologías utilizadas como soporte a la CS

<b>Tecnología utilizada como soporte a la CS</b>	<b>EyeKon</b>	<b>BARS</b>	<b>iARM</b>	<b>ULTRA-Vis</b>	<b>RAIOM</b>
Visualización mediante RA	X		X	X	X
Dispositivos liviano					X
Procesamiento distribuido					X
Sensores (GPS, aceler., inerciales, etc)	X		X	X	X

Reconocimiento vocal		X	X	X	X
Reconocimiento gestual		X	X		X
Reconocimiento facial					X
Reconocimiento de objetivos			X	X	X
Reconocimiento del lugar				X	X
Comunicación colaborativa		X	X	X	X
Software libre					X
Seguridad	X			X	X
Uso dual		X		X	X
Componentes COTS		X	X		X

Analizando las dos tablas comparativas precedentes, es decir, por un lado los elementos necesarios que forman parte de tener una CS amplia para tener el estado de situación de las amenazas circundantes y, por el otro lado, las tecnologías que dan soporte para que se cumpla lo antes detallado, se desprende que no todas las tecnologías cumplen con los requerimientos operacionales. Antes esta situación se optó por desarrollar un framework de RA, denominado RAIOM (Realidad Aumentada para la Identificación de Amenazas y Objetivos Militares) centrado en resolver la problemática de la CS con el objetivo de poder desarrollar aplicaciones de RA que se ejecuten en una arquitectura de sistemas acotada y sugerida en el presente informe.

# Capítulo 4

## Arquitectura y diseño de Software

En este Capítulo se detallan la arquitectura de software propuesta. La importancia de diseñar un framework de uso específico. Adicionalmente se menciona la arquitectura en capas y componentes propuesto como también el ambiente de desarrollo.

### 4.1 Framework y su importancia

La palabra inglesa "framework" (marco de trabajo, en español) define, en términos generales, un conjunto estandarizado de conceptos, prácticas y criterios para enfocar un tipo de problemática particular que sirve como referencia, para enfrentar y resolver nuevos problemas de índole similar<sup>2</sup>.

Al centrarnos en las definiciones desde la perspectiva de desarrollo de software se puede afirmar que un framework es un diseño a gran escala que describe cómo una aplicación de software se descompone en un conjunto de objetos que interactúan y se relacionan. Generalmente está representado por un conjunto de clases abstractas cuyas instancias (objetos) interactúan entre sí. Una característica importante de un framework es el diseño de componentes que lo soporta. Adicionalmente se puede reutilizar la implementación pero quedaría en un segundo plano ya que lo destacable de un framework es la reutilización de las funcionalidades y de los componentes que lo conforman.

Los frameworks se centran en respetar las características de los lenguajes orientados a objetos, es decir, abstracción de datos, polimorfismo y herencia. En este caso una clase abstracta representa una interfaz detrás de la cual las implementaciones pueden cambiar; el polimorfismo se refiere a la propiedad por la que es posible enviar mensajes sintácticamente iguales a objetos de tipos distintos mientras que la herencia es el mecanismo más utilizado para alcanzar la reutilización y la extensibilidad. Por medio de la herencia se pueden crear

---

<sup>2</sup> <http://es.wikipedia.org/wiki/Framework>

nuevas clases partiendo de una clase o de una jerarquía de clases preexistente evitando con ello el rediseño, la modificación y verificación de la parte ya implementada. La herencia facilita la creación de objetos a partir de otros ya existentes e implica que una subclase obtiene todo el comportamiento (métodos) y eventualmente los atributos (variables) de su superclase.

Un framework describe la arquitectura de un sistema orientado a objetos; las clases de objetos y cómo interactúan entre ellos. Describe como un software se descompone en objetos. Los frameworks se encuentran representados por un conjunto de clases, usualmente abstractas [51]. Adicionalmente [52] pone énfasis en definir a un framework como un conjunto de clases abstractas, especificando a una clase abstracta como una clase que no se instancia sino que es utilizada únicamente como una superclase. Debido que una clase abstracta no tiene instancias, se utiliza como un “modelo” para la creación de subclases en lugar de un “modelo” para la creación de objetos. Los frameworks los utilizan como diseños de sus componentes. Una manera en que una clase abstracta actúa como un diseño es que ésta especifica la interfaz de sus subclases. Cuando se indica que se desea utilizar una instancia de una clase abstracta, siempre significa una instancia de una clase que cumple con la interfaz de la clase abstracta. Adicionalmente menciona que los frameworks provén componentes reutilizables de software para el desarrollo de aplicaciones, integrando un conjunto de clases abstractas y definiendo una manera estándar de cómo las instancias de estas clases deben colaborar. Cabe destacar que el alcance de la reutilización en un framework puede ser significativamente mayor que el uso de librerías de funciones tradicionales o librerías convencionales de clases orientadas a objetos.

Generalmente, los frameworks se diferencian de las librerías de clases tradicionales ya que los primeros definen una aplicación “semi completa” que encierra estructuras de objetos de un dominio específico mientras que las librerías de clases proporcionan una pequeña granularidad de reutilización. Por otro lado los componentes que forman parte de un framework colaboran entre ellos a fin de proveer un modelo arquitectónico que sea familiar a un conjunto de aplicaciones específicas.

Es importante remarcar que un framework reduce la cantidad de código de una aplicación específica que pertenece a un dominio específico.

En la práctica, los frameworks y las librerías de clases son tecnologías complementarias. Los frameworks de manera frecuente utilizan librerías de clases internamente para simplificar el diseño del framework.

Por otro lado [53] resalta que la reutilización de arquitecturas software se define dentro un framework. En general, un framework se suele definir de la siguiente forma: *“Es un diseño reutilizable de todo o parte de un sistema, representado por un conjunto de clases abstractas y la forma en la cual sus instancias interactúan”*

Otra forma de expresar este concepto es: *“Es el esqueleto de una aplicación que debe ser adaptado a necesidades concretas por el programador de la aplicación”*

En ambas definiciones podemos ver que un framework encapsula el patrón de la arquitectura software de un sistema o de alguna de sus partes, que puede además estar parcialmente instanciada.

Las principales ventajas que ofrecen los framework son la reducción del costo de los procesos de desarrollo de aplicaciones software para dominios específicos, y la mejora de la calidad del producto final.

En [54] define un framework de software como un esqueleto de programa que puede ser reutilizado o adaptado de acuerdo a reglas bien definidas para resolver un conjunto de problemas similares.

Los frameworks tienen como objetivo ofrecer una funcionalidad definida, auto contenido, siendo construidos usando patrones de diseño, y su característica principal es su alta cohesión y bajo acoplamiento. Para acceder a esa funcionalidad, se construyen piezas, objetos, llamados objetos calientes, que vinculan las necesidades del sistema con la funcionalidad que este presta. Esta funcionalidad, está constituida por objetos llamados fríos, que sufren poco o ningún cambio en la vida del framework, permitiendo la portabilidad entre distintos sistemas.

Por otro lado es relevante destacar la importancia de la abstracción para construir un framework. Por último [54] detalla que la abstracción es un proceso conceptual por el cual un diseñador construye una vista simplificada de un sistema como un conjunto de interfaz. La implementación de esta interfaz en términos de entidades más detalladas se deja posteriormente para un mayor refinamiento. Un sistema complejo puede por lo tanto ser descrito en diferentes niveles de abstracción.

Uno de los objetivos principales en el diseño de un software es la reutilización del código y de las funcionalidades particulares para desarrollar una aplicación determinada.

Como detalla [51], una visión original en la reutilización de software se basó en la utilización de componentes. El interés comercial en el desarrollo orientado a objetos se centró también en la reutilización de código, de hecho, los diseños de software se focalizaron en esta idea dando lugar a los patrones de diseño. El concepto comenzó a madurarse en el tiempo dando lugar a la creación de estructuras de diseño más complejas que posteriormente se denominarían frameworks.

Pero los frameworks son formas intermedias, es decir, se reutiliza parte de código y parte del diseño. Lo destacado de un framework es que elimina la necesidad de realizar un nuevo diseño por completo utilizando como lenguaje de programación el paradigma orientado a objetos.

Cabe mencionar que la reutilización y los frameworks tienen muchas motivaciones. El principal de todas es la de optimizar y acelerar el tiempo de desarrollo que conlleva a ahorrar dinero. En el desarrollo de aplicaciones comerciales el tiempo de mercado (time to market) es de vital importancia estratégica y muchas compañías se encargan de desarrollar sus propios frameworks (o reutilizar algunos existentes) a fin de achicar el tiempo de salida al mercado de un producto de software específico.

Debemos considerar que el diseño de un framework se asemeja al diseño de la mayoría del software reutilizables. Generalmente se da comienzo con el análisis del dominio que se encarga, entre otras funciones, de obtener un número considerable de ejemplos. La primera versión del framework generalmente se diseña a fin de poder implementar los ejemplos, obteniéndose un framework de “caja blanca”. A partir de ese momento se puede utilizar el framework para construir aplicaciones de software. Sin embargo, estas aplicaciones son duras de desarrollar. La experiencia y el método de prueba y error, conduce a mejoras en el diseño del framework pasando a ser un framework de “caja negra”. Este proceso puede continuar durante mucho tiempo hasta que, finalmente, el framework es lo suficientemente bueno donde las sugerencias de mejoras son raras. En este punto el diseñador tiene que decidir la finalización del framework para liberar una versión final [51].



Por último es importante mencionar la diferencia entre librerías (Biblioteca), APIs y frameworks. Una biblioteca es un conjunto de elementos (funciones, clases, tipos predefinidos, constantes, variables globales, macros, etc.) que es posible utilizar en un programa para facilitar la implementación de ese programa.

API viene del inglés "Application programming interface" que significa "Interfaz para programación de aplicaciones". Es la parte de una biblioteca a la que accede un programa que usa la biblioteca; haciendo así el uso de la biblioteca independiente de los detalles de implementación. Una API puede ser implementada por distintas bibliotecas. Un framework es un conjunto integrado de herramientas que facilitan un desarrollo software. Puede incluir APIs y bibliotecas. Pero también puede incluir otros elementos como herramientas de depuración, diseño gráfico, prototipado, edición, etc.

## 4.2 Comparativa de frameworks de RA más utilizados

Conocidos también como SDK o simplemente frameworks, en [55] mencionan que estas herramientas están compuesta por varios componentes, tales como: reconocimiento, tracking y rendering, entre los principales. Teniendo en cuenta el ascenso en el uso de esta tecnología en diferentes segmentos, se concluye que existen diversos frameworks de RA en el mercado, sean estos de libre uso o comerciales. Cabe aclarar que existen en la actualidad una cantidad interesante de frameworks que ejecutan de manera óptima las características necesarias para desarrollar una aplicación de RA.

Analizando los cuadros comparativos realizados en la sección 3.3 - *Comparativa de proyectos militares que utilizan RA* correspondiente a los framework de RA para el uso militar, en las siguientes tablas comparativas se representan los aspectos importantes para desarrollar aplicaciones completas de RA. Por otra parte, tomando como criterio de comparación lo aportado por [56], se definen un conjunto de parámetros funcionales tecnológicos entre los frameworks seleccionados:

- *Reconocimiento de objetos 3D*: reconocer formas tridimensionales de objetos. Los frameworks evaluados deben poder realizar tracking de objetos en 3D.

- *Reconocimiento de imágenes*: de la misma forma que el parámetro anterior, los frameworks seleccionados deben poder soportar el reconocimiento de imágenes en 2D.
- *Geolocalización*: se evaluó cuáles frameworks admiten el uso de geolocalización.
- *Reconocimiento sin uso de marcadores*: los frameworks seleccionados se evaluaron para identificar si soportan el uso de tecnología de RA que no requiera un marcador.
- *Reconocimiento de objetivos vía internet*: Se analizaron los frameworks hacen posible el reconocimiento de objetivos disponibles solo a través de internet en tiempo real.

Tabla 4.1 Parámetros funcionales tecnológicos

Parámetros funcionales tecnológicos	ARTool-Kit	Vuforia	EasyAR	Layar	DroidAR
Reconocimiento de objetos 3D	-	X	-	-	X
Reconocimiento de imágenes	X	X	X	X	X
Geolocalización	X	X	X	X	X
Reconocimiento sin uso de marcadores	-	X	-	-	-
Reconocimiento de objetivos vía internet	-	X	-	X	-

Adicionalmente a las funcionalidades antes mencionadas lo aportado por [57], ofrece una lista de parámetros comparativos centrándose en los aspectos no funcionales. Estos parámetros son los siguientes:

- *Disponibilidad off-line*: En el caso de operaciones militares en regiones donde la conexión es limitada, es necesario contar con aplicaciones que accedan a fuentes de datos almacenados en los dispositivos móviles.
- *Facilidad de integración con otras aplicaciones*: Aunque esto no es esencial, la integración con otros productos facilitarían el acceso o el intercambio de información. Por ejemplo, la aplicación desarrollada podría acceder directamente a la información de una base de datos y mostrar el resultado mediante el uso de RA.

- *Facilidad en ampliar el framework*: En general, no es fácil ampliar los frameworks ya que el código no está disponible y las restricciones de licencia prohíben extender el marco.
- *Tipo de licencia*: La mayoría de los frameworks están disponibles bajo una licencia propietaria, aunque algunos ofrecen una opción desarrollo libre (freeware) con funcionalidades limitadas y otros de código abierto (open source) que facilitan ampliar o modificar el framework.
- *Procesamiento distribuido*: La capacidad de procesar las imágenes en una arquitectura cliente servidor resulta de gran utilidad para computar grandes volúmenes de información.

Tabla 4.2 Parámetros no funcionales

<b>Parámetros no funcionales</b>	<b>ARTool-Kit</b>	<b>Vuforia</b>	<b>EasyAR</b>	<b>Layar</b>	<b>DroidAR</b>
Disponibilidad off-line	X	-	-	-	X
Facilidad de integración	-	-	-	-	-
Facilidad en extender el framework	X	-	-	-	X
Licencia	Open Source	Comercial	Freeware	Freeware	Open Source
Soporta procesamiento distribuido	-	-	-	-	-

Si se toma como referencia principal la tabla comparativa detallada en la sección 3.3

- *Comparativa de proyectos militares que utilizan RA*. De este análisis se llega a la conclusión que ninguno de ellos es completo en cuanto al aporte de funcionalidades que sirvan para hacer frente a la problemática central. Si bien los frameworks se acercan bastante a lo solicitado por los usuarios claves, el esfuerzo por completar las funcionalidades necesarias se consideró muy grande pues estos frameworks no fueron específicamente diseñados para el ámbito militar pues poseen particularidades que difieren bastante de aquellos frameworks utilizados para desarrollar aplicaciones en el ámbito civil. Cabe aclarar que los framework

evaluados aportan parte de las funcionalidades e incluso la información como se debe presentar en los dispositivos de visualización, sean estos, smartphone, gafas de RA, etc,

Por último corresponde señalar que siguiendo lo propuesto por [57], en la evaluación se desestimaron los frameworks que:

- No soportaban GPS,
- No soportan dispositivos móviles,
- No soportaba Android como plataforma de desarrollo,
- No poseían soporte para sensores inerciales,
- Eran únicamente del tipo browser o visualizador 3D de RA,
- Estaban inactivos o eran considerados irrelevantes

## **4.3 Arquitectura de software por capas y componentes**

En [58] definen el estilo en capas como una organización jerárquica tal que cada capa proporciona servicios a la capa inmediatamente superior y se sirve de las prestaciones que le brinda la inmediatamente inferior. En algunos ejemplares, las capas internas están ocultas a todas las demás, menos para las capas externas adyacentes, y excepto para funciones puntuales de exportación; en estos sistemas, los componentes implementan máquinas virtuales en alguna de las capas de la jerarquía. En otros sistemas, las capas pueden ser sólo parcialmente opacas. En la práctica, las capas suelen ser entidades complejas, compuestas de varios paquetes o subsistemas. También mencionan que las ventajas del estilo en capas son obvias. Primero que nada, el estilo soporta un diseño basado en niveles de abstracción crecientes, lo cual a su vez permite a los implementadores la partición de un problema complejo en una secuencia de pasos incrementales. En segundo lugar, el estilo admite muy naturalmente optimizaciones y refinamientos. En tercer lugar, proporciona amplia reutilización.

Como menciona [54] un sistema de software se encuentra organizado por un conjunto de partes o componentes. El sistema en su conjunto, y cada uno de sus componentes, cumple una función que puede ser descrito como la prestación de un servicio. Con el fin de proporcionar su servicio, un componente por lo general se basa en los servicios prestados por otros componentes. Por el bien de la uniformidad, el sistema en su conjunto puede considerarse

como un componente, que interactúa con un entorno definido externamente; el servicio provisto por el sistema se basa en suposiciones acerca de los servicios que el ambiente provee al sistema.

Un componente de software individual es un paquete de software, un servicio web, o un módulo que encapsula un conjunto de funciones relacionadas (o de datos)<sup>3</sup>.

Todos los procesos del sistema son colocados en componentes separados de tal manera que todos los datos y funciones dentro de cada componente están semánticamente relacionados. Debido a este principio, con frecuencia se dice que los componentes son modulares y cohesivos.

Con respecto a la coordinación a lo largo del sistema, los componentes se comunican uno con el otro por medio de interfaz. Cuando un componente ofrece servicios al resto del sistema, éste adopta una interfaz proporcionada que especifica los servicios que otros componentes pueden utilizar, y cómo pueden hacerlo. Esta interfaz puede ser vista como una firma del componente - el cliente no necesita saber sobre los funcionamientos internos del componente (su implementación) para hacer uso de ella. Este principio resulta en componentes referidos como encapsulados.

La reusabilidad es una importante característica de un componente de software de alta calidad. Los programadores deben diseñar e implementar componentes de software de una manera tal que diversos programas puedan reutilizarlos. Además, cuando los componentes de software interactúan directamente con los usuarios, debe ser considerada la prueba de usabilidad basada en componentes.

## 4.4 Ambiente de Desarrollo

El sistema operativo Android es un proyecto de código abierto iniciado por Google, Inc. Es una plataforma diseñada principalmente para dispositivos móviles con pantalla táctil, como teléfonos inteligentes o tabletas, y también para relojes inteligentes, televisores y automóviles. Android contiene una pila de software donde se incluye un sistema operativo, middleware y aplicaciones básicas para el usuario. Android es desarrollado por la Open

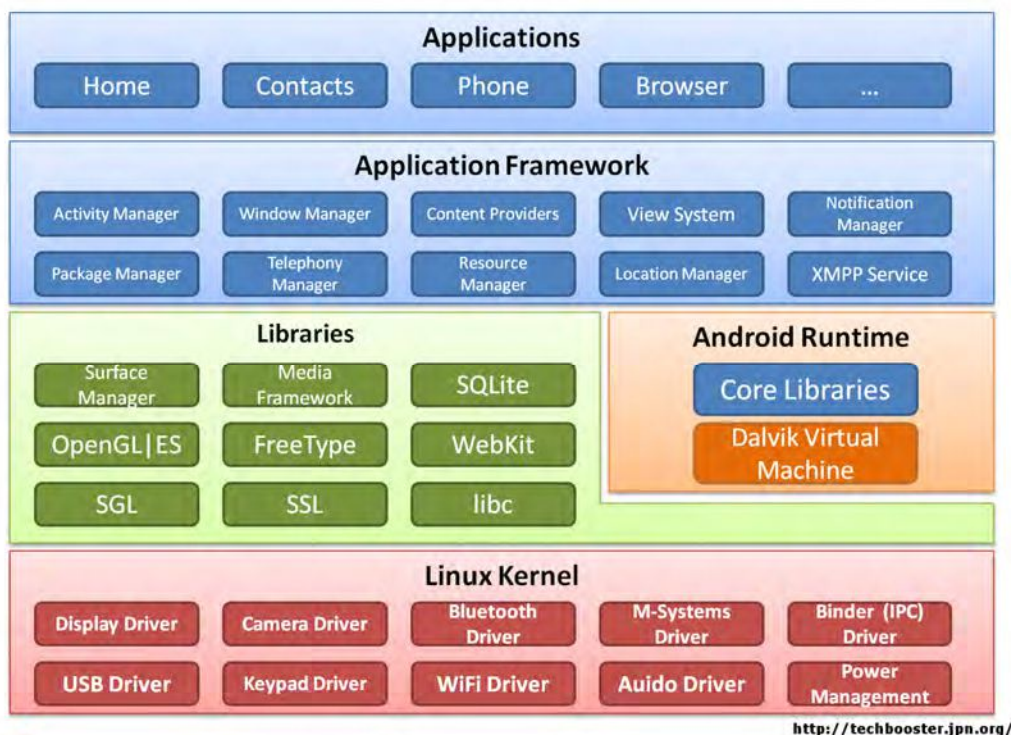
---

<sup>3</sup> [http://es.wikipedia.org/wiki/Ingenier%C3%ADa\\_de\\_software\\_basada\\_en\\_componentes](http://es.wikipedia.org/wiki/Ingenier%C3%ADa_de_software_basada_en_componentes)

Handset Alliance (OHA) un consorcio compuesto por fabricantes y desarrolladores de hardware, software y operadores de servicio., entre las que se encuentran Google, Inc., Motorola, HTC y Qualcomm. La arquitectura de Android está basada en el kernel de Linux versión 2.6. La plataforma de hardware principal de Android es la arquitectura ARM.

La estructura del sistema operativo Android se compone de aplicaciones que se ejecutan en un framework Java de aplicaciones orientadas a objetos sobre el núcleo de las bibliotecas de Java en una máquina virtual Dalvik con compilación en tiempo de ejecución. Las bibliotecas escritas en lenguaje C incluyen un administrador de interfaz gráfica (surface manager), un framework OpenCore, una base de datos relacional SQLite, una interfaz de programación de API gráfica OpenGL ES 2.0 3D, un motor de renderizado WebKit, un motor gráfico SGL, SSL y una biblioteca estándar de C Bionic.

El siguiente gráfico muestra la arquitectura de Android. Como se puede ver está formada por cuatro capas. Una de las características más importantes es que todas las capas están basadas en software libre.



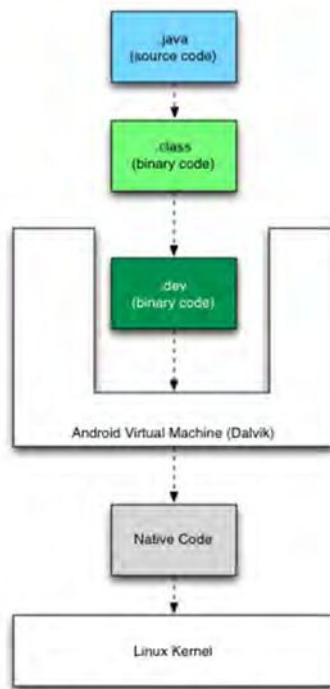
**Figura 4.1 – Arquitectura de Android**  
(fuente: <http://techbooster.jpn.org>)

Entonces, los componentes principales del sistema operativo de Android <sup>4</sup>:

- *Aplicación*: un conjunto de aplicaciones núcleo están en esta capa de alto nivel. Incluye, cliente de email, aplicación de SMS, calendario, aplicación de mapas, web browser, etc. Todas estas aplicaciones están escritas en lenguaje Java.
- *Framework de Aplicación*: el framework de Aplicación es la base para el desarrollo de aplicaciones Android. Los principales componentes del Framework de Aplicación son: El Administrador de Actividades, el Administrador de Ventanas, Los Proveedores de Contenidos, el Sistema de Vistas, el Administrador de Notificaciones, el Administrador de Paquetes, el Administrador de Telefonía, el Administrador de Recursos, el Administrador de Locación y el Servicio de XMPP.
- *Librerías*: Android incluye un conjunto de librerías en C/C++ utilizados por varios componentes del sistema Android. Las librerías principales son: Framework de Medios, WebKit, SGL, OpenGL ES, FreeType, SQLite, etc. Los desarrolladores pueden acceder a estas librerías por medio del Framework de Aplicación.
- *Android Runtime*: cada Aplicación Android se ejecutan en su propio proceso otorgado por el SO y su propia instancia de la máquina virtual Dalvik. La máquina virtual Dalvik ha sido escrita para que un dispositivo pueda ejecutar múltiples máquinas virtuales eficientemente. La máquina virtual Dalvik ejecuta archivos de formato .dex (Dalvik Executable) el cual fue optimizado para utilizar mínimamente los recursos de cpu y memoria.

---

<sup>4</sup> <http://es.wikipedia.org/wiki/Android>



**Figura 4.2 – Secuencia de ejecución de una aplicación Android**  
**(fuente: <http://techbooster.jp.n.org>)**

- *Kernel de Linux:* Android está basado en Linux (Kernel versión 2.6) atendiendo servicios de sistemas tales como Administración de memoria, Administración de procesos, Pila de Red, Seguridad y Modelo de Controlador. El núcleo actúa como una capa de abstracción entre las aplicaciones y el hardware.



## **II. Framework Propuesto**

# Capítulo 5

## Diseño de las Arquitecturas

### 5.1 Introducción

Se considera una arquitectura basada en *capas* para el diseño del middleware del framework propuesto pues facilita la integración tecnológica. Adicionalmente una de las características relevantes del framework propuesto radica en la arquitectura Cliente / Servidor, es decir, la capacidad de implementar (desplegar) las aplicaciones que se desarrollen en una arquitectura distribuida en donde en el lado del Servidor se ejecutarán las tareas más intensivas que corresponden al procesamiento de imágenes, tal cual se menciona en la sección 2.8 - *Arquitectura de software para aplicaciones de RA*.

Por último se plantea como será la arquitectura y los componentes de hardware que forman parte del sistema en el cual se implementarán las soluciones de software que extiendan del framework de RA.

### 5.2 Metodología de desarrollo utilizada

Para el desarrollo del framework se evaluaron diferentes métodos, sobre todo aquellos que se ajustaban mejor al cambio constante por parte de los usuarios de las funcionalidades ya que al tratarse de la construcción de un marco de trabajo y de aplicaciones de prueba nuevas sin tener un conocimiento previo de la tecnología propuesta se hacía difícil interpretar tanto las necesidades como lo que deseaba experimentar el usuario. Para llevar a cabo el proceso de construcción del framework se basó en el desarrollo de pruebas de concepto individuales a modo de prototipos con el fin de que se demuestre el funcionamiento y el cumplimiento de lo solicitado. Para ello el diseño del framework y el desarrollo de las aplicaciones se basaron en una metodología combinada entre el uso de un método ágil y las técnicas de DCU. Por

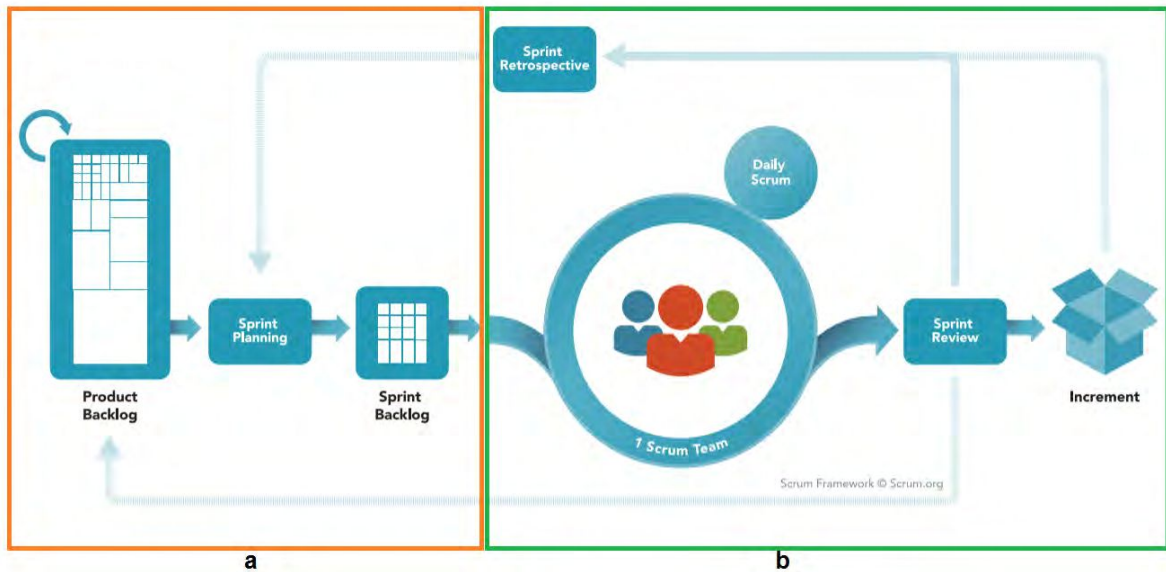
otro lado, se escogió este método de trabajo debido a que el segmento militar era poco explorado y era necesario diseñar, programar y mostrar a los usuarios claves el funcionamiento de cada parte del framework como unidades independientes, que posteriormente, se integrarían en el framework final.

Como menciona [59] el interés en la integración de DCU y los enfoques de desarrollo ágil está creciendo. Aunque las metodologías ágiles y DCU son dos enfoques de desarrollo de software diferentes, comparten principios comunes, tales como: diseño iterativo, participación del usuario, pruebas continuas y creación de prototipos. DCU es un proceso de diseño UX para interfaz de usuario (software) que se puede ejecutar de forma diferente. Sin embargo, es fundamental para esto la alineación con las necesidades del usuario y el procedimiento iterativo (incluso utilizando un método ágil), con pasos de proceso repetitivos.

Por las razones mencionadas anteriormente se optó por utilizar SCRUM<sup>5</sup> como método ágil de desarrollo de software en conjunto con los principios de diseño que aporta el DCU resultando una metodología combinada SCRUM-DCU. Siguiendo el criterio de SCRUM-DCU, una de las principales tareas es la realización de prototipos evolutivos. Desarrollar prototipos de conceptos de diseño permite entender un diseño particular, un entendimiento en común de lo que el sistema podría hacer y cómo podría verse. Estos prototipos proporcionan la base para detectar problemas de diseño y deficiencias al principio del proceso de diseño. Sin el desarrollo de estos prototipos y modelos, se detectarían muchos problemas y deficiencias importantes en etapas tardías del proceso de diseño y desarrollo de la interfaz.

---

<sup>5</sup> SCRUM: <https://www.scrum.org/>



**Figura 5.1 – SCRUM + DCU**

(fuente: scrum.org)

En la figura anterior se encuentra cada tarea del proceso SCRUM, incluido los principios de diseño del DCU.

Al tratarse del diseño de una interfaz avanzada es importante interactuar con el o los usuarios claves en el proceso de diseño. Para ello es importante trabajar con prototipos de diseño conceptual de lo que será la interfaz y el tipo de información que formará parte de la misma. Es importante considerar que es necesario realizar un diseño orientado a la CS que sea eje central en el proceso de DCU. Cabe destacar que sin un análisis sistemático de los requisitos operacionales de los usuarios, sin una guía de factores humanos sólidos, y un programa de pruebas sistemático y objetivo, es casi imposible crear un sistema centrado en el usuario.

En [60] detalla que la CS solo existe en la mente del operador humano, presentar grandes cantidades de datos no servirá de nada a menos que la persona lo transmita, absorba y asimile de manera oportuna para formar la CS. Uno de los beneficios clave de observar la CS es que ésta le transmite al diseñador cómo deben combinarse todos los datos disponibles para facilitar la comprensión del usuario. Proporcionar sistemas que respalden el proceso de la CS es esencial para enfrentar un desafío crítico en los entornos complejos de la actualidad.

En entornos complejos en donde la CS es un factor clave para la toma de decisiones determina que el desarrollo de software se centre en un enfoque estructurado que incorpore las consideraciones de la CS en el proceso de diseño, incluya la determinación de los requisitos de CS, el diseño para la mejora de CS y la medición y evaluación de la CS en el proceso de construcción.

## 5.3 Arquitectura del Framework. Middleware

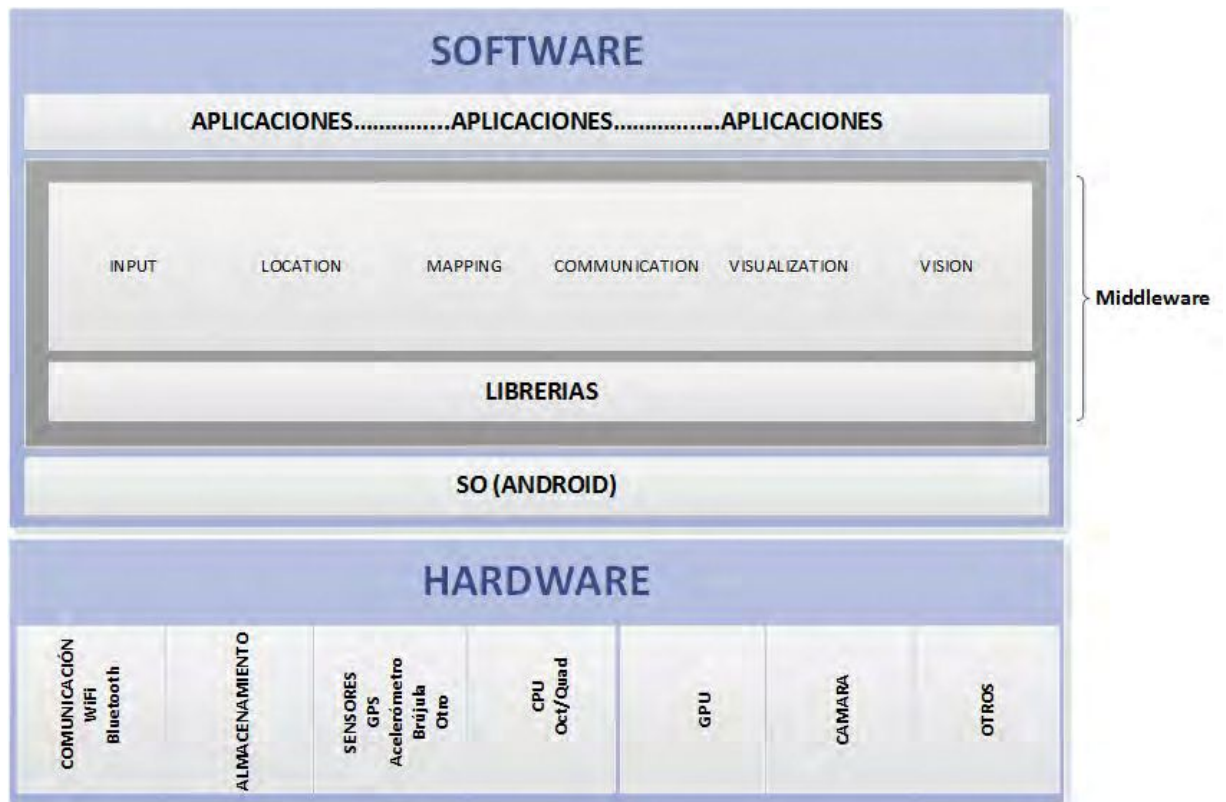
El principal trabajo es diseñar la arquitectura de la capa intermedia, es decir, el *middleware* que está compuesto por la capa de los *Módulos* y la capa de las *Librerías* que se detallan más abajo. Cada capa representa un nivel de abstracción en la arquitectura y está formada por un conjunto de componentes específicos. Las capas de software están formadas por las aplicaciones (que se desarrollarán tomando como base el framework) en la capa de más alto nivel, *middleware* (es decir, el framework propiamente dicho), librerías (bibliotecas de procesamiento de imágenes o visión artificial, manejo de mapas, comunicaciones, desarrollo de librerías propias, etc.) y el sistema operativo, en el nivel más bajo. Cada una de estas capas utiliza servicios ofrecidos por las anteriores, y ofrece a su vez los suyos propios a las capas de niveles superiores. De la misma manera, los componentes que forman parte del *middleware* se comunican entre sí ofreciéndose servicios entre ellos.

### 5.3.1 Capa: Módulos

Tomando como base el Modelo de Referencia para un Sistema de RA detallado en la sección 2.2 - *Partes de una aplicación de RA*, se diseñaron los componentes de software o módulos que forman parte del *middleware*:

- *Input*: Este módulo permite que se puedan ingresar datos al sistema, ya sea en forma de voz o gestos. La entrada vocal es útil a la hora de dar instrucciones simples al sistema (acciones que no requieran una comprensión total del entorno), por ejemplo mostrar u ocultar un mapa, mostrar la ruta recorrida, establecer algún recordatorio o punto de referencia, etc. Este módulo cuenta con las clases relacionadas con el reconocimiento de los gestos y de la voz.

- *Location/Tracking*: Este módulo provee una interfaz unificada para la obtención de información de georeferenciación del dispositivo y al mismo tiempo devuelve la información (si se encuentra disponible) del sensor magnético y de la unidad de medición inercial (IMU). Permite realizar el *tracking* del usuario a través del uso de los sensores del dispositivo.
- *Mapping*: El módulo de mapeo implementa una interfaz a un SIG completamente funcional que permite cargar cartas prediseñadas a la aplicación (creadas con cualquier aplicación GIS compatible como ser ArcGIS, QGIS, OSM, GRASS, etc) y ubicar elementos fácilmente para su posterior visualización y manipulación ya sea en una carta 2D como en un mundo 3D.
- *Communications*: El módulo de comunicaciones consta de una simple abstracción que permite a dos o más dispositivos comunicarse entre sí mediante mensajes estructurados. El objetivo es testear la comunicación entre un cliente (implementado en JAVA-Android) y un servidor (implementado en JavaScript-Node.js). Como tecnología de comunicación entre estos componentes se utiliza Socket.io
- *Visualization*: Representación gráfica del lado del Cliente (dispositivo móvil) de la simbología u objeto renderizado. Radar 360 grados con alertas y amenazas, mapas de operaciones, estados del dispositivo, etc., son representados visualmente en la pantalla del dispositivo móvil. El filtrado de información prioritaria es procesada en este módulo para su visualización.
- *Vision*: El objetivo de este módulo es brindar las implementaciones de las clases necesarias para el reconocimiento de objetos. Se implementarán diferentes algoritmos, por ejemplo, detección y extracción de características, de matching (emparejamiento) y de clasificación. La finalidad es brindar al desarrollar de aplicaciones la mayor cantidad de algoritmos para que pueda implementarlo en el desarrollo de las aplicaciones correspondientes.

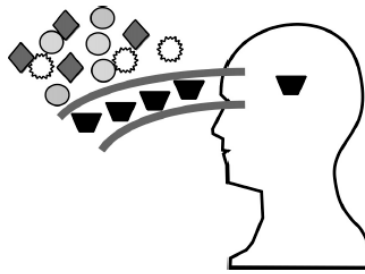


**Figura 5.2 – Diseño basado en componentes de la arquitectura del framework (fuente: propia)**

A continuación se detallan las características principales de los módulos de Visualización y Vision:

### **Módulo Visualization**

El filtrado de información prioritaria corresponde a aquellos datos que el operador puede visualizar como primer capa de información contextual. El objetivo principal es de no abrumar al operador con abundantes datos del contexto ya que lo puede llevar a perder la concentración en las tareas que está ejecutando. De acuerdo a lo detallado por [61], el DCU no sólo significa presentar a los usuarios la información que necesitan en un momento dado sino que un intento bien intencionado para ayudar a administrar el problema de sobrecarga de información es la idea del filtrado de información.



**Figura 5.3 – Filtrado de información. Cierta conjunto de información es prioritaria.**

**Se excluye el resto de información**

**(fuente: Endsley at al. [61])**

En el framework propuesto se implementó la funcionalidad para que soporte hasta 2 capas de información. El diseño y disposición de los objetos en la pantalla se desarrolló siguiendo el criterio de DCU focalizado en la CS de los usuarios tal cual se detalló en la sección mencionada con anterioridad.

La primera capa de información se visualiza siempre directamente en la pantalla del dispositivo móvil Cliente.

En cuanto a la segunda capa, la información es solicitada a demanda por el operador ya sea a través de reconocimiento gestual o vocal.

#### **Filtro de información 1:**

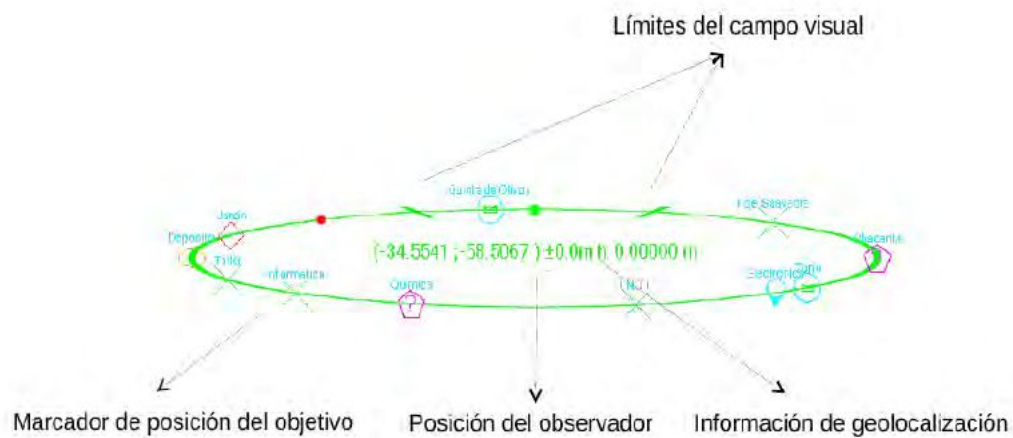
La primera capa de información está compuesta por:

- Radar 360 grados con información de alertas y amenazas
- Ubicación del operador y de los soldados en el Radar
- Información de geolocalización
- Comunicación con el CII
- Información del estado general del dispositivo e información del lugar y fecha

En la siguiente figura se detalla la estructura de visualización del radar de 360 que consta de una elipse que representa la orientación del usuario. El observador u operador se ubica en el centro de dicha elipse. Un punto rojo marca el norte y sobre la línea verde se sitúan los objetivos precargados en el sistema (o recibidos mediante mensajería). Por otro lado el radar



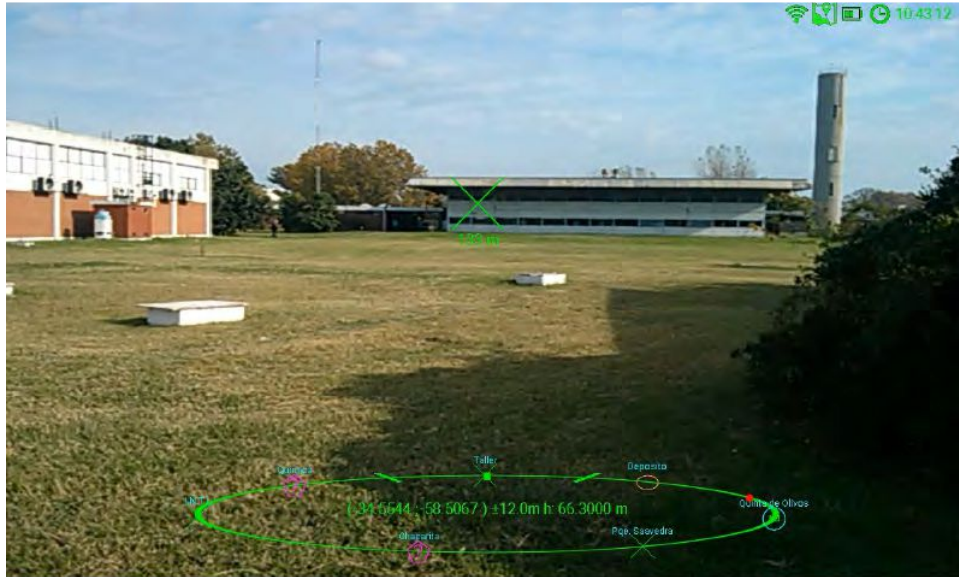
está provisto de información de geolocalización y, además, posee los límites del campo visual en donde los objetivos son mostrados en el campo visual del observador. Los elementos dentro del campo visual son representados espacialmente en pantalla mientras que el resto son ocultados.



**Figura 5.4 – Simbología del radar. Estructura de la visualización del radar (fuente: propia)**

El radar es la parte central del sistema y es lo que facilita la navegación, marcando la dirección y distancia de los distintos elementos que el sistema reconoce. Utilizando la API de localización, provista por el módulo *Location*, el sistema obtiene las coordenadas GPS del usuario y al mismo tiempo su orientación y velocidad de desplazamiento. Esta información es la que permite calcular los diferentes rumbos y distancias necesarios para construir la visualización.

En la siguiente figura se puede apreciar que en la parte inferior de la pantalla se encuentra el radar de 360 grados con el uso de simbología (iconografía) específica.



**Figura 5.5 – Mapa 360 grados del Prototipo desarrollado utilizando el framework (fuente: propia)**

Uno de los principales problemas o limitaciones de este radar es la precisión que puede obtener un GPS de dispositivo móvil. La precisión de la ubicación siempre es mayor a los 10 metros, haciendo que la ubicación de los objetivos cercanos sea poco confiables y experimenten cambios repentinos durante la marcha. Para mitigar estos errores se utiliza un *filtro pasabajos* en la ubicación del GPS, evitando así los saltos bruscos. No es una solución al problema real, pero evita los saltos repentinos de los objetos, brindando una experiencia más amigable al usuario.

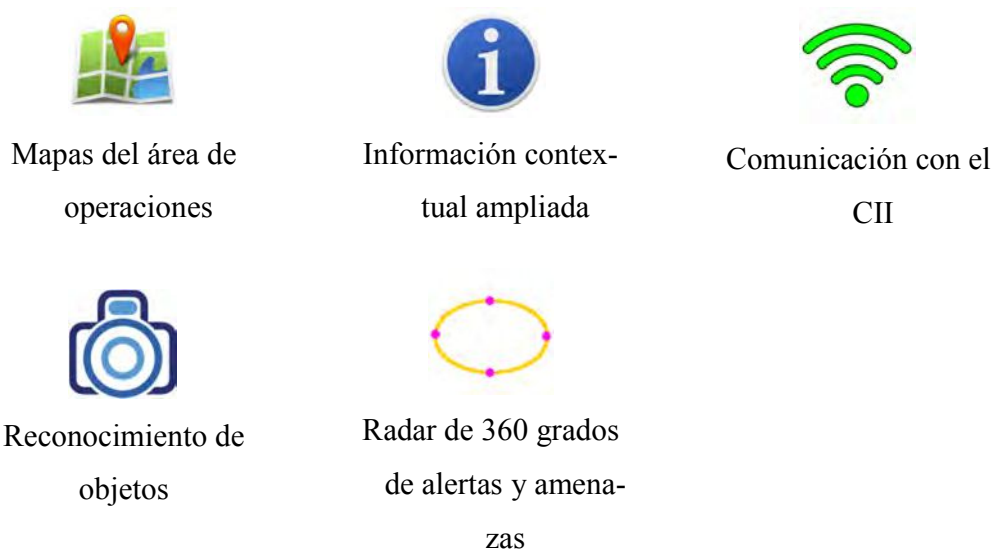
Dentro de la simbología militar utilizada se dispuso un conjunto de símbolos con el objetivo de probar el fácil intercambio de éstos en la configuración de la aplicación. Para una mejor interpretación de los mismos se los diferencié del tipo de alerta / amenaza.

### **Filtro de información 2:**

La segunda capa de información está compuesta por:

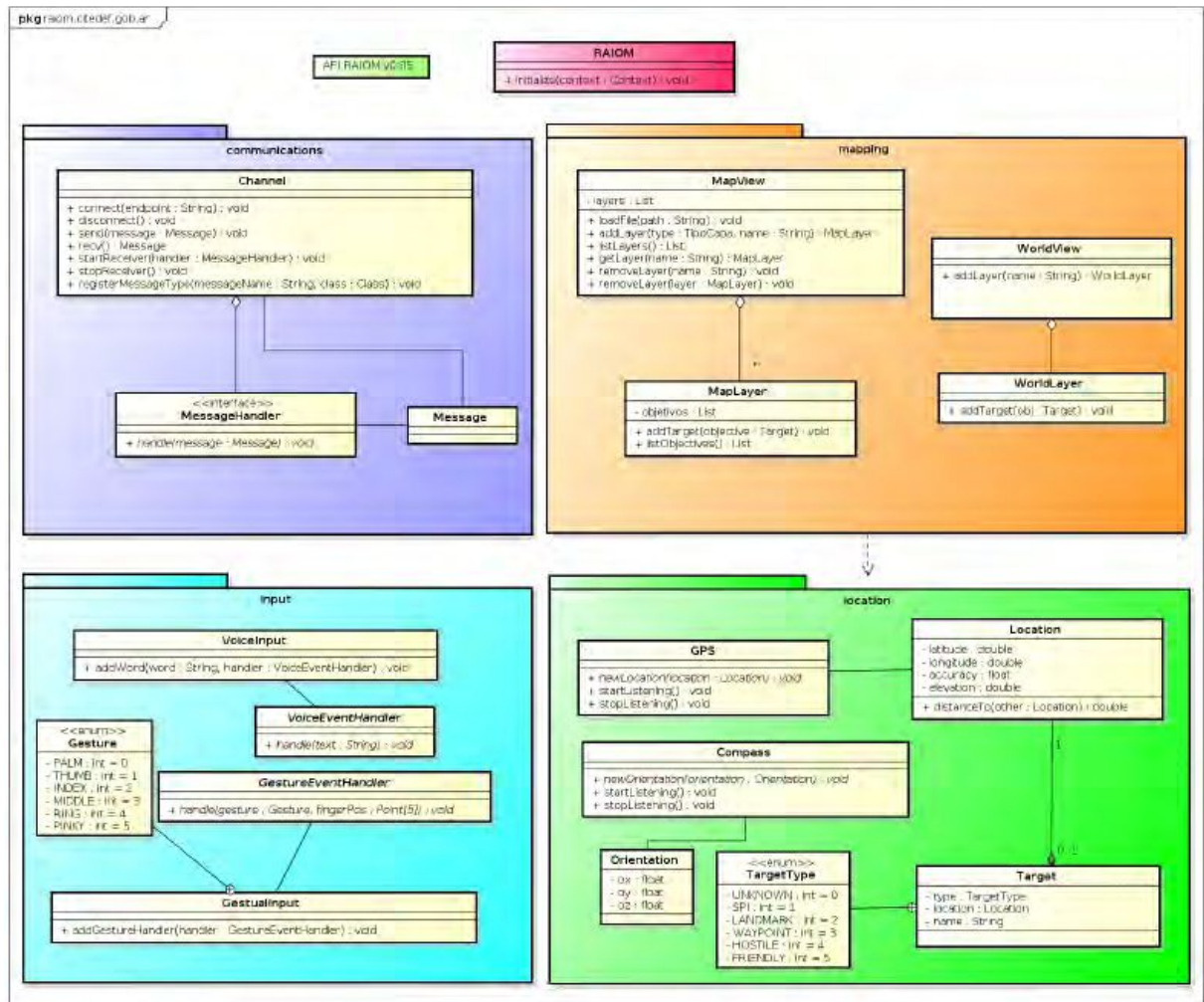
- Mapas del área de operaciones
- Información contextual ampliada
- Reconocimiento de objetos
- Radar de alertas y amenazas

La capa de información siguiente está compuesta por acciones que se solicitan a demanda por el usuario a través de comandos gestuales o vocales. El framework soporta hasta 5 comandos gestuales diferentes cada una asociada a un dedo de la mano.



**Figura 5.6 – Simbología gestual  
(fuente: propia)**

En la siguiente figura se presenta el diseño completo de la estructura de paquetes / módulos del framework propuesto:



**Figura 5.7 – Diseño de la estructura de paquetes / módulos del framework (fuente: propia)**

Una de las características más relevantes de la arquitectura del framework es que el diseño arquitectónico soporte el despliegue de las aplicaciones que se desarrollen (y utilicen al framework) en un ambiente distribuido del tipo Cliente / Servidor. Más detalles en la sección 5.5 - *Proceso distribuido*. Se decidió que el framework sea desarrollado bajo el sistema operativo Android pues al ser de código abierto permitirá tener mayor libertad de integración de las clases de este marco de trabajo con los recursos de las capas más bajas. En cuanto al lenguaje de programación principal utilizado en el diseño del framework será Java ya que estará basado en Android, en el lado del Cliente y Python en el lado del Servidor.

A continuación se detalla el módulo de *Vision* que es el más complejo y más extenso del framework propuesto:

## Módulo Vision

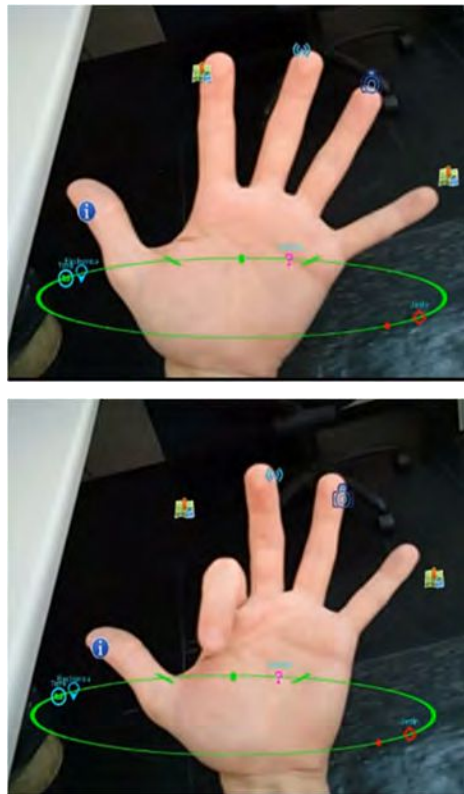
El módulo de *Vision* y el módulo de *Input* se apoyan en el uso de la librería OpenCV que se encuentra en la capa de *Liberías* (ver sección 5.3.2 - *Capa: Librerías*) del framework propuesto.

Como se mencionó en la sección 5.5 - *Proceso distribuido*, el procesamiento de imágenes se realiza del lado del Servidor ejecutándose sobre un dispositivo ODROID-XU3. Se han experimentado con diversos algoritmos de extracción de características con el objetivo de reconocer objetos determinados y se ha evaluado tanto su nivel de invariancia como de performance. Se tomó como referencia para determinar cuál o cuáles algoritmos son los más adecuados las características del ambiente. Estas características estaban determinadas por la luminosidad del entorno, tamaño y rotación de los objetos.

El framework posee un componente de entrada (módulo *Input*) que le permite al usuario ingresar datos a través del reconocimiento gestual y vocal. Con gestos simples se le puede indicar al dispositivo que se quiere ejecutar uno de varios comandos posibles. Al colocar la palma de la mano abierta frente a la cámara se presentan hasta un máximo de 5 opciones de menús. Para elegir una opción basta con flexionar el dedo en cuestión, pasando a otro menú o ejecutando una acción específica. Existen varios métodos de reconocimiento gestual. Todos presentan un paso de segmentación, y la mayor parte de ellos utiliza segmentación por color. La segmentación por color consiste en buscar en la imagen todas las regiones que contienen colores parecidos a la piel. Luego de encontradas las regiones se analizan con técnicas que varían desde reconocimiento por forma hasta reconocimiento por clasificador de imágenes utilizando características o PCA (Principal Component Analysis). El problema central de casi todos los algoritmos en esta área es que están pensados o desarrollados para ser utilizados en ambientes controlados. Cuando se utilizan en ambientes exteriores y/o con movimientos bruscos de cámara los resultados son pobres. Para el reconocimiento gestual se utiliza una técnica de segmentación adaptativa por color. Esta técnica requiere una calibración inicial con una imagen de la mano en cuestión. La calibración extrae un clúster de colores de la

mano. El procedimiento incluye una conversión de la imagen al espacio de color HSV. Mediante esta conversión se obtienen los rangos de HS (Hue y Saturation) que nos interesan para la mano, admitiendo cambios en V (Value) que permite que el reconocimiento sea medianamente robusto ante cambios de brillo.

Para realizar la segmentación de la mano se aplica la misma transformación a HSV y se buscan los sectores de la imagen con el rango HS adecuado. Se extraen los sectores candidatos (blobs) y se analiza la forma del contorno obtenido. Si se puede considerar que el sector corresponde a una mano se sigue con el procedimiento utilizando un algoritmo de *convex hull* y *convexity defects* que sirve para detectar la región de los dedos.



**Figura 5.8 – Pruebas iniciales de reconocimiento gestual y visualización de iconos  
(fuente: propia)**

El reconocimiento de objetos se lleva a cabo mediante técnicas de emparejamiento de características. Estas técnicas consisten en extraer un conjunto de características de una

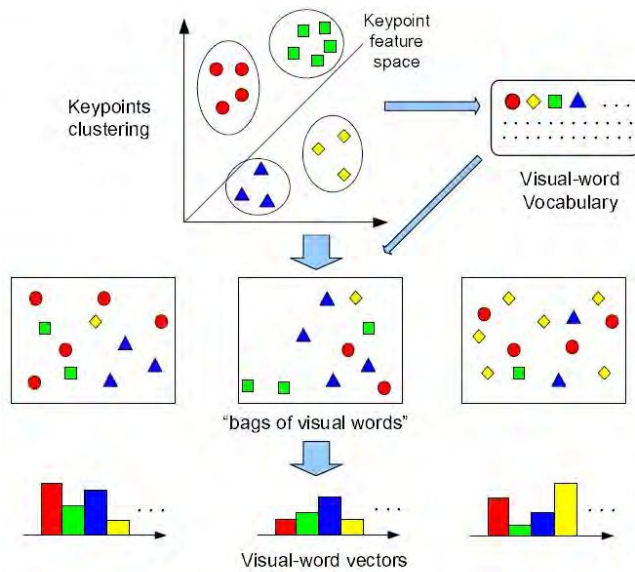
serie de imágenes de entrenamiento del objeto en cuestión. Cada conjunto de características está asociada al objeto de cuya imagen fueron extraídas.

Una vez extraídas todas las características se arma un modelo Bag of Word (BoW) [62], que es una técnica de clasificación de documentos aplicado a clasificación de imágenes. Cada imagen es tratada como un documento, mientras que cada característica como una palabra. El conjunto de todas las características componen el vocabulario. Con este vocabulario se arma un índice invertido o un hash que permita la recuperación de las imágenes a las que pertenece. BoW es un algoritmo utilizado originalmente para clasificación de documentos de texto. Cada documento es considerado una bolsa de palabras, donde no importa el orden de cada una. En procesamiento de imágenes se utiliza de manera similar, considerando a cada grupo de imágenes que represente un objeto como un documento, y por lo tanto como una bolsa. En vez de utilizar palabras se utilizan las características extraídas de la imagen mediante algún método de extracción de características como SURF, SIFT, FREAK o FAST.

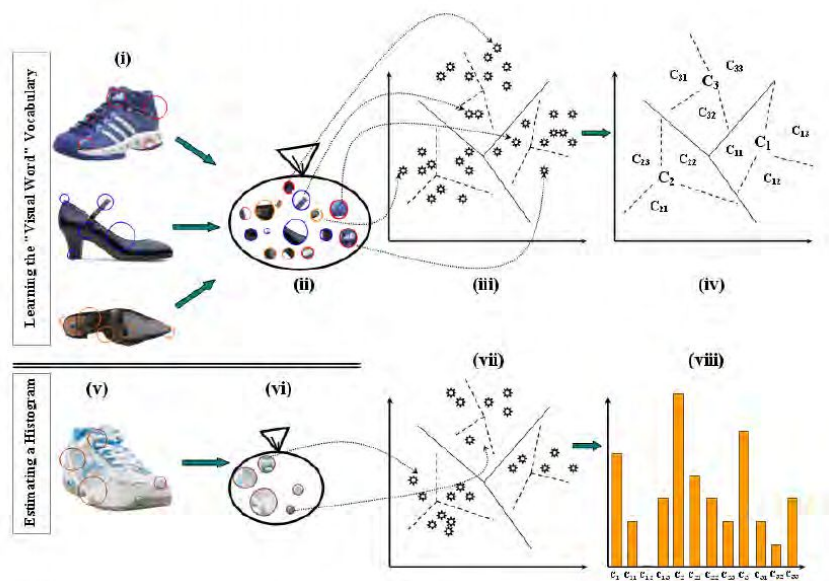
Para realizar el reconocimiento se extraen las características de la imagen en cuestión y se realiza la búsqueda en el índice. La técnica entonces permite encontrar imágenes similares y además clasificar imágenes según su parecido con otras.

En la siguiente figura se ilustra el modelo utilizado. Las características (features o landmarks) son descripciones de vértices, esquinas o algún otro punto saliente de una imagen. En definitiva son una descripción de puntos particulares (puntos clave, keypoints o interest points) que pueden ser fácilmente distinguidos (estas descripciones en la práctica son vectores multidimensionales).





**Figura 5.9 – Representación del modelo Bag Of Words utilizado para clasificación.  
(Fuente: Jun Yang et al [62])**



**Figura 5.10 – Explicación gráfica de Bag Of Words: (i) imágenes representando una palabra visual (ii) características (iii) descriptores ubicados en el espacio (iv) agrupación en clusters (v) imagen a clasificar (vi) características (vii) correspondencias con los descriptores conocidos (viii) histograma de correspondencias  
(fuente: Jun Yang et al [62])**



En general se quiere que estas características sean reconocibles aun habiendo sufrido la imagen cambios en el punto de vista, rotaciones o cambios de iluminación (es decir, que sean invariantes ante tales cambios).

Existen muchos algoritmos diferentes tanto para búsqueda de puntos clave como para extracción de características.

Del lado de la búsqueda de puntos clave se encuentran algoritmos como los detectores de esquinas de Harris y Shi-Tomasi, los detectores LoG (Laplacian of Gaussian), DoG (Difference of Gaussians), DoH (Determinant of Hessian), FAST (Features from Accelerated Segment Test), PCBR (Principal Curvature-Based Region detector) o MSER (Maximally Stable Extremal Regions) entre otros. Algunos de los algoritmos son más estables que otros ante cambios de brillo y orientación, algunos son más lentos que otros, pero en definitiva ninguno es perfecto y el uso de cada uno depende en gran parte de una elección personal y el compromiso que se pueda tomar entre velocidad y precisión necesaria [63].

Por el lado de los descriptores de puntos clave se encuentran algoritmos como SIFT, SURF, ORB, GLOH y LESH entre otros. Cada uno de los algoritmos realiza una o varias operaciones en los puntos clave encontrados.

SIFT [64] por ejemplo (el más popular y preciso, pero a la vez el que más proceso computacional requiere) calcula histogramas orientados en vecindades de 4x4 píxeles dentro de un parche de 16x16 píxeles en torno al punto clave. En total calcula 16 histogramas con 8 posibles orientaciones cada uno, resultando en un vector de características de 128 elementos.

Por otro lado, SURF utiliza sumas de transformadas Wavelet Harr, resultando en un algoritmo mucho más rápido. En este trabajo se optó por utilizar el detector FAST con el descriptor FREAK.

FAST [65] es ideal para aplicaciones de RA en dispositivos móviles gracias al escaso tiempo de procesamiento requerido. Es significativamente más rápido que los detectores tradicionales como DoG o Harris.

FREAK [66] es un descriptor que intenta imitar el funcionamiento del ojo humano. Utiliza un patrón de muestreo centrado en el punto clave que consiste en zonas circulares con un incremento en el tamaño y superposición de las zonas a medida que se alejan del centro, similar a la distribución de células receptoras en el ojo: alta concentración de conos (muy

sensibles) en el centro y luego en disminución hasta quedar solo bastones (menos sensibles). Esta forma de muestreo permite tener mejor resolución en el centro (el punto de interés) pero sin perder los detalles que lo rodean.

Adicionalmente el algoritmo introduce una búsqueda rápida de correspondencia: búsqueda sacádica. Esta técnica también intentando imitar la forma en la que funciona el ojo: el ojo no se queda generalmente fijo en un punto, si no que recorre la escena con movimientos rápidos (sacádicos).

El algoritmo busca correspondencias primero comparando la parte del descriptor correspondiente a la zona de menor resolución, si esta zona presenta cierta similitud dentro de márgenes razonables el algoritmo continúa la comparación con los niveles de mayor resolución. Si la zona de menor resolución no presenta parecido, pasa a otro descriptor. De esta forma los descriptores sin correspondencias se descartan rápidamente.

En contrapartida a todas las ventajas que presenta, FREAK tiene la desventaja de no ser tan robusto ante cambios de escala para factores de 2.5 en adelante como lo son otros descriptores como SIFT, SURF o BRIEF.

En las siguientes figuras se muestra una prueba de concepto implementada en un dispositivo móvil para la identificación de objetos utilizando los algoritmos antes mencionados (FAST + FREAK).



**Figura 5.11 –Prueba de identificación del edificio de Electrónica del CITEDEF utilizando FAST + FREAK  
(fuente: propia)**

## 5.3.2 Capa: Librerías

Por otro lado en la capa de *Librerías* se han integrado al framework propuesto un conjunto de librerías entre las que se destacan:

- *OpenCV*: librería de procesamiento de imágenes y visión por computador.
- *Mapsforge*: ofrece una librería libre, offline y de código abierto para el sistema operativo Android. El objetivo es la creación de aplicaciones basadas en OpenStreetMap<sup>6</sup> (OpenStreetMap es un proyecto perteneciente a la comunidad open source que permite crear y editar mapas de manera libre). Las herramientas y APIs proporcionadas incluyen soluciones para representación en mapas, planificación y navegación de rutas, indexado y búsqueda de POI, capas en los mapas entre otras opciones. También soportan capas (POIs y polígonos arbitrarios).
- *Libstreaming*: es una API que permite realizar *stream* de la cámara y/o del micrófono para aplicaciones que se ejecutan en el sistema operativo Android utilizando el protocolo RTP<sup>7</sup>.
- *Socket.io*<sup>8</sup>: permite la comunicación bidireccional en tiempo real basada en eventos. Funciona en todas las plataformas, navegadores o dispositivos, centrándose igualmente en la fiabilidad y la velocidad.
- *ZeroMQ*<sup>9</sup>: esta biblioteca utiliza tecnología de comunicación sobre sockets TCP/IP. Además es un sistema completo de colas de mensaje, concurrencia, balanceo de cargas, enrutamiento muy eficaz y rápido que abre la oportunidad para crear aplicaciones de cómputo distribuido. Es una tecnología GPL y de código abierto.

---

<sup>6</sup> OpenStreetMap: <https://www.openstreetmap.org/>

<sup>7</sup> RTP: Real-time Transport Protocol. Es un protocolo de nivel de sesión utilizado para la transmisión de información en tiempo real) sobre UDP.

<sup>8</sup> Socket.io: <https://socket.io/>

<sup>9</sup> ZeroMQ: <http://zeromq.org/>

- *PocketSphinx*<sup>10</sup>: es un motor ligero de reconocimiento de voz, específicamente adaptado para dispositivos de mano y dispositivos móviles.

## 5.4 Arquitectura del Sistema. Componentes de hardware utilizados

La arquitectura del sistema propuesta está representada por un conjunto de componentes de hardware en dónde se ejecutarán las aplicaciones desarrolladas que extenderán el funcionamiento del framework propuesto. Estos componentes de hardware principales de la arquitectura del sistema está compuesta por:

- *Smart Glasses Moverio BT-300 Smart Glasses*: gafas de RA donde integra sensores inerciales, GPS, brújula magnética, luminosidad. Estas gafas oficiarán de prototipo para la proyección y visualización de los elementos digitales desarrollados en las pruebas de concepto.
- *Cardboard box*: soporte para el dispositivo móvil que se encargará de proyectar y visualizar las imágenes sobreimpresas. Adicionalmente, de los dispositivos móviles se utilizarán los sensores que trae incorporado en la mayoría de los modelos, tales como, sensores inerciales (acelerómetros y giróscopos), GPS, brújula magnética, entre los principales. Dichos dispositivos oficiarán de pantalla y, adicionalmente como parte de la arquitectura de componentes de hardware propuesta, se utilizarán los diferentes sensores IMU.
- *Procesador/es externo (ODROID-XU3)*: este dispositivo se encargará de procesar las imágenes capturadas por la cámara. El resultado es enviado al dispositivo móvil (smartphone o smart glasses) para su proyección y visualización.
- *Cámara HD*: este componente capturará las escenas exteriores. Posteriormente las imágenes obtenidas serán procesadas por los dispositivos ODROID XU3.
- *Micrófono*: se utilizará para la entrada de comandos vocales mediante la utilización de reconocimiento de voz.

---

<sup>10</sup> PocketSphinx: <https://github.com/cmuspinx/pocketsphinx>

- *Sensores externos*: estos componentes son provistos por el Ejército y básicamente son todos aquellos dispositivos que toman datos del contexto (UAV, Cámaras, Magnéticos, etc.) y los datos son enviados al CII para su procesamiento.
- *Medio de comunicación*: en los prototipos desarrollados (aplicaciones que utilizan el framework diseñado) se utilizaron antenas Wireless para transmitir datos desde el CII y el JPat.

A continuación en la siguiente figura se representa la Arquitectura del sistema sugerida en su totalidad. Se detallan los principales componente de hardware propuesto:



**Figura 5.12 – Arquitectura del Sistema de RA propuesto  
(fuente: propia)**

El propósito del presente trabajo es centrarse en el diseño del framework para que las aplicaciones desarrolladas sean ejecutadas en dispositivos móviles con sistema operativo Android y en procesadores con arquitectura ARM. El framework deberá estar diseñado de manera tal que pueda paralelizar funciones de visión por computador en procesadores ARM distribuidos en la arquitectura de sistemas propuesto.

Cabe aclarar que en esta primera etapa se evaluó el uso de componentes de hardware del tipo COTS, ya que los mismos se encuentran muy probados por los usuarios finales y los costos de adquisición son muy inferiores a los componentes de hardware militares.

La arquitectura del sistema está pensada para que las aplicaciones que utilicen el framework propuesto en el presente trabajo puedan ejecutarse tanto en gafas de RA como también en dispositivos móviles tales como smartphone, tablets, etc.

Siguiendo la arquitectura planteada en la sección 3.2.3 - *Proyecto iARM* y en la arquitectura de sistemas detallado en la sección 3.2.4 - *Proyecto ULTRA-Vis*, se decidió desarrollar una arquitectura similar pero con procesamiento distribuido con el objetivo de optimizar las tareas de procesamiento de imágenes.



**Figura 5.13 – Diseño de las capacidades del framework propuesto  
(fuente: propia)**

Esta arquitectura responde a las necesidades de derivar carga de trabajo intensivo a procesadores externos bajo una arquitectura distribuida para, por ejemplo, procesar los algoritmos necesarios de visión por computador y devolver el resultado al dispositivo móvil que se

encargará de proyectar el resultado correspondiente. La razón principal de enviar parte del procesamiento a los procesadores externos distribuyendo la carga se debe a que los componentes móviles cuentan con procesadores de bajo poder de cómputo con lo cual dificulta obtener resultados al procesar gran volumen de información en un tiempo razonable.



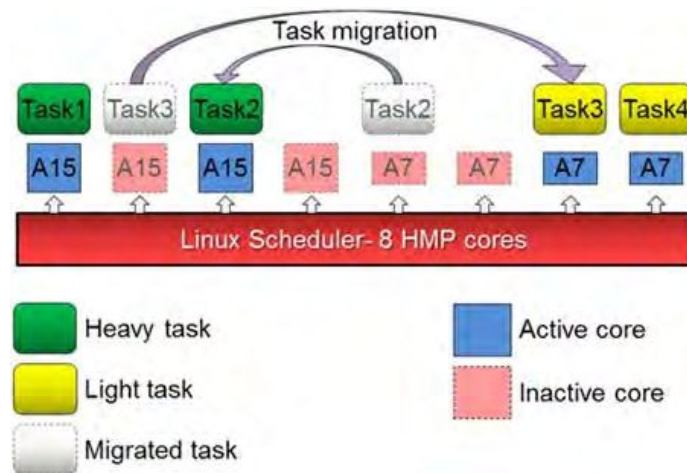
**Figura 5.14 – Ejemplo de la Arquitectura propuesta utilizando gafas de RA (1) y ODROID-XU (2)  
(fuente: propia)**

Por lo tanto, para ejecutar el procesamiento distribuido se optó por los procesadores ODROID-XU3 de la empresa *Hardkernel co., Ltd* de Corea del Sur. Estos componentes de hardware fueron los seleccionados para formar parte de la arquitectura integral del sistema y cuya función será la de procesar externamente imágenes y otras funciones. Se escogió este procesador por su alto poder de cómputo y por las dimensiones pequeñas del componente lo que era fácil integrarlo, por ejemplo, a un casco militar.

La arquitectura del ODROID-XU3 presenta un bloque multimedia soportado por una poderosa GPU. El GPU es un componente de hardware con características multi-core que puede paralelizar cálculos en dicha unidad de proceso gráfico (GPGPU). En la actualidad muchos móviles modernos poseen GPU en el que se aprovecha el poder procesamiento paralelo.



ODROID-XU3 es una nueva generación de dispositivos poderosos, eficientes en el consumo de energía y factor de forma pequeños. Una característica interesante es que este procesador tiene soporte para ejecutar diferentes sistemas operativos tales como Ubuntu y Android versión 4.4.x. y superior. Está equipado con 4 núcleos grandes (Samsung Exynos-5422 ARM Cortex-A15 hasta 2.0 GHz) y 4 núcleos pequeños (ARM Cortex-A7 hasta 1.0 GHz). Posee capacidades de procesamiento mejoradas a bajo consumo de energía. Como parte de la arquitectura tienen una solución denominada big.LITTLE HMP (Heterogeneous Multi-Processing (HMP) solution) lo que permite que los procesadores Exynos-5422 puedan utilizar los 8 cores para administrar computacionalmente las tareas intensivas.



**Figura 5.15 – ODROID XU3. Solución Heterogeneous Multi-Processing (HMP) solution**

(fuente: Hardkernel [http://www.hardkernel.com/main/products/prdt\\_info.php?g\\_code=G143452239825](http://www.hardkernel.com/main/products/prdt_info.php?g_code=G143452239825) )

Una de las características principales del ODROID-XU3, además de su poder de procesamiento, es que posee un pequeño factor de forma reduciendo de manera considerable el tamaño del dispositivo. Las dimensiones de la placa de circuito impreso son de 94x70x18mm, lo que permite adaptarlo a un casco militar o dispositivo de transporte externo sin ocupar demasiado espacio ni peso.





**Figura 5.16 – ODROID XU3**

**(fuente: Hardkernel [http://www.hardkernel.com/main/products/prdt\\_info.php?g\\_code=G143452239825](http://www.hardkernel.com/main/products/prdt_info.php?g_code=G143452239825) )**

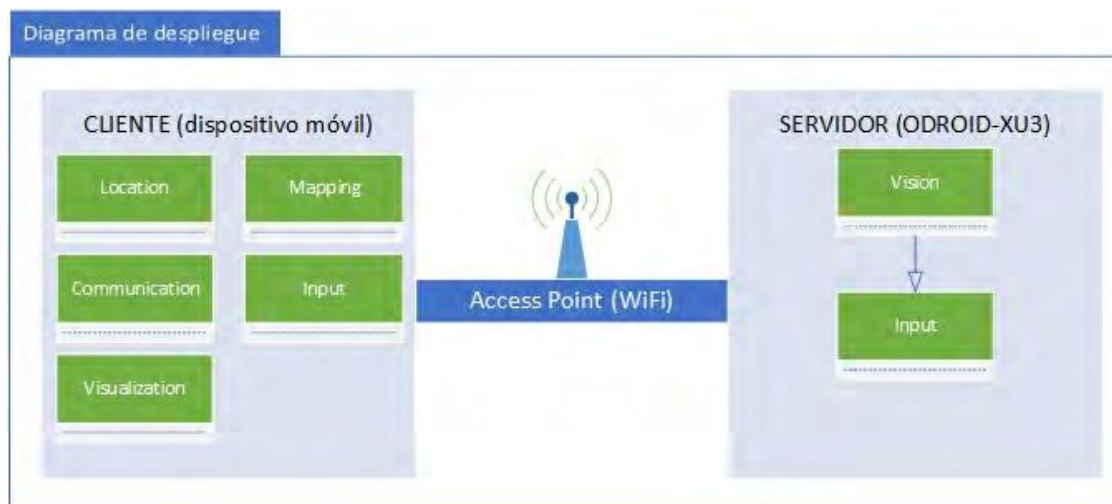
El ODROID XU3, posee el GPU ARM Mali-T628 MP6 y ofrece soporte para OpenGL ES 1.1, OpenGL ES 2.0 y OpenGL ES 3.0, OpenCL 1.1 como también Google RenderScript. El GPU Mali-T628 está optimizado para procesar aplicaciones gráficas en 3D, computación visual, RA, generación de texturas y reconocimiento de voz, entre otros.

## **5.5 Proceso distribuido**

Como parte de la arquitectura del sistema propuesta y siguiendo los niveles de procesamiento de RA sugeridos por [14], el proceso distribuido planteado está soportado por el diseño de una arquitectura Cliente / Servidor.

Del lado del Cliente se ejecutan los procesos menos intensivos tales como:

- Captura de video
- Uso de sensores (ubicación, tracking)
- Comunicación
- Mapeo
- Renderizado de imágenes



**Figura 5.17 – Diagrama de despliegue**  
(fuente: propia)

Mientras que del lado del Servidor se implementó el módulo de *Vision* (y algunas funcionalidades del módulo *Input* de entrada de datos) ya que en este módulo se instauró el procesamiento de imágenes pues el ODROID-XU3 soporta cargas intensivas de cómputo permitiendo distribuir dicha carga sobre sus procesadores.

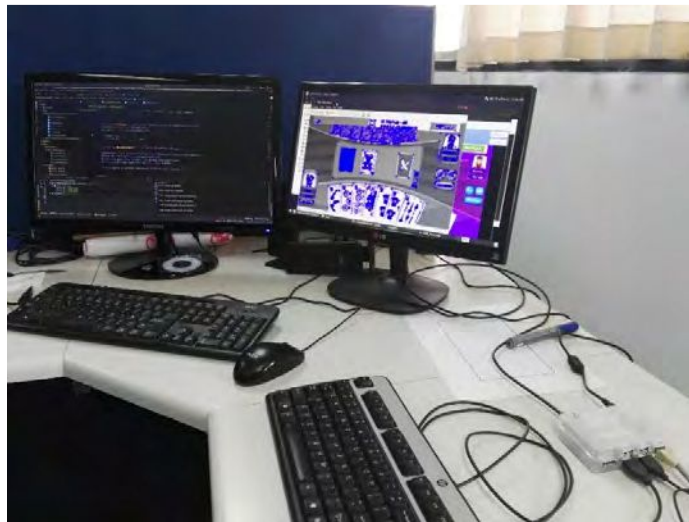
Con el objetivo de implementar el procesamiento distribuido se evaluó la comunicación de datos entre un *Cliente* (implementado en JAVA-Android) y un *Servidor* (implementado en JavaScript-Node.js), utilizando Socket.io como tecnología de comunicación entre estos componentes. Cabe aclarar que en el lado del Servidor se ejecutarán los algoritmos pertenecientes al módulo de *Vision*. El funcionamiento entre el Cliente y el Servidor se ejecuta mediante el envío de un *frame* (imagen) desde la aplicación Cliente hacia la aplicación Servidor. Para lograr esta acción se codifica en formato JSON de la siguiente manera:

```
private void sendImage(Uri uri) throws JSONException, FileNotFoundException {
    JSONObject data = new JSONObject();
    data.put("image", encodeImage(uri));

    mSocket.emit("new message", data);
}
```

El servidor recibe el mensaje con la imagen. Guarda el archivo en el *filesystem* y luego crea un proceso donde se ejecuta el algoritmo (implementado en Python) para el procesamiento de la imagen. En este caso se utiliza el algoritmo FAST. El resultado se guarda como una imagen y por otro lado, la respuesta (en formato JSON) es enviada al Cliente.

Los componentes de software instalados en el lado de Cliente fueron Android 6.0, OpenCV 3.1, Socket.io y librerías (ver sección 5.3.2 - *Capa: Librerías*); mientras que en la lado Servidor se instaló Ubuntu Mate versión 16.04, Python versión 3.0, OpenCV 3.1 y Node.js.



**Figura 5.18 – Extrayendo características de una imagen mediante procesamiento distribuido utilizando un ODROID-XU3 y una PC que simula a un dispositivo móvil (fuente: propia)**

# Capítulo 6

## Despliegue de la solución

### 6.1 Introducción

En este capítulo se ha trabajado el despliegue de la solución propuesta. Para ello se han seleccionado e integrado distintas tecnologías, nodos de procesamiento y facilidades de comunicación e interacción necesarias para poder utilizar la tecnología descrita en el capítulo anterior.

Los despliegues propuestos en este capítulo tienen en cuenta las distintas exigencias, tanto funcionales como no funcionales, identificadas en capítulos previos, especialmente aquellas recogidas en el capítulo quinto y que tienen que ver con la CS, la RA y la usabilidad general de los desarrollos y despliegues propuestos en este informe. En este sentido, en este capítulo se remarcan las facilidades logradas y relacionadas con:

- Identificación de la localización del usuario,
- Identificación del entorno (construcciones edilicias, aliados, enemigos, etc.),
- Facilidades de operación, considerando técnicas interactivas basadas en gestos,
- Compatibilidad e interoperabilidad de dispositivos.

El objetivo de este capítulo es demostrar la viabilidad de la solución propuesta y de su distribución.

### 6.2 Dispositivos utilizados

A la hora de realizar el despliegue de la misma se ha trabajado inicialmente en la selección de los dispositivos, las características comunes entre ellos serían nodos de procesamiento móviles, basados en Android y con distintas características. Concretamente, los dispositivos seleccionados se recogen en la siguiente tabla:

Tabla 6.1 Dispositivos móviles. Características técnicas.

Dispositivo	Cámara	SO	CPU	GPU	Conectividad	Sensores
Epson Move- rio BT-300	5 Mp	Android 5.1 API Le- vel 22	Intel® Cherry Trail Atom™ x5 1.44GHz Quad Core	Intel HD Graphics (Cherry Trail)	GPS	Geomagnetic sensor/accel- erometer/gyroscopic sen- sor/illumination sensor
Hauwei G8	Primaria resolu- ción 13 Mp – 30 fps (a resolución máxima)	Android 5.1.1 Lo- llipop	Qualcomm Snapdragon 616 MSM8939v2 4 x 1.5 GHz ARM Cor- tex-A53 4 x 1.2 GHz ARM Cor- tex-A53	Qualcomm Adreno 405	GPS con soporte A- GPS y GLONASS. 4G/LTE. WiFi (b, g, n). Bluetooth 4.0	Acelerómetro, giróscopo (emulado), proximidad, brújula, luz ambiente, lec- tor de huellas dactilares
Samsung SM- G930F	Principal de 12 megapíxeles con apertura f/1.7, tec- nología Dual Pixel Sensor, OIS y dual-LED. Secun- daria de 5 megapí- xeles con apertura f/1.7	Android 6.0.1 Marsh- mallow con Samsung Touch- Wiz	Samsung: 4 núcleos Exynos M1 Mongoose a 2,6/2,3 GHz + 4 nú- cleos ARM Cortex-A53 a 1,5 GHz (GTS)	Samsung: Mali-T880 MP12 a 650 MHz	4G LTE Cat.12/13, Wi-Fi doble banda, Wi-Fi Direct, Blue- tooth 4.2, NFC, GPS con A-GPS y GLONASS, Micro- USB 2.0, USB Host, jack de 3,5 mm	Acelerómetro, giróscopo, proximidad, brújula, luz ambiente, barómetro
ODROID XU3	Externa. Logitech	Android 5.1.1 Lo- llipop	Samsung Exynos 5422 4 x 2.0 GHz ARM Cor- tex-A15 4 x 1.4 GHz ARM Cor- tex-A7 Total de núcleos: 8	Mali-T628 MP6 (OpenGL ES 3.0/2.0/1.1 and OpenCL 1.1 Full pro- file)	Módulo USB externo GPS. Módulo USB externo WiFi (b, g, n). Módulo USB ex- terno Bluetooth 4.0	No posee



**Figura 6.1 – Dispositivos móviles utilizados para en los despliegues**  
**(Por orden de aparición: Epson Moverio BT 300, Hauwei G8, Samsung SM-930F,**  
**ODROID XU3)**  
**(fuente: varias)**

En la figura anterior se muestran los dispositivos móviles que se han seleccionado para el despliegue. Los cuatro móviles se han usado como dispositivos de adquisición de video, utilización de sensores y visualización de imágenes en la pantalla de los mismos, en el caso de la gafa *Epson Moverio BT 300* la proyección del video se realiza sobre el lente mientras que en los smartphone se utiliza el *cardbox* como adaptador para la proyección y visualización.

### 6.3 Despliegues considerados

Las arquitecturas hardware utilizadas para realizar el despliegue se centraron en dos tipos, por un lado, *optical see-through* mediante la utilización de la gafa de RA y un mini board ODROID-XU3 como servidor para el procesamiento de imágenes y, por el otro, *video see-through* mediante el uso de un *glass cardbox* como soporte a los diferentes smartphone propuestos más un mini board ODROID-XU3 con el mismo fin propuesto en la arquitectura anteriormente mencionada.

A continuación, se detallan los dos despliegues que se han realizado y que serán utilizados en el próximo capítulo para realizar la validación de la solución propuesta:

Tabla 6.2 Despliegues utilizados para validar la solución propuesta.

Opción 1	Opción 2
<ul style="list-style-type: none"> <li>• Componentes de hardware:               <ul style="list-style-type: none"> <li>○ Gafa de RA                   <ul style="list-style-type: none"> <li>▪ Antena WiFi</li> <li>▪ Cámara HD</li> </ul> </li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>• Componentes de hardware:               <ul style="list-style-type: none"> <li>○ Glass cardboard</li> <li>○ Smartphone                   <ul style="list-style-type: none"> <li>▪ Antena WiFi</li> </ul> </li> </ul> </li> </ul>

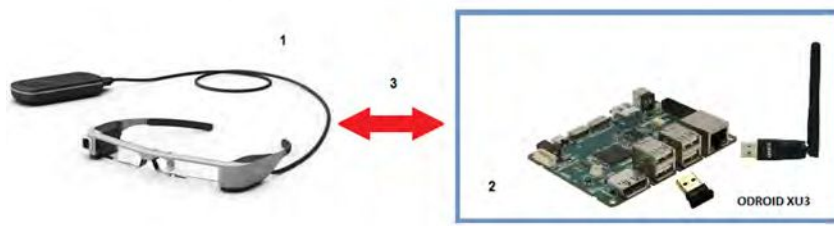
<ul style="list-style-type: none"> <li>▪ Sensores: Sensores: GPS, acelerómetro, giróscopo, brújula magnética</li> <li>○ <i>ODROID XU3</i> <ul style="list-style-type: none"> <li>▪ Antena Bluetooth</li> <li>▪ Antena WiFi</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>▪ Cámara HD</li> <li>▪ Sensores: GPS, acelerómetro, giróscopo, brújula magnética</li> <li>○ <i>ODROID XU3</i> <ul style="list-style-type: none"> <li>▪ Antena Bluetooth</li> <li>▪ Antena WiFi</li> </ul> </li> </ul>
<p>El ciclo de captura de video se procesa de la siguiente manera:</p> <ol style="list-style-type: none"> <li>1) El video es capturado desde la cámara de la gafa de RA</li> <li>2) Los frames del video capturado desde la cámara de la gafa de RA son enviados al <i>mini board ODROID XU3</i></li> </ol> <p>Se utilizó el módulo de <i>Communication</i> del framework que utiliza <i>Socket.io</i> como tecnología de comunicación entre la gafa de RA y el <i>mini board ODROID XU3</i>.</p>	<p>El ciclo de captura de video se procesa de la siguiente manera:</p> <ol style="list-style-type: none"> <li>1) El video es capturado desde la cámara del Smartphone</li> <li>2) Los frames del video capturado desde la cámara del smartphone son enviados al <i>mini board ODROID XU3</i></li> </ol> <p>Se utilizó el módulo de <i>Communication</i> del framework que utiliza <i>Socket.io</i> como tecnología de comunicación entre el <i>Smartphone</i> y el <i>mini board</i>.</p>

Los smartphone seleccionados se acoplaron al *glass cardboard* emulando la técnica de *video see-through*:

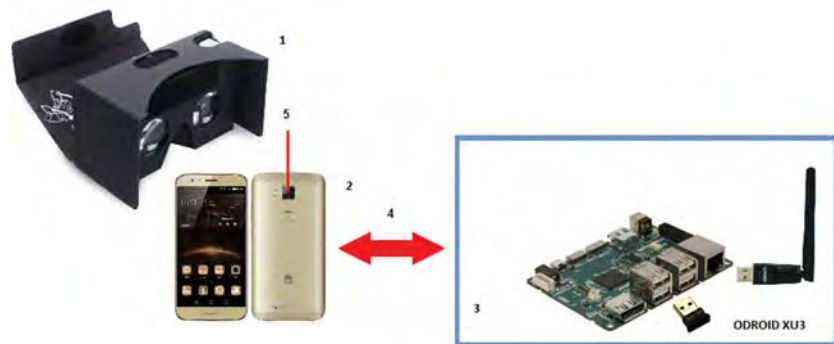


**Figura 6.2 – *Glass cardboard* acoplado a los smartphone utilizado en los despliegues (fuente: propia)**

A continuación se ilustran las dos opciones de arquitecturas propuestas:



**Opción 1: Optical see-through. 1. Gafa de RA EPSON MOVERIO BT-300; 2. ODROID XU3; 3. Comunicación entre la Gafa de RA y el ODROID XU3**



**Opción 2: Video see-through. 1. Glass cardboard; 2. Smartphone; 3. ODROID XU3; 4. Comunicación entre los ODROID XU3; 5. Smartphone utilizando la cámara HD.**

**Figura 6.3 – Opciones de arquitecturas de sistemas propuestos (fuente: varias)**

Una vez mencionadas las componentes hardware utilizadas para el despliegue de la solución, se está en condiciones de detallar los prototipos evolutivos como parte de la metodología sugerida y la implementación de las funcionalidades obtenidas para dar soporte a facilidades relacionadas con la CS.

Además de la infraestructura descrita se utilizó un *access point* dentro de un mismo segmento de red (LAN) desde donde se conectaban los dispositivos móviles.

Seguidamente, se analiza, de manera más detenida, algunos aspectos que se consideran fundamentales para el éxito de la solución propuesta. Concretamente se discute el despliegue de los módulos de comunicaciones, de visión y de integración de la tecnología.



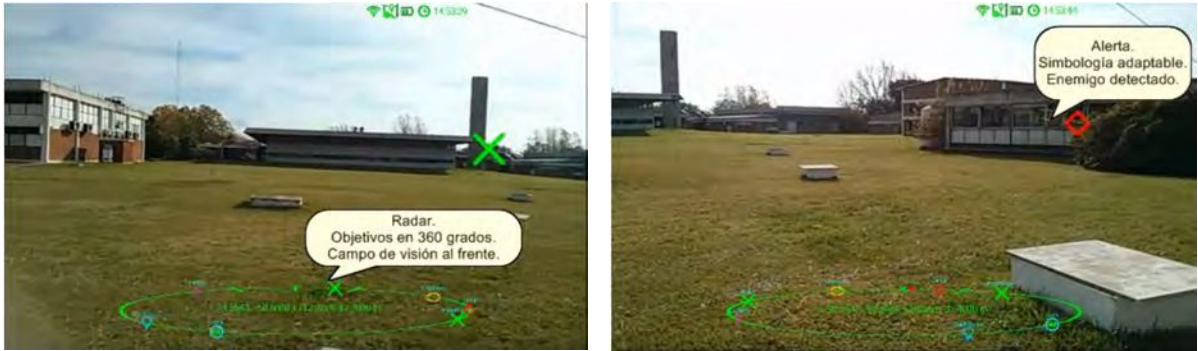
## 6.4 Módulo *Vision*

En el tercer prototipo desarrollado se integraron los algoritmos de extracción de características en el módulo de *Vision* del framework. Para ello se utilizó la librería *OpenCV* y algunas estrategias de optimización. El algoritmo de detección de características utilizado fue SIFT. Se entrenaron cinco construcciones edilicias, de las cuales se extrajeron características las que fueron almacenadas en una base de datos. A medida que se iba recorriendo las instalaciones edilicias, se visualizaba una imagen representativa donde visualmente se daba a conocer al operador de la aplicación que dicho objeto se estaba procesando, tanto el reconocimiento como la clasificación de dicho objeto. Luego la aplicación imprimía en la parte superior izquierda de la pantalla, el texto del objeto reconocido y clasificado.



**Figura 6.4 – Prueba de extracción de características utilizando SIFT. Reconocimiento y Clasificación de objetos edilicios (fuente: propia)**

En el despliegue realizado se integró una aplicación de visualización de alertas y amenazas mediante el uso de iconografía y simbología militar. Para ello se implementó en la aplicación una función de visualización donde se imprime en el campo de visión del operador la información contextual obtenida remotamente desde una unidad de comunicación. El objetivo es potenciar las facilidades de la CS.



**Figura 6.5 – Visualización de iconografía y radar de 360 grados con información contextual impresa**  
(fuente: propia)

Adicionalmente, se integró el framework propuesto en una aplicación para el manejo de menús a través de reconocimiento gestual. Por ejemplo, en la siguiente figura se puede apreciar como el dedo índice de la mano izquierda del operador controla la activación y desactivación de un mapa digital. El objetivo es potenciar la CS y la usabilidad de la solución desarrollada.



**Figura 6.6 – Activación de un mapa por medio de reconocimiento gestual**  
(fuente: propia)

## 6.5 Integración de tecnologías

Se integró una aplicación software para que sea ejecutada en cada uno de los dispositivos propuestos utilizando la arquitectura de sistema mencionada en la *Sección 6.3 - Despliegues*.

Entre las funcionalidades desarrolladas en la aplicación y extendidas del framework se implementaron las siguientes características:

- *Visualización de objetivos tácticos en un radar en 360 grados*: los objetivos tácticos son cargados en la aplicación para que sean visualizados en la misma mediante la iconografía,

que por defecto, se diseñó. Los objetivos representan coordenadas latitud y longitud que son interpretados por el paquete *Location* del framework. Luego, los símbolos tácticos son impresos en el radar en 360 grados, permitiendo al operador tener una “visión” perimetral de las amenazas u objetivos tácticos.

- *Posicionamiento y orientación espacial de los objetos y del operador*: el norte magnético sirve de referencia para ubicar y orientar al operador. Mientras el operador se desplaza, se calcula su posición relativa con respecto al norte magnético. De la misma manera que en la funcionalidad anterior, se utiliza el paquete *Location* a fin de calcular la ubicación y orientación del usuario.
- *Ayudas visuales mediante iconografía/simbología adaptable*: la iconografía o símbolos tácticos utilizados en el radar de 360 grados, son adaptables y configurables permitiendo identificar la/s amenaza/s u objetivo/s táctico/s de acuerdo a la simbología militar, en este proyecto en particular. No obstante, se puede utilizar otra simbología, como por ejemplo, iconos logísticos u otros que sirvan de referencia para el operador.
- *Uso de menús interactivos mediante Reconocimiento Gestual*: mediante la utilización de técnicas de reconocimiento gestual se puede obtener información a demanda. Esta funcionalidad se implementó en el paquete *Input* del framework. Si bien esta funcionalidad posee características de procesamiento de imágenes y visión por computador, no fue implementado en el paquete de *Vision* ya que se analizó dentro de la lógica del framework que era una mejor opción de diseño implementarlo en el paquete *Input*.
- *Reconocimientos de objetos edificios*: como parte principal de esta prueba de concepto se implementó en la aplicación desarrollada la posibilidad de reconocer varios objetos edificios previamente entrenados. Esta funcionalidad fue implementada en el módulo de *Vision*. No obstante, para reconocer el primer frame del objeto a clasificar, se utilizó la técnica de reconocimiento gestual donde, a través de la detección del dedo anular y del dedo índice del operador, se especifica el área aproximada de captura del primer frame. A partir de esta detección inicial, el siguiente paso es la clasificación del objeto en cuestión utilizando las técnicas de reconocimiento de objetos adecuada implementadas en el framework y extendida en la aplicación de prueba.

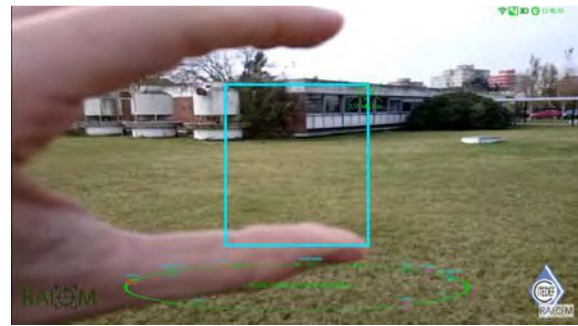
- *Uso de cartografía en 3D mediante uso de RA:* esta funcionalidad se encuentra implementada en el módulo de *Mapeo*. Este módulo implementa una interfaz a un SIG completamente funcional que permite cargar cartas prediseñadas a la aplicación y ubicar elementos para su posterior visualización y manipulación ya sea en una carta 2D como en un mundo 3D.

Básicamente se integraron en una aplicación todas las funcionalidades demostradas en cada uno de los prototipos desarrollados anteriormente.

En las siguientes figuras se puede observar cada una de las características implementadas en la aplicación utilizando el framework propuesto:



a) Radar 360 grados



b) Reconocimiento de objeto (Edificio)



c) Reconocimiento gestual e iconografía



d) Visualización de mapa activado por gesto

**Figura 6.7 – QRCode para acceder a vídeo en el cual se muestra la integración de las funcionalidades principales del framework**

(fuente: propia)



# Capítulo 7

## Conclusiones y Líneas de trabajo futuras

### 7.1 Conclusión del Trabajo

En el presente trabajo final, se ha profundizado en los fundamentos de procesamiento de imágenes y visión por computador dándole principal importancia a los aspectos técnicos relacionados con la Realidad Aumentada.

Se ha analizado diferentes proyectos militares que utilizan tecnología de visión artificial y RA en la ayuda de la toma de decisiones en terrenos hostiles. Se han listado y comparado diferentes proyectos militares que utilizan estas tecnologías para dar soporte al personal militar. Luego de haber analizado estas tecnologías se construyó una tabla comparativa a fin de determinar la necesidad concreta de construir un framework propio de RA que sea aplicado en la construcción de aplicaciones. Por otro lado, se analizaron los elementos necesarios que forman parte de tener una CS amplia para tener el estado de situación de las amenazas circundantes y, por el otro lado, las tecnologías que dan soporte para que se cumpla lo antes mencionado. De este análisis se concluyó que no todas las tecnologías cumplen con los requerimientos operacionales, es decir, ninguno de ellos es completo en cuanto al aporte de funcionalidades que sirvan para hacer frente a la problemática de ayudar al personal militar a tomar decisiones en casos de estrés. Por lo tanto, se optó por desarrollar un framework de RA, centrado en resolver la problemática de la CS con el objetivo de poder desarrollar aplicaciones de RA que se ejecuten en una arquitectura Cliente / Servidor. En resumen, se realizó un framework propio para que soporte lo que la mayoría de las tecnologías militares mencionadas no soportan.

Para solucionar la carencia de frameworks que utilicen componentes reutilizables para desarrollar aplicaciones de RA, se consideró una arquitectura basada en capas para el diseño del middleware del framework propuesto que facilita la integración tecnológica. Adicionalmente una de las características relevantes del framework fue la arquitectura Cliente / Servidor, es decir, la capacidad de implementar (desplegar) las aplicaciones que se desarrollen en una arquitectura distribuida

en donde en el lado del Servidor se ejecutarán las tareas más intensivas que corresponden al procesamiento de imágenes.

Para llevar a cabo el proceso de construcción del framework se basó en el desarrollo de pruebas de concepto individuales a modo de prototipos con el fin de que se demuestre el funcionamiento y el cumplimiento de lo solicitado. Para ello el diseño del framework y el desarrollo de las aplicaciones se basaron en una metodología combinada entre el uso de un método ágil y las técnicas de DCU. Por otro lado, se escogió este método de trabajo debido a que el segmento militar era poco explorado y era necesario diseñar, programar y mostrar a los usuarios claves el funcionamiento de cada parte del framework como unidades independientes, que posteriormente, se integrarían en el framework final. Se optó por utilizar SCRUM como método ágil de desarrollo de software en conjunto con los principios de diseño que aporta el DCU resultando una metodología combinada SCRUM-DCU. Siguiendo el criterio de SCRUM-DCU, una de las principales tareas es la realización de prototipos evolutivos.

El principal trabajo fue diseñar la arquitectura de la capa intermedia, es decir, el middleware que está compuesto por la capa de los Módulos y la capa de las Librerías. Se centró en el diseño del framework para que las aplicaciones desarrolladas sean ejecutadas en dispositivos móviles con SO Android y en procesadores con arquitectura ARM. Por otro lado se logró implementar una arquitectura Cliente / Servidor en donde se derivó, del lado del Servidor, la carga de trabajo más intensivo, por ejemplo, procesar los algoritmos necesarios de visión por computador y devolver el resultado al dispositivo móvil que se encargará de proyectar el resultado correspondiente.

Una vez diseñado el framework con los módulos correspondientes se procedió a realizar diferentes Despliegues a fin de poner a prueba lo propuesto. Para ello se han seleccionado e integrado distintas tecnologías, nodos de procesamiento y facilidades de comunicación e interacción necesarias para poder utilizar la tecnología sugerida. Se utilizaron para tal fin hardware que diera soporte a técnicas basadas en *optical see-through* mediante el uso de gafas de RA y *video see-through* mediante el uso de *glass cardbox* y en los 2 casos complementados por mini board ODROID-XU3.

Se trabajó intensamente sobre el Módulo de Visión en donde se encontraban los principales algoritmos de procesamiento de imágenes.

Se logró implementar funcionalidades con las siguientes características:

- Visualización de objetivos tácticos en un radar en 360 grados
- Posicionamiento y orientación espacial de los objetos y del operador

- Ayudas visuales mediante iconografía/simbología adaptable
- Uso de menús interactivos mediante Reconocimiento Gestual y Vocal
- Reconocimientos de objetos edilicios
- Uso de cartografía en 3D mediante uso de RA

Básicamente se integraron en una aplicación todas las funcionalidades demostradas en cada uno de los prototipos desarrollados.

Por lo tanto y a modo de conclusión final se demostró la viabilidad de la solución propuesta cumpliendo el objetivo principal de aportar un soporte a la toma de decisiones.

## 7.2 Líneas futuras de trabajo

Como línea de trabajo futura, se propone la posibilidad de desarrollar un conjunto de funcionalidades que no fueron implementadas en el framework propuesto ya sea por tiempos y/o por el alcance definido. Estas funcionalidades están relacionadas con potenciar el uso de la CS y brindar un soporte más amplios y efectivo en los procesos de toma de decisiones.

Estas funcionalidades adicionales permitirán ampliar el uso pues las aplicaciones que extiendan del framework estarán preparadas para ser utilizadas en misiones en donde, por ejemplo, la falta de luz visible es un factor clave en las operaciones nocturnas.

Por otro lado, la utilización de mecanismos más efectivos de clasificación conlleva a pensar en nuevas herramientas que sean incorporadas en la estructura middleware del framework.

Por lo tanto, y de acuerdo a lo expresado anteriormente, existen diversas líneas de investigación que quedan abiertas luego de la finalización del informe, a saber:

- Diseñar un modelo que sirva para la identificación, selección y clasificación de requisitos de la CS.
- Evaluar el cumplimiento de la CS mediante el uso de distintas técnicas de evaluación donde se contemple la experiencia del usuario.
- Capacidad de reconocimiento de objetos en modalidad nocturno: esta funcionalidad implicaría desarrollar un Módulo para operaciones nocturna implementando algoritmos específicos en el módulo de *Vision*. De esta manera se ampliará la capacidad del módulo lo que



posibilitará desarrollar aplicaciones extendiendo del framework esta característica. Se podrán desarrollar aplicaciones que requieran la utilización de reconocimiento de objetos bajo condiciones de poca luz o la carencia total de ésta.

- Deep Learning (DL): se propone la implementación de técnicas de clasificación basado en la utilización de redes neuronales implementada en el módulo de *Vision* o en su defecto en un módulo específico de DL diseñado separadamente. La idea principal es sustituir los algoritmos de clasificación utilizados en el framework actual por técnicas de DL tales como Convolutional Neural Network (CNN) para el procesamiento de imágenes. Se propone CNN ya que debido a que su aplicación es realizada en matrices bidimensionales, son muy efectivas para tareas de visión artificial, como en la clasificación y segmentación de imágenes, entre otras aplicaciones.

# Bibliografía

- [1] D. W. Brown, «A Survey of Mobile Augmented Reality Technologies for Combat Identification Applications,» USA, 2012.
- [2] M. A. Livingston, L. J. Rosenblum, S. J. Julier, D. Brown, Y. Baillot, J. E. Swan II, J. L. Gabbard y D. Hix, An Augmented Reality System for Military Operations in Urban Terrain, -, Ed., December 2 -5, Orlando, Florida, USA: Proceedings of Interservice / Industry Training, Simulation & Education Conference (I/ITSEC), 2002.
- [3] Fundación Telefónica, Realidad Aumentada: una nueva lente para ver el mundo, Madrid, España: Editorial Ariel, S.A., 2011. Avda. Diagonal, 662-664. 08034 Barcelona (España), 2011.
- [4] J. Carmigniani y B. Furht, Augmented Reality: An Overview, Department of Computer and Electrical Engineering and Computer Sciences, Florida Atlantic University, Boca Raton, Florida, USA: B. Furht (ed.), Handbook of Augmented Reality, DOI 10.1007/978-1-4614-0064-6 1, © Springer Science+Business Media, LLC 2011, 2011.
- [5] M. A. Livingston, S. J. Julier y D. G. Brown, Situation Awareness for Teams of Dismounted Warfighters and Unmanned Vehicles, -, Ed., Naval Research Laboratory. Washington, DC, USA: SPIE Defense and Security Symposium. Enhanced and Synthetic Vision Conference., 2006.
- [6] J. Hicks, R. Flanagan, P. Dr. Petrov y A. Dr. Stoyen, «Eyekon: Distributed Augmented Reality for Soldier Teams,» 704 Edgewood Blvd, Papillion, NE, 68046, 2003.
- [7] T. Gjosaeter, «Interaction with mobile augmented reality,» Dissertation for the degree philosophiae doctor (PhD) at the University of Bergen, Oslo, 2015.
- [8] S. Sestito, S. Julier, M. Lanzagorta y L. Rosenblum, «Intelligent Filtering for Augmented Reality,» Sydney, Australia, 2000.
- [9] R. T. Azuma, «A Survey of Augmented Reality,» Teleoperators and Virtual Environments 6, 4, 355-385, -, 1997.
- [10] C. McDonald, Hand Interaction in Augmented Reality, Ottawa, Ontario, Canada: The Ottawa-Carleton Institute for Computer Science. School of Computer Science. Carleton University., 2003.

- [11] W. Le Roux, The use of Augmented Reality in Command and Control Situation Awareness, South Africa: Council for Scientific and Industrial Research, 2010.
- [12] F. M. Jiménez Montilla, Reconocimiento de objetos con Realidad Aumentada, Granada: Universidad de Granada, 2013.
- [13] M. Specht, S. Ternier y W. Greller, Dimensions of Mobile Augmented Reality for Learning: A First Inventory, Open University, the Netherlands: Journal of the Research Center for Educational Technology (RCET) 117. Vol. 7, No. 1, Spring 2011, 2011.
- [14] D. Wagner, «Handheld Augmented Reality,» Graz, Austria, 2007.
- [15] M. J. Abásolo Guerrero, C. Manresa Yee, R. Más Sansó y M. Vénere, Realidad virtual y realidad aumentada. Interfaces avanzadas., La Plata, Buenos Aires, Argentina: Editorial de la Universidad Nacional de La Plata (Edulp). 47 N° 380 / La Plata B1900AJP / Buenos Aires, Argentina, 2011.
- [16] A. MacWilliams, A Decentralized Adaptive Architecture for Ubiquitous Augmented Reality System, Institut für Informatik der Technischen Universität München, 2005.
- [17] A. MacWilliams, T. Reicher, G. Klinker y B. Bruegge, Design Patterns for Augmented Reality Systems, Institut für Informatik, Technische Universität München: In Proc. International Workshop Exploring the Design and Engineering of Mixed Reality Systems - MIXER 2004, 2004.
- [18] M. Kölsch, R. Bane y T. a. T. M. Höllerer, «Multimodal Interaction with a Wearable Augmented Reality,» IEEE Computer Graphics and Applications ( Volume: 26, Issue: 3, May-June 2006 ), 2006.
- [19] D. Datcu y S. Lukosch, «Free-Hands Interaction in Augmented Reality,» New York, NY, USA , SUI '13 Proceedings of the 1st symposium on Spatial user interaction, 2013, pp. Pages 33-40.
- [20] P. Tiefenbacher, S. Wichert, D. Merget y G. Rigoll, «Impact of Coordinate Systems on 3D Manipulations in Mobile Augmented Reality,» ICMI '14 Proceedings of the 16th International Conference on Multimodal Interactio, New York, NY, USA, 2014.
- [21] Y. Pang, M.L. Yuan, A.Y.C. Nee, S.K. On y Kamal Youcef-Toumi, A Markerless Registration Method for Augmented Reality based on Affine Properties, Australian Computer Society, Inc. Seventh Australasian User Interface Conference (AUIC2006), Hobart, Australia: Conferences in Research and Practice in Information Technology (CRPIT), Vol. 50. Wayne Piekarski, Ed. Reproduction for academic, 2006.

- [22] B. Furht, «Handbook of Augmented Reality,» Springer New York. Editor Affiliations: ID1. Dept. of Computer Science & Engineering, Florida Atlantic University, New York, USA, 2011.
- [23] S. You, U. Neumann y R. Azuma, «Orientation tracking for outdoor augmented reality registration,» IEEE Computer Graphics and Applications (Volume:19 , Issue: 6 ), Los Angeles, CA, USA, 2002.
- [24] B.-K. Seo , J. Park y J.-I. Park, «3-D visual tracking for mobile augmented reality applications,» Multimedia and Expo (ICME), 2011 IEEE International Conference on, Barcelona, Spain, 2011.
- [25] H. Kandil y A. Atwan, «A Comparative Study between SIFT- Particle and SURF-Particle Video Tracking Algorithms,» International Journal of Signal Processing, Image Processing and Pattern Recognition, Egypt, 2012.
- [26] A. Yilmaz, O. Javed y M. Shah, «Object Tracking: A Survey,» School of EECS, University of Central Florida, USA, 2006.
- [27] K. Pulli, A. Baksheev, K. Korniyakov y V. Eruhimov, «Real-time computer vision with OpenCV,» *Magazine Communications of the ACM*, vol. Volume 55 Issue 6, pp. Pages 61-69, 2012.
- [28] G. Bleser, «Towards Visual-Inertial SLAM for Mobile Augmented Reality,» (Thesis) Vom Fachbereich Informatik der Technischen Universität Kaiserslautern, 2009.
- [29] J. M. X. Natário Teixeira, «Analysis and evaluation of optimization techniques for tracking in augmented reality applications,» Recife, Brasil, 2013.
- [30] M. Guerrero, «A Comparative Study of Three Image Matching Algorithms: Sift, Surf, and Fast,» Thesis of MASTER OF SCIENCE in Civil and Environmental Engineering. Utah State University., Logan, Utah, USA, 2011.
- [31] M. Maida, M. Preda y V. H. Le, «Markerless Tracking for Mobile Augmented Reality,» Signal and Image Processing Applications (ICSIPA), 2011 IEEE International Conference on, Kuala Lumpur, 2011.
- [32] D. van Krevelen y R. Poelman, «A Survey of Augmented Reality Technologies, Applications and Limitations,» The International Journal of Virtual Reality, 2010, 9(2):1-20, Delft University of Technology, The Netherlands, 2010.
- [33] R. Azuma, S. You y U. Neumann, «Hybrid Inertial and Vision Tracking for Augmented Reality Registration,» Proceeding, VR '99 Proceedings of the IEEE Virtual Reality, Washington, DC, USA, 1999.

- [34] K. Mikolajczyk y C. Schmid, «A Performance Evaluation of Local Descriptors,» IEEE Transactions on Pattern Analysis and Machine Intelligence (Volume:27 , Issue: 10 ), 2005.
- [35] T. Tuytelaars y K. Mikolajczyk, «Local Invariant Feature Detectors: A Survey,» Foundations and Trends in Computer Graphics and Vision Vol. 3, No. 3 (2007) 177–280, 2008.
- [36] S. Tremblay, P. Jeuniaux, P. Romano, J. Lowe y R. Grenier, «A Multi-Perspective Approach to the Evaluation of a Portable Situation Awareness Support System in a Simulated Infantry Operation,» Miami Beach, FL, USA, 2011.
- [37] M. Gheisari y J. Irizarry, «Investigating Facility Managers' Decision Making Process through a Situation Awareness Approach,» International Journal of Facility Management. Vol 2, Nro. 1, 2011.
- [38] M. R. Endsley, C. A. Bolstad, D. G. Jones y J. M. Riley, «Situation Awareness Oriented Design: From User's Cognitive Requirements to Creating Effective Supporting Technologies,» Denver, Colorado, USA, 2003.
- [39] M. R. Endsley, «Design and evaluation for situation awareness enhancement,» Santa Mónica, CA, USA, 1988.
- [40] M. R. Endsley, «A taxonomy of situation awareness errors,» Aldershot, England, 1995.
- [41] J. G. Morrison, D. A. Kobus y C. M. Brown, «DARPA Improving Warfighter. Information Intake Under Stress. Augmented Cognition,» San Diego, CA, USA, 2006.
- [42] RDECOM, «FUTURE SOLDIER 2030 Initiative», U. A. N. S. R. Center, Ed., USA: -, 2009.
- [43] T. Saarelainen y J. Jormakka, «C4I2-Tools for the Future Battlefield Warriors. », -: IEEE - Fifth International Conference on Digital Telecommunications, 2010.
- [44] M. A. Livingston, S. J. Julier, G. S. Schmidt y D. G. Brown, «Mobile Augmented Reality: Applications and Human Factors Evaluations», A. I. T. C. 5. N. R. Laboratory, Ed., USA: -, 2006.
- [45] J. Timonen y J. Vankka, «Enhancing Situational Awareness by Means of Information Integration of Sensor Networks and Visualization», S. 8756, Ed., Baltimore, Maryland, USA: SPIE 8756, 2013.

- [46] J. Hicks, R. Flanagan, P. Dr. Petrov y A. Dr. Stoyen, *Eyekon: Distributed Augmented Reality for Soldier Teams.* », -, Ed., USA: Copyright 21st Century Systems, Inc., 2003.
- [47] L. J. Rosenblum, S. K. Feiner, S. J. Julier, J. E. Swan II y M. A. Livingston, «The Development of Mobile Augmented Reality,» de *Expanding the Frontiers of Visual Analytics and Visualization*, -, Ed., Expanding the Frontiers of Visual Analytics and Visualization, pp 431-448, Springer London, 2012.
- [48] J. Juhnke, A. Kallish, D. Delaney, K. Dziedzic y R. Chou, «Tanagram Partners. Aiding Complex Decision Making through Augmented Reality: iARM, an Intelligent Augmented Reality Model,» USA, 2010.
- [49] Broad Agency Announcement, «Urban Leader Tactical Response, Awareness & Visualization (ULTRA-Vis),» Broad Agency Announcement (BAA), USA, 2008.
- [50] D. Roberts, A. Menozzi, J. Cook, T. S. S. Sherrilla, P. Russler, B. K. R. Clipp, E. Wenger, M. Bennett, J. Mauger, W. Church, H. Towles, S. MacCabe, J. Webb, J. Lupo, J.-M. Frahm, . E. Dunn, C. Leslie y G. Welch, «Testing and evaluation of a wearable augmented reality system for natural outdoor environments», P. R. H. M. P. B. J. E. M. Peter L. Marasco, Ed., *Head- and Helmet-Mounted Displays XVIII: Design and Applications: SPIE Vol. 8735*, 2014.
- [51] R. E. Johnson, «Components, Frameworks, Patterns», University of Illinois: Department of Computer Science, 1997.
- [52] D. C. Schmidt, «Applying Patterns and Frameworks to Develop Object-Oriented Communication Software», D. o. C. Science, Ed., Washington University: *Handbook of Programming Languages*, 1997.
- [53] L. Fuentes, J. M. Troya y A. Vallecillo, «Desarrollo de Software Basado en Componentes», E. I. C. Teatinos, Ed., Málaga, España: Dept. Lenguajes y Ciencias de la Computación, 2001.
- [54] S. Krakowiak, «Middleware Architecture with Patterns and Frameworks, France: Project Sardes - INRIA Grenoble Rhône-Alpes , 2009.
- [55] D. Amin y S. Govilkar, «Comparative Study of Augmented Reality SDK's,» *International Journal on Computational Sciences & Applications*, 5, 11-26., 2015.
- [56] F. Herpich, R. L. Martins Guarese y L. M. Rockenbach Tarouco, «A Comparative Analysis of Augmented Reality Frameworks Aimed at the Development of Educational Applications,» *Creative Education*, Vol.08, Article ID:77994,19 pages , 2017.

- [57] V. Rautenbach y S. Coetzee, «Results of an Evaluation of Augmented Reality Mobile Development Frameworks for Addresses in Augmented Reality,» Conference: Conference: FOSS4G 2015, Seoul, Korea, 2015.
- [58] C. Reynoso y N. Kicillof, «Estilos y Patrones en la Estrategia de Arquitectura de Microsoft,» Buenos Aires, Argentina, 2004.
- [59] C. Ardito, M. T. Baldassarre, D. Caivano y R. Lanzilotti, «Integrating a SCRUM-based process with Human Centred Design: an Experience from an Action Research Study,» IEEE/ACM 5th International Workshop on Conducting Empirical Studies in Industry (CESI), 2017.
- [60] M. R. Endsley, «The Designer's Situation Awareness Toolkit: Support for User-Centered Design,» Article in Human Factors and Ergonomics Society Annual Meeting Proceedings, 2004.
- [61] M. R. Endsley y D. G. Jones, «Designing for Situation Awareness. An approach to User-Centered Design.,» CRC Press Taylor & Francis Group, 6000 Broken Sound Parkway NW, Suite 300. Boca Raton, FL 33487-2742, 2004.
- [62] Y.-G. J. A. G. H. C.-W. N. Jun Yang, «Evaluating bag-of-visual-words representations in scene classification,» MIR '07 Proceedings of the international workshop on Workshop on multimedia information retrieval, ACM New York, NY, USA, 2007, 2007.
- [63] T. M. K. Tuytelaars, «Local invariant feature detectors: a survey,» Journal Foundations and Trends® in Computer Graphics and Vision archive Volume 3 Issue 3, January 2008, Hanover, MA, USA, 2008, 2008.
- [64] D. Lowe, «Object recognition from local scale-invariant features,» The Proceedings of the Seventh IEEE International Conference on. vol. 2, pp. 1150-1157, Washington, DC, USA, 1999.
- [65] E. P. R. D. T. Rosten, «Faster and better: A machine learning approach to corner detection,» Intern Analysis and Machine Intelligence, IEEE Transactions on 32(1), 105-119, Washington, DC, USA, 2010.
- [66] R. Ortiz, «Freak: Fast retina keypoint,» Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 510-517. IEEE Computer Society, DC, USA, 2012.

