INTERNATIONAL ORGANISATION FOR STANDARDISATION
ORGANISATION INTERNATIONALE DE NORMALISATION
ISO/IEC JTC1/SC29/WG11 and SC24/WG9

ISO/IEC JAhG MAR J0007
Warsaw, PL, June 2015

ISO/IEC JTC1/SC29/WG11 N14769
Sapporo, JP, July 2014

ISO/IEC JTC1/SC24/WG19 Nxxxx
August 2014

Source SC24 WG9 and SC29 WG11
Title Study Text of Mixed and Augmented Reality reference model CD
Editors Gerard J. Kim, Christine Perey, Marius Preda

Recipients of this draft are invited to submit, with their comments, notification of any relevant patent rights of which they are aware and to provide supporting documentation.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO [and/or] IEC shall not be held responsible for identifying any or all such patent rights.

ISO/IEC JTC 1 Nxxxx

Date: 2014-10-13

ISO/IEC CD 18039

ISO/IEC JTC 1

Secretariat: SC24 and SC29

Information technology — Computer graphics, image processing and environmental data representation and Coding of audio, picture, multimedia and hypermedia information — Mixed and Augmented Reality reference model

Technologie de l'information — Infographie, traitement d'image et données d'environment ET Codage de l'audio, image, multimédia et hypermédia — Modèle de référence pour la Réalité Augmentée

Document type: International Standard
Document subtype:
Document stage: (30) Committee
Document language: E

STD Version 2.1c2

Contents

Page

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of the joint technical committee is to prepare International Standards. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

ISO/IEC 18039 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 24 and 29, *Computer graphics, image processing and environmental data representation AND Coding of audio, picture, multimedia and hypermedia information*.

This second/third/... edition cancels and replaces the first/second/... edition (), [clause(s) / subclause(s) / table(s) / figure(s) / annex(es)] of which [has / have] been technically revised.

Information technology — Computer graphics, image processing and environmental data representation and Coding of audio, picture, multimedia and hypermedia information — Mixed and Augmented Reality Reference Model

1 Scope

This International Standard defines the scope and key concepts of mixed and augmented reality, the relevant terms and their definitions, and a generalized system architecture that together serve as a reference model for Mixed and Augmented Reality (MAR) applications, components, systems, services, and specifications. This reference model establishes the set of required modules and their minimum functions, the associated information content, and the information models that shall be provided and/or supported by a compliant MAR system.

The reference model is intended for use by current and future developers of MAR applications, components, systems, services, or specifications to describe, compare, contrast, and communicate their architectural design and implementation. The MAR-RM is designed to apply to MAR systems independent of specific algorithms, implementation methods, computational platforms, display systems, and sensors or devices used.

This International Standard does not specify how a particular MAR application, component, system, service, or specification shall be designed, developed, or implemented. It also does not specify the bindings of those designs and concepts to programming languages, or the encoding of MAR information through any coding technique or interchange format. This specification contains a list of representative system classes and use cases with respect to its reference model.

This standard does not rely on any documents as normative references.

2 Terms, Definition, Symbols and Abbreviated Terms

For the purpose of this specification, the following terms and definitions apply. This will help MAR practioners to communicate more effectively.

Note that other SDOs or organizations may have slightly different definitions of some of the terms used in this document. In most cases, unless otherwise stated, it is not the intent of this document ot redefine such terms and they are only used in the generic sense. This section also provides a table of symbols and abbreviated terms used throughout the document.

- **Augmentation**

Virtual object data (computer-generated information) added on to or associated with target physical object data in MAR scene, or physical object data added on to or associated with target virtual object data.

- **Augmented reality system**

Type of mixed reality system in which virtual-world data (e.g., computer-generated information) are embedded and registered in the physical-world data representation

- **Augmented virtuality system**

Type of a mixed reality in which physical-world data (e.g., live video) are embedded and registered in the virtual-world data representation

- **Display**

Device by which rendering results are presented to user. It can use various modalities such as visual, auditory, haptics, olfactory, thermal, motion, etc. In addition, any actuator can be considered display if it is controlled by MAR system.

- **Feature**

Primitive geometric elements (e.g., points, lines, polygons, colour, texture, shapes, etc.) or attributes of given (usually physical) object used in its detection, recognition and tracking.

- **MAR event**

Result of detection of condition relevant to MAR content (e.g., as condition for augmentation).

- **MAR execution engine**

MAR execution engine is a collection of hardware and software elements that produce the result of combining components that represent on the one hand the real world and its objects, and on the other those that are virtual, synthetic and computer generated.

- **MAR experience**

MAR experience is the human visualization and interaction of a MAR scene.

- **MAR scene**

A MAR Scene is the observable spatio-temporal organization of physical and virtual objects. This is the result of a MAR scene representation being interpreted by a MAR execution engine. A MAR scene has at least one physical and one virtual object.

- **MAR scene representation**

A data structure that arranges the logical and spatial representation of a graphical scene including the physical and virtual objects that is used by the MAR execution engine to produce a MAR scene.

MAR scene representation content can be produced by MAR authoring tools and stored within or external to MAR-enabled systems.

- **Mixed and augmented reality system**

Term that is synonymous with mixed reality system[1].

- **Mixed reality continuum**

Spectrum spanning physical and virtual realities according to a proportional composition of physical and virtual data representations (originally proposed by Milgram et al [3])

- **Mixed reality system**

[1] The word “augmented” is often used together with word “mixed”.

System that uses mixture of physical world data and virtual world data representation as its presentation medium.

- **Natural feature**

Features that are not artificially inserted for purpose of easy detection/recognition/tracking.

- **Physical object**

Physical object that is designated for augmentation with virtual data representation.

- **Physical reality**

Term synonymous to physical world itself or medium that represents the physical world (e.g., live video or raw image of real world)

- **Physical world**

Spatial organization of multiple physical objects.

- **Point of interest**

Single or collection of target locations. Aside from location data, a point of interest is usually associated with metadata such as identifier and other location specific information.

- **Recognizer**

MAR component (hardware/software) that processes sensor output and generates MAR events based on conditions indicated by the CC.

- **Sensor**

Device that return detected values related to detected or measured condition or property. Sensor may be an aggregate of sensors.

- **Spatial registration**

The establishment of the spatial relationship or mapping between two models, typically between virtual object and target physical object.

- **Target image**

A target object represented by a 2D image.

- **Target object**

A target physical object designed or chosen to allow detection, recognition and tracking (and finally augmentation).

- **Tracker**

MAR component (hardware/software) that analyses signals from sensors and provides some characteristics of tracked entity (e.g., position, orientation, amplitude, profile).

- **Virtual object**

Computer-generated entity that is designated for augmentation in association with a physical object data representation. In context of MAR, it usually has perceptual (e.g., visual, aural) characteristics and optionally, dynamic reactive behaviour.

- **Virtual world or Environment**

Spatial organization of multiple virtual objects, potentially including global behaviour.

2.1 Table of abbreviated terms

Table X shows a list of abbreviated terms and symbols used in this document.

AbbreivateTerms and Symbols	**Terms**
API	Application Program Interface
AR	Augmented Reality
GNSS	Global Nagivation Satellite System
MAR	Mixed and Augmented Reality
MAR-RM	Mixed and Augmented Reality Reference Model
MR	Mixed Reality
POI	Points of Interest
PTAM	Parallel Tracking and Mapping
SLAM	Simultaneous Localization and Mapping
UI	User Interface
VR	Virtual Reality

Table X: Abbreivated terms used in this document.

3 MAR Domain and Concepts

3.1 Introduction

Mixed and Augmented Reality (MAR) refers to a spatially coordinated combination of media/information components that represent on the one hand the real world and its objects, and on the other those that are virtual, synthetic and computer generated. The virtual component can be represented and presented in many modalities (e.g., visual, aural, touch/haptic, olfactory, etc.) as illustrated in Figure 1. The figure shows a MAR system in which a virtual fish is augmented above a real world object (registered by using markers), visually, aurally and haptically[2].

[2] Illustration provided by Magic Vision Lab, University of South Australia.

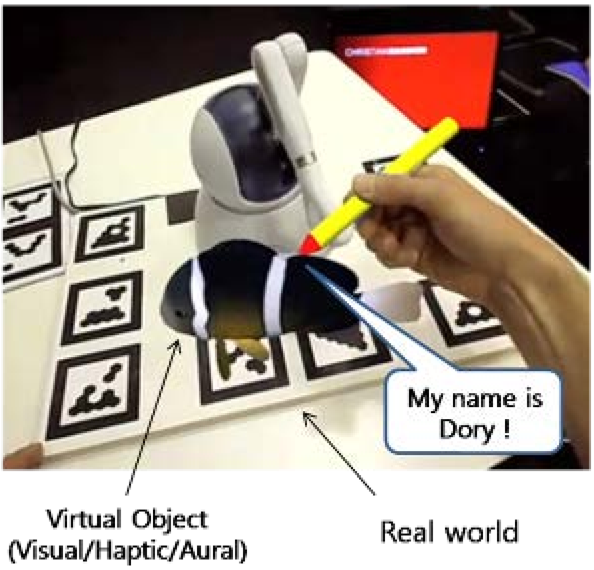

Figure 1. The concept of MAR combines representations of physical objects and computer mediated virtual ones in various modalities (e.g. text, voice, and force feedback).

Through such combinations, the physical (or virtual) object can be presented in an information-rich fashion through "augmentation" with the virtual (or real) counterpart. Thus, the idea of *spatially coordinated combination* is important for highlighting the mutual association between the physical and virtual worlds. This is also often referred to as *registration* and can be done in various dimensions. The most typical registration is spatial, where the position and orientation of a real object is computed and used to control the position and orientation of a virtual object. Temporal registration may also occur when the presence of a real object is detected and a virtual object will be displayed. Registration may have various precision performances; it can vary in its degree of tightness (as illustrated in Figure 2). For example, in the spatial dimension, it can be measured in terms of distance or angles; in the temporal dimension in terms of milliseconds.

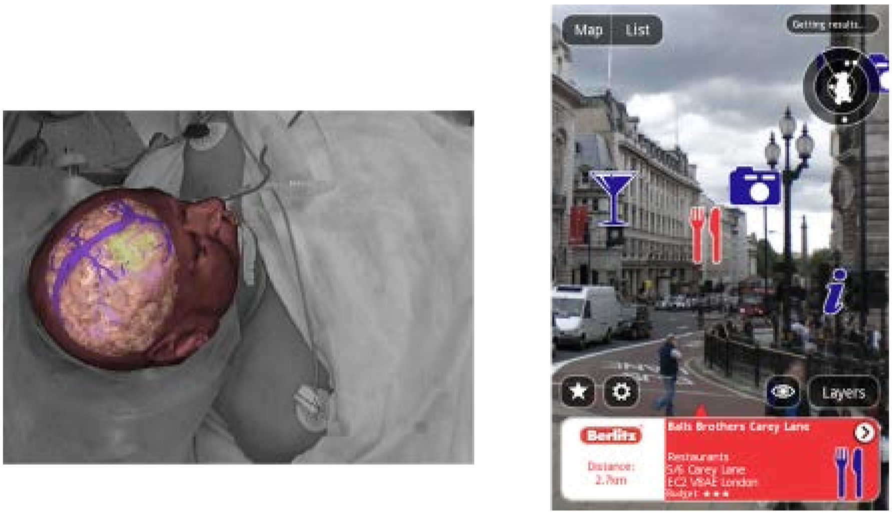

Figure 2. The notion of registration precision is shown at different degrees: (1) virtual brain imagery tightly registered on a real human body image (left) [1], (2) tourist information overlaid less tightly over a street scene [source Layar].

A MAR system refers to real time processing [2]. For example, while a live close-captioned broadcast would qualify as a MAR service, an offline production of a subtitled movie would not.

3.2 MAR continuum

Since a MAR system or its contents combines real and virtual components, a MAR continuum can be defined according to the relative proportion of the real and virtual, encompassing the physical reality ("All Physical, No Virtual") on one end, and the virtual reality ("All Virtual, No Physical") on the other end (illustrated in Figure 3.). At any point on this continuum [3], i.e., a single instance of a system that uses a mixture of both real and virtual presentation media is called a mixed reality system. In addition, for historical reasons, "mixed reality" is often synonymously or interchangeably used with augmented reality, which is actually a particular type of mixed reality (see Section 7). In this International Standard, the term "mixed and augmented reality" is used to avoid such confusion and emphasize that the same model applies to all combinations of real and digital components along the continuum. The two extreme ends in the continuum (the physical reality and the virtual reality) are not in the scope of this document.

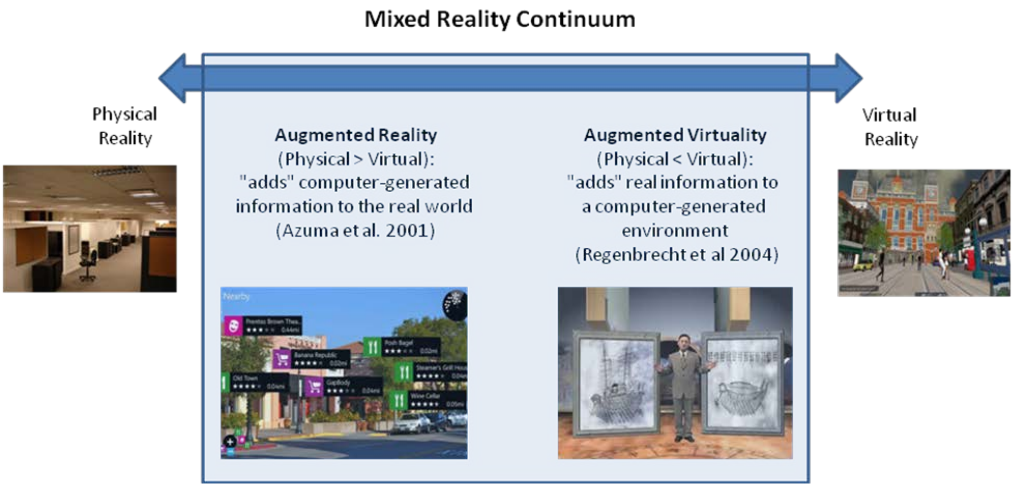

Figure 3. The MAR (or Reality-Virtuality) continuum defines different genres of MR according to the relative portion between the real world representation and the virtual

Two notable genres of MAR or points in the continuum are the Augmented Reality (AR) and Augmented Virtuality. An augmented reality system is a type of mixed reality system in which the medium representing the virtual objects is embedded into the medium representing the physical world (e.g., video). In this case, the physical reality makes up a larger proportion of the final composition than the computer-generated information. An augmented virtuality system is a type of a mixed reality system in which the medium representing physical objects (e.g., video) is embedded into the computer-generated information (as illustrated in Figure 3).

4 MAR reference model Usage Example

4.1 Designing a MAR Application or Service

The MAR-RM is a reference guide in designing a MAR service and developing a MAR system, application, or content. With respect to the given application (or service) requirements, the designer may refer to, and select the needed components from those specified in the MAR model architecture. The functionalities, the interconnections between components, the data/information model for input and output, and relevant existing standards for various parts can be cross-checked to ensure generality and completeness. The classification scheme described in Section 9 can help the designer to specify a more precise scope and capabilities while the specific system classes defined in Section 10 can facilitate the process of model, system or service refinement.

4.2 Deriving a MAR Business Model

The MAR-RM document introduces an enterprise viewpoint with the objective of specifying the industrial ecosystem, identifying the types of actors and describing various value chains. A set of business requirements is also expressed. Based on this viewpoint, companies may identify current business models or invent new ones.

4.3 Extend Existing or Create New Standards for MAR

Another expected usage of the MAR-RM is in extending or creating new application standards for MAR functionalities. MAR is an interdisciplinary application domain involving many different technologies, solutions, and information models, and naturally there are ample opportunities for extending existing technology solutions and standards for MAR. The MAR-RM can be used to match and identify components for those that might require extension and/or new standardization. The computational and information models can provide the initial and minimum basis for such extensions or for new standards. In addition, strategic plans for future standardization can be made. In the case when competing de facto standards exist, the reference model can be used to make comparisons and evaluate their completeness and generality. Based on this analysis and the maturity of the standards, incorporation of de facto standards into open ones may be considered (e.g., markers, API, points of interest constructs, etc.).

5 MAR Reference System Architecture

5.1 Overview

A Mixed and Augmented Reality (MAR) system requires several different components to fulfil its basic objectives: real time recognition of the physical world context, the registration of target physical objects with their corresponding virtual objects, display of MAR content and handling of user interaction(s). A high-level representation of the typical components of a MAR system is illustrated in Figure 4. The central pink area indicates the scope of the MAR-RM. Blue round boxes are main computational modules and the dotted box represents the required information constructs. Arrows indicate data flow, and control signals are not explicitly shown based on the principle to avoid having particular technical or implementational details in this standard. Also, note that this reference system architecture should not to be taken as something that is rigid and unchangeable, but as a depiction of a typical case at the macro scale and flexibility exists in its actual application.

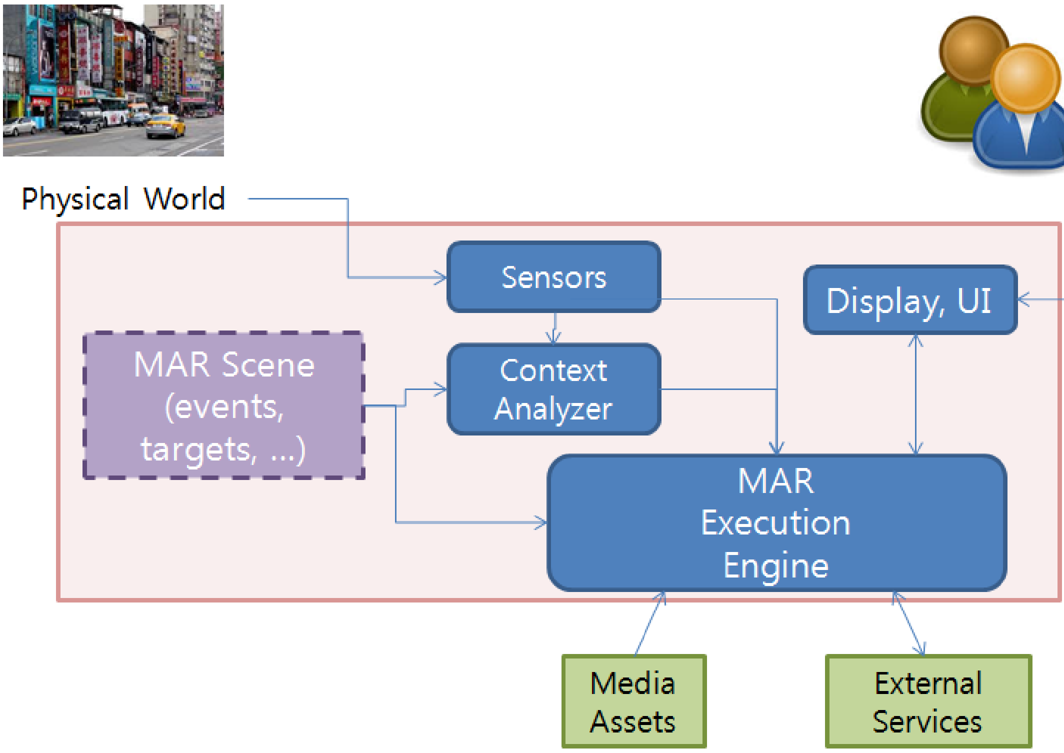

Figure 4. Major components and their interconnection in an MAR system are shown at a high level macro level. Arrows illustrate data flow.

The MAR execution engine has a key role in the overall architecture and is responsible for:

- Processing the content as specified and expressed in the MAR scene including additional media content provided in media assets.
- Processing the user input(s);
- Processing the context provided by the sensors capturing the real world;
- Managing the presentation of the final result (aural, visual, haptic and commands to additional actuators);
- Managing the communication with additional services.

5.2 Viewpoints

In order to detail the global architecture presented in Figure 4, this reference model considers several analysis angles, called viewpoints, namely: Enterprise, Computation and Information. This viewpoint-wise exposition permits readers, who may be interested in or focused on particular aspects or viewpoints, to understand the MAR architecture better. The definition of each viewpoint is provided in the following table.

The notion of view is separate from that of the viewpoint, where a viewpoint identifies the set of concerns and the representations/modeling techniques, etc. used to describe the architecture to address those concerns, and a view is the result of applying a viewpoint to a particular system [25].

Table 1. Definition of the MAR viewpoint

Viewpoint	Viewpoint Definition	Topics covered by MAR-RM
Enterprise	Articulates the business entities in the system that should be understandable by all stakeholders. This focuses on purpose, scope, and policies and introduces the objectives of different actors involved in the field.	• Actors and their roles; • Potential business models for eac h actor; • Desirable characteristics for the a ctors at both ends of the value ch ain (creators and users).

Viewpoint	Viewpoint Definition	Topics covered by MAR-RM
Computational	Identifies the functionalities of system components and their interfaces. It specifies the services and protocols that each component exposes to the environment.	• Services provided by each AR main component; • Interface description for some use cases.
Information	Provides the semantics of information in the different components in the views, the overall structure and abstract content type as well as information sources. It also describes how the information in processed inside each component. This view does not provide a full semantic and syntax of data but only a minimum of functional elements and should be used to guide the application developer or standard creator for creating their own information structures.	• Context information such as spatial registration, captured video and audio, etc.; • Content information such as virtual objects, application behaviour and user interaction(s) management; • Service information such as remote processing of the context data.

5.3 Enterprise viewpoint

The **Enterprise viewpoint** describes the actors involved in a MARS, their objectives, roles and requirements. The actors can be classified according to their role. There are four classes of actors.

5.3.1 Classes of Actors

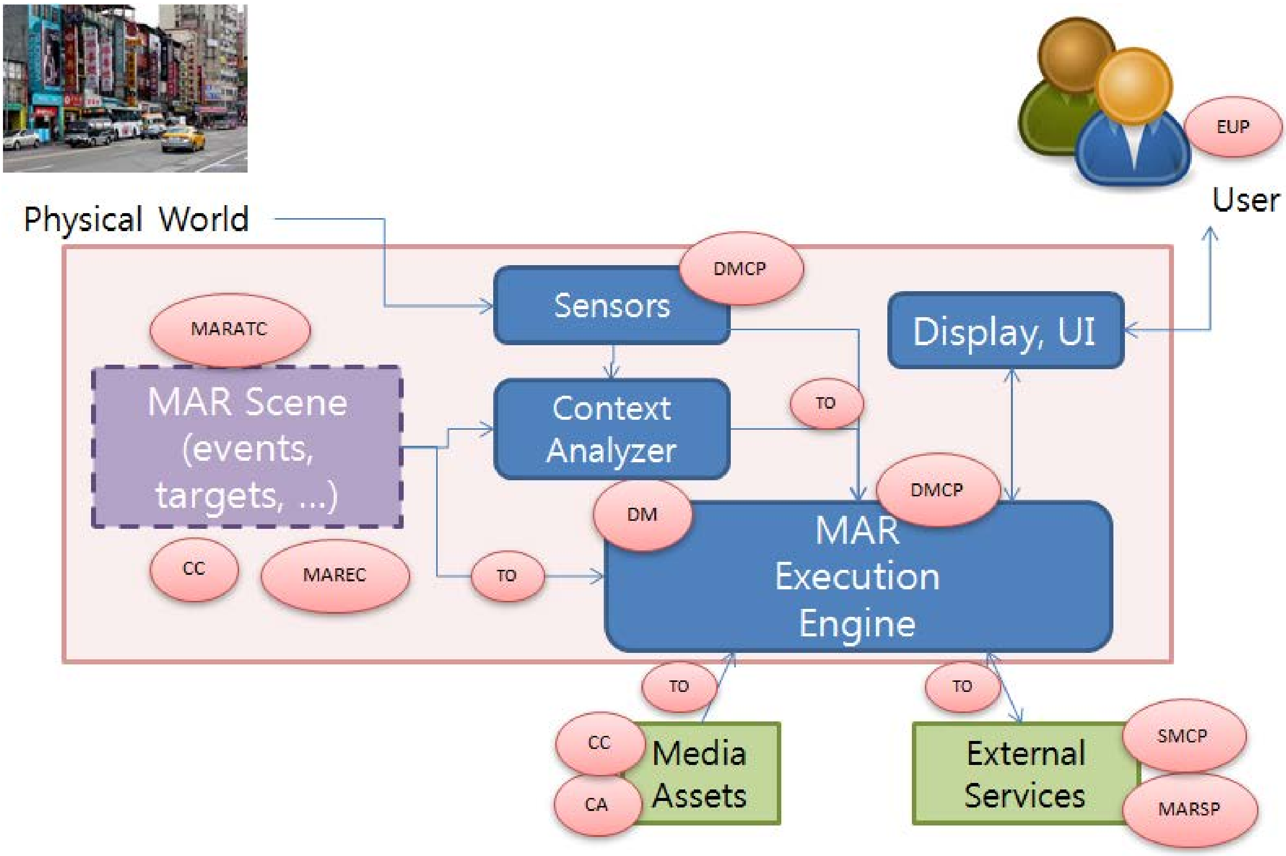

Figure 5. The “Enterprise Viewpoint” viewpoint of a MAR system is shown along with the main “actors” (the acronyms are described below, as well as in Section 2, “*Symbols and abbreviated terms*”).

Class 1: Providers of authoring/publishing capabilities

- MAR Authoring Tools Creator (MARATC)
  - A software platform provider of the tool used to create (author) a MAR-enabled Application or Service. The output of the MAR authoring tool is called MAR rich scene representation.
- MAR experience creator
  - A person that designs and implements a MAR-enabled Application or Service.
- Content Creator (CC)
  - A designer (person or organization) that creates multimedia content (scenes, objects, etc.). Note that even the end user of the MAR system can be the designer or the content.

Class 2: Providers of MAR execution engine components

- Device Manufacturer (DM)
  - An organization that produces devices in charge of augmentation and used as end-user terminals.
  - An organization that produces devices in charge of augmentation and used as end-user terminals.
- Device Middleware/Component Provider (DMCP)
  - An organization that creates and provides hardware, software and/or middleware for the augmentation device. In this category belong modules such as:
    - Multimedia player or browser engine provider (rendering, interaction engine, execution, etc.).
    - Context knowledge provider (satellites, etc.).
    - Sensor manufacturers (inertial, geomagnetic, camera, microphone, etc.).

Class 3: Service Providers

- MAR Service Provider (MARSP)
  - An organization that discovers/delivers (SA-024) services.
- Content Aggregator (CA)
  - An organization aggregating, storing, processing and serving content.
- Telecommunication Operator (TO)
  - An organization that manages telecommunication among other actors.
- Service Middleware/Component Provider (SMCP)

  - An organization that creates and provides hardware, software and/or middleware for processing servers. This category includes services such as:
    - Location providers (network-based location services, image databases, RFID based location, etc.).
  - Semantic provider (indexed image or text databases, etc.).

Class 4: MAR User

- MAR Consumer/End-User Profile (EUP)
  - A person who experiences the real world synchronized with digital assets. He/she uses a MAR scene representation, a MAR execution engine and MAR services in order to satisfy information access and communication needs. By means of their digital information display and interaction devices, such as smart phones, desktops and tablets, users of MAR hear, see and/or feel digital information associated with natural features of the real world, in real time.

Several types of actors from the list above can commercially exploit an MAR system.

5.3.2 Business Model of MAR systems

The actors in the MAR system have different business models:

- A MAR Authoring Tools Creator may provide the authoring software or content environment to a MAR experience creator. Such a tool ranges in complexity from full programming environments to relatively easy-to-use online content creation systems.

- The Content creator prepares a digital asset (text/picture/video/3D model/animation) that may be used in the MAR experience.

- A MAR experience creator creates a MAR experience in the form of a MAR rich media representation. He/she can associate media assets with features in the real world, thereby transforming them into MAR enabled digital assets. The MAR experience creator also defines the global/local behaviour of the MAR experience. The creator should consider the performances of obtaining and processing the context as well as performance of the AR Engine. A typical case would be one where the MAR experience creator will specify a set of minimal requirements that should be satisfied by the hardware or software components.

- A middleware/component provider produces the components necessary for core enablers to provide key software and hardware technologies in the fields of sensors, local image processing, display, remote computer vision and remote processing of sensor data for MAR experiences. There are two types of middleware/component providers: device (executed locally) and services (executed remotely).

- MAR Service Provider is a broad term meaning an organization that supports the delivery of MAR experiences. This can be via catalogues or to assist in discovering a MAR experience.

5.3.3 Criteria for Successful MAR system

The requirements for the successful implementation of MAR system are expressed with respect to two types of actors. While the end user experience for MAR should be more engaging than browsing Web pages, it should be possible to create, transport and consume MAR experiences with the same ease as is currently possible for Web pages.

5.3.4 Actor Requirements

MAR experience creator requirements:

- Tools for the authoring workflow should be available for both non-technical people and experts.
- It should be possible to create indoor and outdoor MAR experiences and it should be possible to seamlessly integrate them.
- It should be possible to create MAR experiences independently of user location and registration technology.
- It should be possible to augment at least aural and/or visual senses and, in general, all the human senses, in a realistic manner.
- It should be possible to incorporate MAR experiences in existing applications and services.
- It should be possible to connect to additional data providers, service providers, etc. not originally intended to be used in MAR experiences.

MAR end-user requirements:

- Accurate real time registration of the MAR experience with real world and composition in a natural-synthetic scene, based on context (e.g., geospatial coordinates, vision, etc.).
- Automatic, continuous consideration of user context in consuming the MAR experience (e.g., user profile, user interaction(s), user preference, user location, device status, etc.).

5.4 Computational viewpoint

The Computational viewpoint describes the overall interworking of a MAR system. It identifies major processing components (hardware and software), defines their roles and describes how they interconnect.

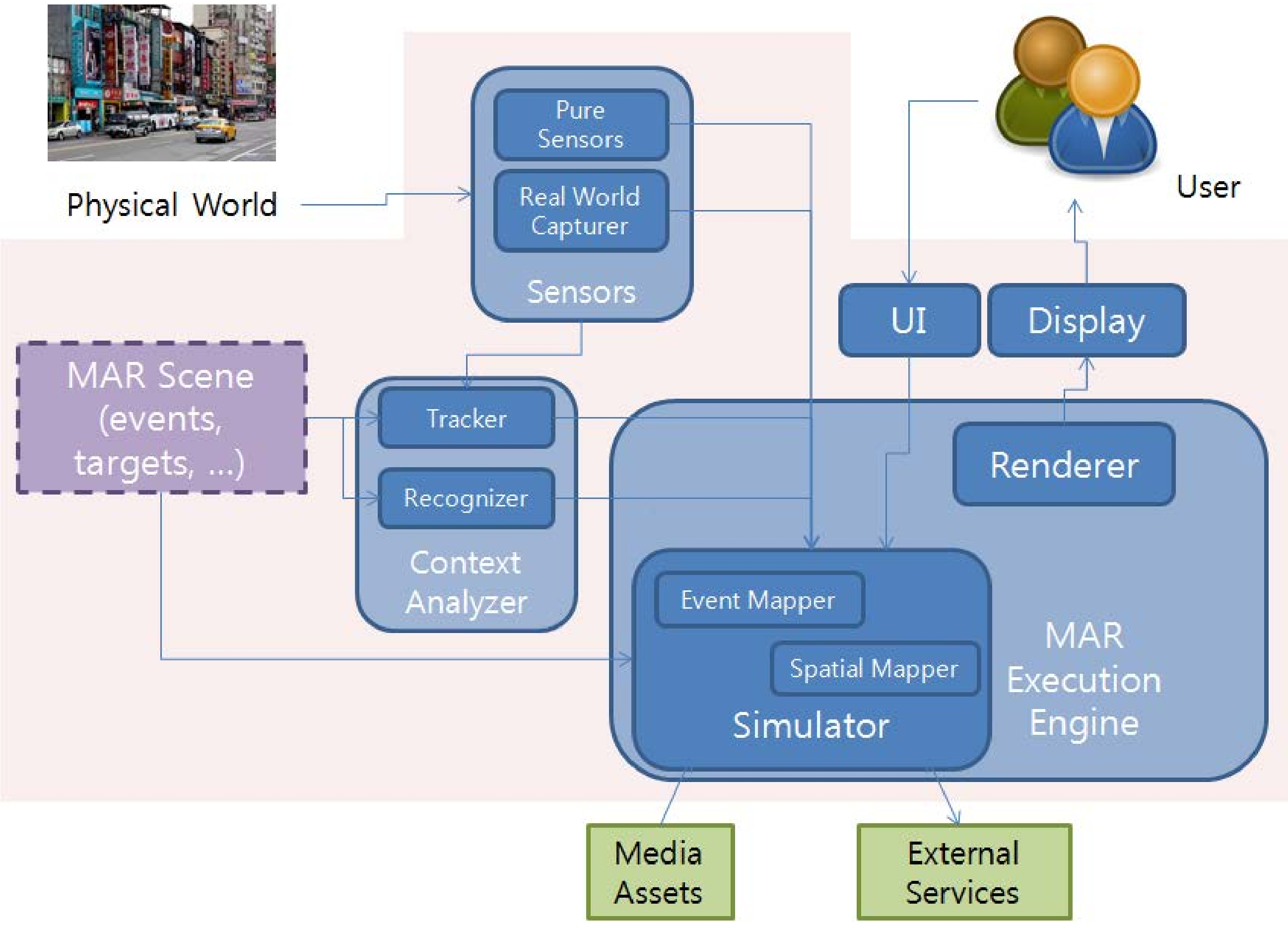

Figure 6. The “Computation” viewpoint illustrates and identifies the major computational blocks in the MAR system/service.

5.4.1 Sensors: Pure Sensor and Real World Capturer

A Sensor is a hardware (and optionally) software component able to measure specific physical property. In the context of MAR, a sensor is used to detect, recognize and track the target physical object to be augmented. In this case, it is called a “pure sensor.” Another use of a sensor is to capture and stream to the Execution Engine, the data representation of the physical world or objects for composing a MAR scene. In such a case, it is called a “real world capturer” A typical example is the video camera that captures the real world as a video to be used as a background in an augmented reality scene. Another example is “Augmented Virtuality,” where a person is filmed in the real world and the corresponding video is embedded into a virtual world. Note that the captured real world data can be in any modality such as visual, aural, haptic, etc.

A sensor can measure different physical properties, and interpret and convert these observations into digital signals. The captured data can be used (1) to only compute the context in the tracker/recognizer, or (2) to both compute the context and contribute to the composition of the scene. Depending on the nature of the physical property, different types of devices can be used (cameras, environmental sensors, etc.). One or more sensors can simultaneously capture signals.

The input and output of the Sensors are:

- Input: Real world signals.
- Output: Sensor observations with or without additional metadata (position, time, …).

The Sensors can be categorized as follows:

Table 2. Sensor categories

Dimension	Types					
1. Modality/type of the sensed/captured data	Visual	Auditory	Electro-magnetic waves (e.g., GNSS)	Haptic/tactile	Temperature	Other physical properties
2. State of sensed/captured data	Live	Pre-captured	—	—	—	—

5.4.2 Recognizer and Tracker

The Recognizer is a hardware or software component that analyses signals from the real world and produces MAR events and data by comparing with a local or remote target signal (i.e., target for augmentation).

The Tracker is able to detect and measure changes of the properties of the target signals (e.g., pose, orientation, volume, etc.).

Recognition can only be based on prior captured target signals. Both the Recognizer and Tracker can be configured with a set of target signals provided by or stored in an outside resource (e.g. third party DB server) in a consistent manner with the scene definition, or by the MAR scene description (See Section 5.5.6) itself.

Recognizer and Tracker can be independently implemented and used.

- The input and output of the Recognizer are:
  - Input: Raw or processed signals representing the physical world (provided by sensors) and ta rget object specification data (reference target to be recognized).
  - Output: At least one event acknowledging the recognition.

Table 3. Recognizer categories

Dimension	Types					
1. Form of target signal	2D image patch	3D primitives (points, lines, polygons, shapes)	3D model	Location (e.g., earth-reference coordinates)	Audio patch	Other
2. Form of the output event	Indication only of the recognized event	Additional data such as data type, timestamp, recognition confidence level, other attributes	—	—	—	—
3. Place of execution	Local system	Remote system (server, cloud, etc.)	—	—	—	—

- The input and output of the Tracker are:

- Input: Raw or processed signals representing the physical world and target object specificatio n data (reference target to be recognized).
- Output: Instantaneous values of the characteristics (pose, orientation, volume, etc.) of the rec ognized target signals.

Table 4. Tracker categories

Dimension	**Types**				
1. Form of target signal	2D image patch	3D primitives (points, lines, polygons, shapes)	3D Model	Location (e.g., earth-reference coordinates)	Other
2. Form of the output event	Spatial (2D, 3D, 6D, …)	Aural (intensity, pitch, …)	Haptic (force, direction, …)	Others	—
3. Place of execution	Local system	Remote system (server, cloud, etc.)	—	—	—

5.4.3 Spatial mapper

The role of the Spatial mapper is to provide spatial relationship information (position, orientation, scale and unit) between the physical space and the space of the MAR scene by applying the necessary transformations for the calibration. The spatial reference frames and spatial metrics used in a given sensor needs to be mapped into that of the MAR scene so that the sensed real object can be correctly placed, oriented and sized. The spatial relationship between a particular sensor system and an augmented space is provided by the MAR experience creator and is maintained by the Spatial mapper.

- The input and output of the Spatial mapper are:
  - Input: Sensor identifier and sensed spatial information.
  - Output: Calibrated spatial information for the given MAR scene.

The notion of the Spatial mapper can be extended to mapping other domains such as audio (e.g., direction, amplitude, units, scale) and haptics (e.g., direction, magnitudes, units and scale).

5.4.4 Event mapper

The Event mapper creates an association between a MAR event, obtained from the Recognizer or the Tracker, and the condition specified by the MAR Content creator in the MAR scene.

It is possible that the descriptions of the MAR events produced by the Recognizer or the Tracker are not the same as those used by the Content creators even though they are semantically equivalent. For example, a recognition of a particular location (e.g., longitude of -118,24 and latitude of 34.05) might be identified as “MAR_location_event_1” while the Content creator might refer to it in a different vocabulary or syntax, e.g., as “Los Angeles, CA, USA.”

The event relationship between a particular recognition system and a target scene is provided by the MAR experience creator and is maintained by the Event mapper.

- The input and output of the Event mapper are:

  - Input: Event identifier and event information.
  - Output: Translated event identifier for the given MAR scene.

5.4.5 MAR execution engine

The MAR execution engine constitutes the core of any MAR system. Its main purpose is to interpret the sense d data to further recognize and track the target data to be augmented, import the real world or object data, co mpuationally *simulate* the dynamnic behaviour of the augmented world, compose the real and virtual data tog ether for proper rendering in the required modalities (e.g. visually, aurally, haptically). The Execution Engine m ight require additional and external media assets or computational services for supporting these core function alities. The MAR execution engine can be part of a software application able to load a full scene description (i ncluding assets, scene behaviour, user interaction(s), …) for its simulation and presentation or part of a stand-alone application with pre-programmed behaviour.

The Execution Engine is a software component capable of (1) loading the MAR scene description as provided by the MAR experience creator or processing the MAR scene as specified by the application developer, (2) interpret data provided by various mappers, user interaction(s), sensors, local and/or remote services, (3) execute and simulate scene behaviours, (4) compose various types of media representations (aural, visual, haptics, …).

- The input and output of the Execution Engine are:
  - Input: MAR scene description, user input(s), (mapped) MAR events and external service even ts.
  - Output: an updated version of the scene description.

The Execution Engine might be categorized according to the following dimensions:

Table 5. Execution Engine categories

Dimension	**Types**			
1. Space & time	2D + time	3D + time	—	—
2. User Interactivity	Yes	No	—	—
3. Execution place	Local	Remote	Hybrid	—
4. Number of simultaneous users	Single-user	Multi-user	—	—

5.4.6 Renderer

The Renderer refers to the software and optionally hardware components for producing, from the MAR scene description (See Section 5.5.6), updated after a tick of simulation, a presentation output in a proper form of signal for the given display device. The rendered output and the associated displays can be in any modality. When multiple modalities exist, they need to be synchronized in proper dimensions (e.g., temporally, spatially).

- The input and output of the Renderer are:
  - Input: (Updated) MAR scene graph data.
  - Output: Synchronized rendering output (e.g., visual frame, stereo sound signal, motor comma

nds, etc.).

The Renderer can be categorized in the following way:

Table 6. Renderer categories

Dimension	Types			
1. Modality	Visual	Aural	Haptics	Others
2. Execution place	Local	Remote	Hybrid	—

5.4.7 Display and User Interface

The Display is a hardware component that produces the actual presentation of the MAR scene to the end-user in different modalities. Displays and UI include monitors, head-mounted displays, projectors, scent diffusers, haptic devices and sound speakers. A special type of display is an actuator that does not directly stimulate the end-user senses but may produce a physical effect in order to change some properties of the physical objects or the environment. The UI is a hardware component used to capture user interaction(s) (touch, click) for the purpose of modifying the state of the MAR scene. The UI requires sensors to achieve this purpose. However, these sensors may have a similar usage as those known as pure sensors. The difference consists then in the fact that the only physical object sensed is the user.

- The input and output of the Display are:
    - Input: Render signals
    - Output: Display output
- The input and output of the UI are:
    - Input: User action
    - Output: UI event

The Displays may be categorized according to their modalities with each of them having their own attributes as follows:

Table 7. Visual display categories

Dimension	Types			
1. Presentation	Optical see through	Video see through	Projection	—
2. Mobility	Fixed	Mobile	Controlled	—
3. No. of channels	2D (mono)	3D stereoscopic	3D holographic	—

Table 8. Aural display categories

Dimension	Types			
1. No. of channels	Mono	Stereo	Spatial	—

2. Acoustic space coverage	Headphones	Speaker	—	—

Table 9. Haptics display categories

Dimension	**Haptic mode**			
Type	Vibration	Pressure	Temperature	Other

Table 10. UI categories

Dimension	**Input method**			
Type	click	Drag and drop	Touch	Natural interface (voice, facial expression, gestures, etc.)

5.4.8 MAR system API

The MAR components defined in the Computational viewpoint may have an exposed API, thereby simplifying application development and integration. Additionally, higher-level APIs can be specified in order to make abstractions for often-used MAR functionalities and data models in the following way (not exhaustive):

- Defining the markers and target objects for augmentation.
- Setting up multi-markers and their relationships.
- Setting up and representing the virtual/physical camera and viewing parameters.
- Detection and recognition of markers/target objects.
- Managing markers/target objects.
- Extracting specific spatial properties and making geometric/matrix/vector computations.
- Loading and interpreting MAR scene representation.
- Calibrating sensors and virtual/augmented spaces.
- Mapping of MAR events between those that are user defined and those that are system defined.
- Making composite renderings for specific displays, possibly in different modalities.

Such APIs are designed to simplify the development of special-purpose MAR systems.

5.5 Information viewpoint

The Information viewpoint provides some key semantics of information associated with the different components in other viewpoints, including the semantics of input and output for each component as well as the overall structure and abstract content type. This viewpoint does not provide a full semantic and syntax of data but only **minimum functional elements** and it should be used to guide the application developer or standard creator in creating their own information structures. Let us note that for some components, there are already standards available providing full data models.

5.5.1 Sensors

This component is a physical device characterized by a set of capabilities and parameters. A subclass of Sensors is the Real World Capturer whose output is an audio, video or haptics stream to be embedded in the

MAR scene or analysed by specific hardware or software components. Additionally, several parameters are associated with the device or with the media captured such as intrinsic parameters (e.g., focal length, field of view, gain, frequency range, etc.), extrinsic parameters (e.g., position and orientation), resolution, sampling rate. The captured audio data can be mono, stereo or spatial. The video can be 2D, 3D (colour and depth) or multi-view. As an example, the following table illustrates possible sensor specifications:

Table 11. Sensor attribute example

Sensor Attribute	**Values**
Identifier	“Sensor 1”, “Sensor 2”, “My Sensor”, etc.
Type	Video, audio, temperature, depth, image etc.
Sensor specific attributes	120° (field of view), 25 (frequency), 41000 (sampling rate), etc.

The input and output of the Sensors are:

- Input: the real world (no information model is required).
- Output: sensor observations (optionally post-processed in order to extract additional metadata such a s position, time, etc.). They depend on the type of the sensor used (e.g., binary image, colour image, depth map, sound stream, force, etc.).

5.5.2 Recognizer

There are two types of information used by the recognizer: the sensors output and the target physical object r epresentation. By analysing this information, the Recognizer will output a MAR event.

- The input data model of the Recognizer is the output of the sensors.
- The target physical object data should contain the following elements. First, it will have an identifier in dicating the event when the presence of the target object is recognized. The target physical object spe cification may include raw template files used for the recognition and matching process such as image files, 3D model files, sound files, etc. In addition to the raw template files or data, it could also include a set of feature profiles. The types of features depend on the algorithms used by the Recognizer. For i nstance it could be a set of visual feature descriptors, 3D geometric features, etc..

Table 12. Target physical object attribute

Target Physical Object Attribute	**Values**
Recognition event identifier	“Image_1”, “Face_Smith”, “Location_1”, “Teapot3d”, etc.
Raw template file / data	hiro.bmp, smith.jpg, teapot.3ds, etc.
Feature set definition	Set of visual features, set of aural features, set of 3D geometry features, etc.

- The output is an event that at least identifies the recognized target, and optionally provides additional information, that should follow a standard protocol, language, and naming convention. As an example, the following table illustrates possible event specification.

Table 13. Attributes for the Recognizer output

Attribute	**Values**
Identifier	“Event 1”, “Location 1”, “My_Event”, etc.
Type	Location, Object, Marker, Face, etc.
Value	Paris, Apple, HIRO, John_Smith, etc.
Time stamp	12:32:23, 02:23:01

5.5.3 Tracker

There are two types of information used by the Recognizer: the sensors output and the target physical object r epresentation. By analysing this information, the Tracker will output a MAR event.

- The input data model of the Recognizer is the output of the sensors.
- The target physical object data should contain the same elements as for Recognizer.
- Output : A continuous stream of instantaneous values of the characteristics (pose, orientation, volume, etc.) of the recognized target signals.

Table 14. Attribute for the Tracker output

Attribute	**Values**
Identifier (of the stream of tracking data)	“GNSS_location_stream”, “Marker_location_stream”, “Object_orientation_stream”, etc.
Type	Location, object, marker, face, etc.
Tracking data (elements of the stream)	Inertial position, 4x4 transformation matrix, current volume level, current force level, etc.
Optional: Time stamp	12:32:23, 02:23:01

5.5.4 Spatial mapper

In order to map the physical sensor space into the MAR scene, explicit mapping information must be supplied by the content or system developer. The spatial mapping information can be modelled as a table with each entry characterizing the translation process from one aspect of the spatial property (e.g., lateral unit, axis direction, scale, etc.) of the sensor to the given MAR scene. There is a unique table defined for a set of sensors and a MAR scene.

Table 15. Spatial mapping table example

Sensor 1	MAR Scene 1
ID_235 (Sensor ID)	MyMarkerObject_1 (a scene graph node)
Sensor position and orientation	T (3.0, 2.1, 5.5), R (36°, 26°, 89°). Used to convert from physical space to the scene space (align the coordinate systems)
Scale in (X, Y, Z)	(0.1, 0.1, 0.1). Used to convert from physical space to the scene space (align the coordinate systems)

5.5.5 Event mapper

In order to map MAR events as defined by the content developer or specified within the MAR scene represent ation, as well as events identified and recognized by the Recognizer, a correspondence table is needed. The t able provides the matching information between a particular recognizer identifier and an identifier in the MAR scene. There is a unique table defined for a set of events and a MAR scene.

Event Set	MAR Scene Event Set
Location =(2.35, 48.85)	Location= Paris, France
R_event_1	My_Event_123
Right_Hand_Gesture	OK_gesture

5.5.6 Execution Engine

The Execution Engine has several inputs. The main input is the MAR scene description that contains all information about how the MAR experience creator set up the MAR experience, such as:

- Initial scene description including spatial organisation.
- Scene behaviour.
- Specification of the representation of the real objects to be detected and tracked (targeted for augmentation) as well as the virtual assets to be used for augmentation, and the association between the representation of the real objects and their corresponding synthetic assets.
- The calibration information between the sensor coordinate system and the MAR scene coordinate system (supplied to the Spatial mapper).
- The mapping between identifiers or conditions outputted by the recognizer or tracker and elements of the MAR scene graph (supplied to the Event mapper).
- The set of sensors and actuators used in the MAR experience.

- The way in which the user may interact with the scene.
- Access to remote services like maps, image databases, processing servers, etc..

The Execution Engine output is an “updated” scene graph data structure.

5.5.7 Renderer

The input of the AHV renderer is a updated scene graph.

The output is a visual, aural or/and haptic stream of data to be fed into display devices (such as a video frame, stereo sound signal, motor command, pulse-width modulation signal for vibrators, etc.).

The MAR system can specify various capabilities of the AHV renderer, so the scene can be adapted and simulation performance can be optimized. For instance, a stereoscopic HMD and a mobile device might require different rendering performances. Multimodal output rendering might necessitate careful millisecond-level temporal synchronization.

Render Type	**Capabilities Dimensions**
Visual	Screen size, resolution, FOV (Field of View), number of channels, signal type
Aural	Sampling rate, number of channels, maximum volume
Haptics	Resolution, operating spatial range, degrees of freedom, force range

5.5.8 Display / User Interface

The input of the Display is a stream of visual, aural and/or haptics data.

The output of the UI is a set of signals to be sent to the Execution Engine in order to update the scene.

6 MAR Classification Framework

This section introduces a classification framework serving to translate abstract MAR-RM concepts into real world MAR implementations.

Table 16. MAR classification framework

Component	**Dimension**	**Types**					
Pure Sensors	**Modality**	Visual	Auditory	Electro-magnetic waves (e.g., GNSS)	Haptic/tactile	Temperature	Other physical properties
	Source Type	Live	Pre-captured	—	—	—	—
Real World	**Modality**	Visual	Auditory	Haptics	Other	—	—

Component	Dimension	Types					
Capturer				properties			
	Form of Visual Modality	Still image	2D video	3D video (video + depth)	3D mesh	Other	—
	Source Type	Live	Pre-captured	—	—	—	—
Recognizer	Form of Target Signal	Image patch	3D primitives	3D model	Location (e.g., earth-reference coordinates)	Audio patch	Other
	Form of the Output Event	Recognized or not	Additional data: Type, timestamp, recognition confidence level, other attributes	—	—	—	—
	Execution Pace	Local	Remote	—	—	—	
Tracker	Form of Target Signal	Image patch	3D primitives	3D Model	Earth-reference coordinates	Audio patch	Other
	Form of the Output Event	Spatial (2D, 3D, 6D, etc.)	Aural (intensity, pitch, …)	Haptic (force, direction, …)	—	—	—
	Execution Place	Local	Remote	—	—	—	—
Space Mapper	Space Type	Spatial	Audio	Haptics	Others	—	—
Event Mapper	Modality	Visual	Temporal	Aural	Location	Others	_
Execution Engine	Space & Time	2D + t	3D + t	—	—	—	—
	User Interactivity	Yes	No	—	—	—	—
	Execution Place	Local	Remote	Hybrid	—	—	—
	Number of Simultaneous Users	Single-user	Multi-user	—	—	—	—
Renderer	Modality	Visual	Aural	Haptics	Other	—	—

Component	Dimension	Types					
	Execution Place	Local	Remote	Hybrid	—	—	—
Visual Display	**Presentation**	Optical see through	Video see through	Projection	—	—	—
	Mobility	Fixed	Mobile	Controlled	—	—	—
	No of Channels	2D (mono)	3D stereoscopic	3D holographic	—	—	—
Aural Display	**No of Channels**	Mono	Spatial	—	—	—	—
	Acoustic Space Coverage	Headphones	Speaker	—	—	—	—
Haptics Display	**Type**	Vibration	Pressure	Temperature	—	—	—
UI	**Interaction Type**	Touch	Click	Drag	Gesture	Other	—

7 MAR System Classes

This section uses the classification framework provided in the previous section to describe several classes of mixed and augmented reality application or services. Each class focuses on parts of the system architecture and provides illustrations of how to use the reference model. The classes are defined as illustrated in Table 17. An instance of a MAR system can be a combination of several classes.

Table 17. MAR classification framework

MAR Class V	The class of MAR systems augmenting the 2D visual data captured by real camera
MAR Class R	The class of MAR systems augmenting the 2D visual data continuously captured by one or several real cameras and reconstructing the 3D environment (note: SLAM, PTAM)
MAR Class A	The class of MAR systems augmenting the aural modality
MAR Class H	The class of MAR systems augmenting the haptics modality
MAR Class G	The class of MAR systems using global position system to register synthetic objects in the real world
MAR Class 3DV	The class of MAR systems augmenting the 3D visual data captured by multiple cameras or video plus depth cameras
MAR Class 3DA	The class of MAR systems augmenting the scene by using 3D audio data

Note: Other MAR system classes may be added in the future.

7.1 MAR Class V—Visual Augmentation Systems

7.1.1 Local Recognition and Tracking

The Device detects the presence of target resources (images or their corresponding descriptors) in a video stream, optionally computes the transformation matrix (position, orientation and scaling) of the detected ones and augments the video stream with associated graphical objects. The video can be the result of a real time scene capture using a local camera, a remote video source or a video track stored in the device. The content specified in the Information viewpoint is:

- URLs to target images (compressed, raw or corresponding descriptors).
- URL to the video stream (a local camera, a local video track or a remote video resource).
- Media used for the augmentation.
- Optional: a 2d region (in the video frame) to be considered in the recognition process and delay constraints.

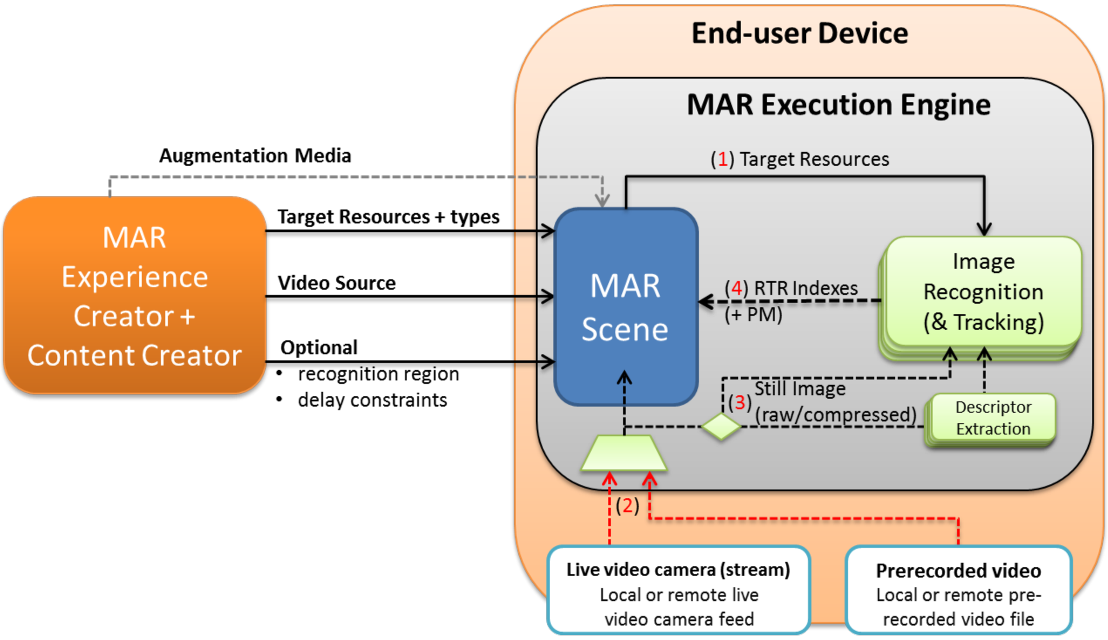

Figure 7. Local recognition (and tracking) (Pose Matrix or PM)

7.1.2 Local Registration, Remote Recognition and Tracking

The Device sends the target resources (images or their corresponding descriptors) and the video stream sampled at a specified framerate (provided by a local camera, a local video track or a remote video resource) to a Processing Server which detects and optionally tracks the target resources in the video stream. An ID mask and the computed transformation matrix of the detected resources are returned. The content specified in the Information viewpoint is:

- URLs to target images (compressed, raw or corresponding descriptors).
- URL to the video stream (a local camera, a local video track or a remote video resource) .
- The format in which the video data is sent to the processing server (raw/compressed image or the corresponding video frame descriptors).
- Media used for the augmentation.
- URL to the Processing Servers.

- Optional: a 2d region (in the video frame) to be considered in the recognition process and delay constraints.

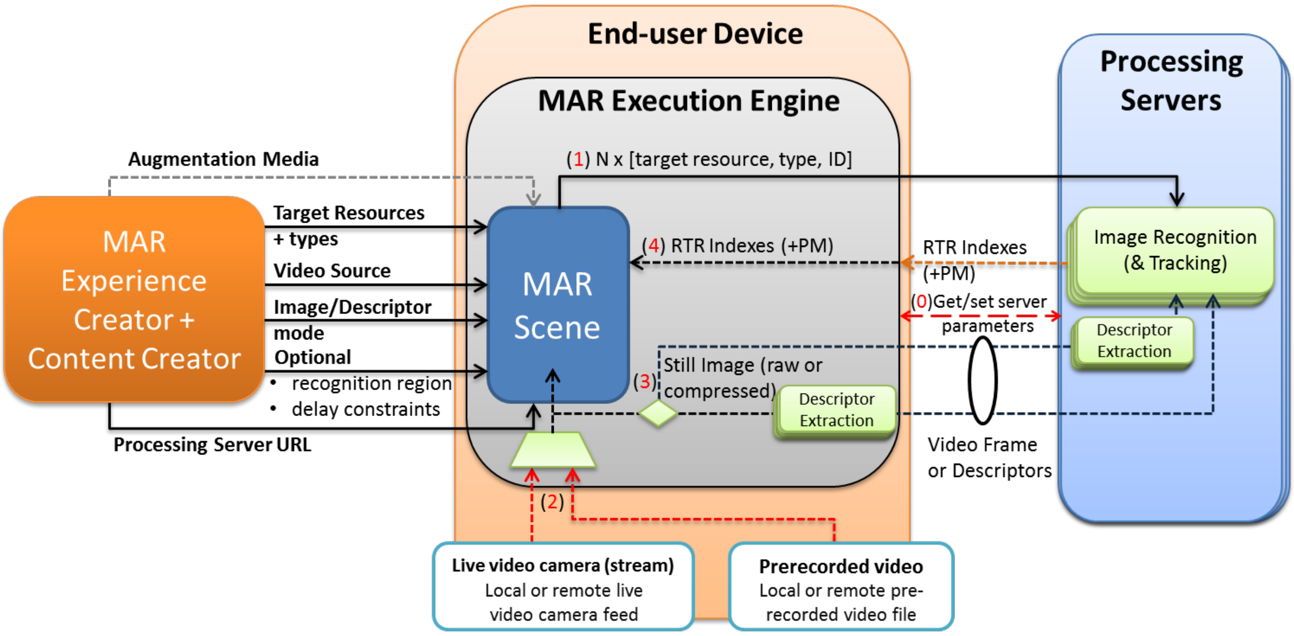

Figure 8. Local registration, remote recognition and tracking

In addition, a communication protocol has to be implemented between the MAR Execution Engine and the Processing Servers.

7.1.3 Remote Recognition, Local Tracking & Registration,

The Device sends video frames in a format that can be specified by MAR experience creator (from a local camera capture, a local video track or a remote video resource) to a Processing Server that is analysing the data and detects one or several target resources that stored in its local database. The server returns the position and size of one or several target resources detected in the frame, as well as the augmentation content (virtual objects, application behaviour). By using position and size, the device will crop the target images from the frame and use them for local tracking. The content specified in the Information viewpoint is:

- URLs of the Processing Servers.
- URL to the video stream (a local camera, a local video track or a remote video resource).
- The format in which the video data is sent to the processing server (raw/compressed image or the corresponding video frame descriptors).
- Optional: a 2d region (in the video frame) to be considered in the recognition process and delay constraints.

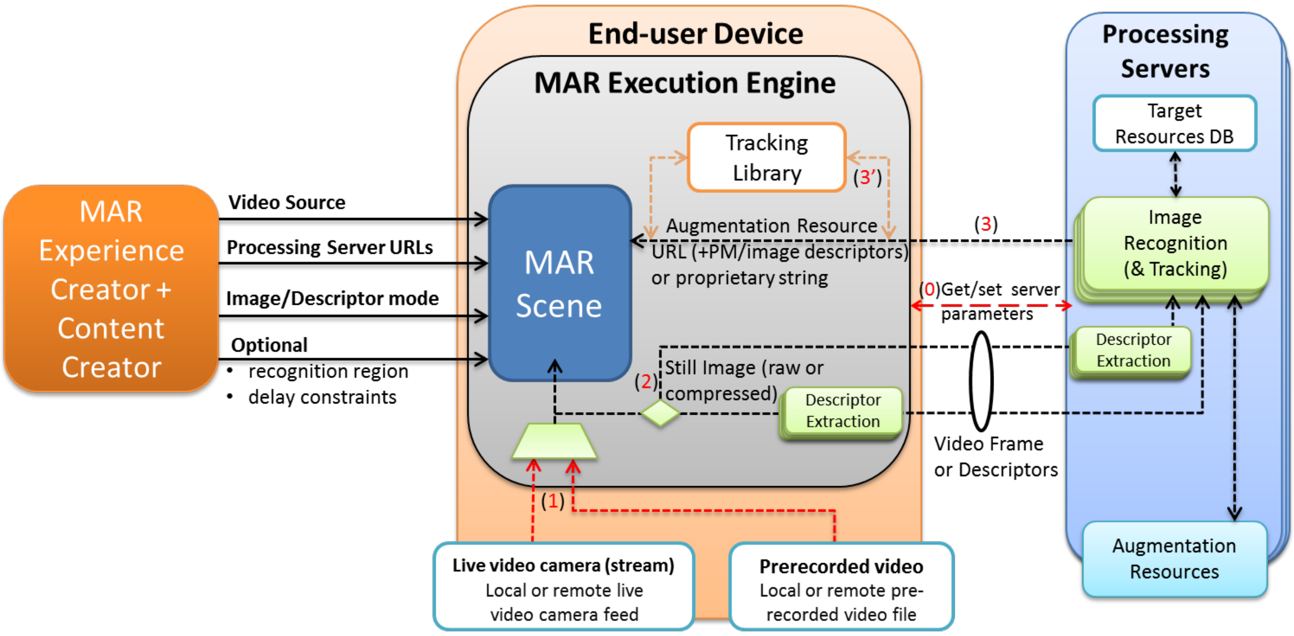

Figure 9. Remote recognition, local tracking & registration

In addition, a communication protocol has to be implemented between the MAR Execution Engine and the Processing Servers.

7.1.4 Remote Recognition, Registration and Composition

The Device sends a video stream (from a local camera capture, a local video track or a remote video resource) to a Processing Server that is analysing the data and detects one or several target resources that are stored in its local (or remote) database. Additionally, the Processing Server does the composition and rendering of the video frames, and sends back to the device the composed (augmented) video stream. The content specified in the Information viewpoint is:

- URL to the video stream (a local camera, a local video track or a remote video resource).
- URL to the Processing Servers.
- Optional: a 2d region (in the video frame) to be considered in the recognition process and the number of frames per second expected from the processing server.

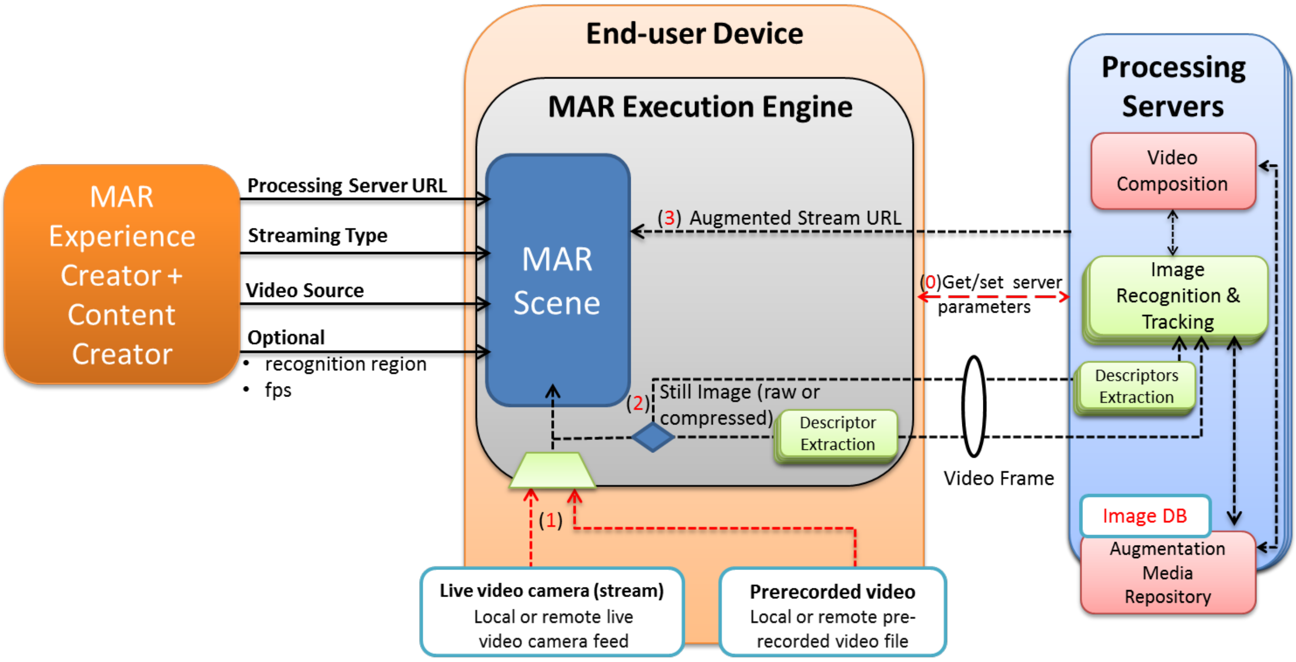

Figure 10. Remote recognition, recognition and composition

In addition, a communication protocol has to be implemented between the MAR Execution Engine and the Processing Server.

7.2 MAR Class G: Points of Interest—GNSS-based Systems

7.2.1 Content-embedded POIs

A MAR Execution Engine is used by the end-user to open a MAR file containing (locally in the scene) POIs from a specific region. The POIs are filtered with respect to user preferences as follows: Either the engine has access to a local resource (file) containing predefined POI-related user preferences or the engine exposes an interface allowing users to choose (on the fly) their preferences. The POIs corresponding to the user selections/preferences are displayed. The MAR content also describes how the POIs are displayed, either on the map or in AR view, by creating MapMarker instances and using the metadata provided by the POIs. The content specified in the Information viewpoint is:

- POI data.
- The MapMarker shape (representation).
  -referenced by the MAR Execution Engine.
- User preferences (optional).

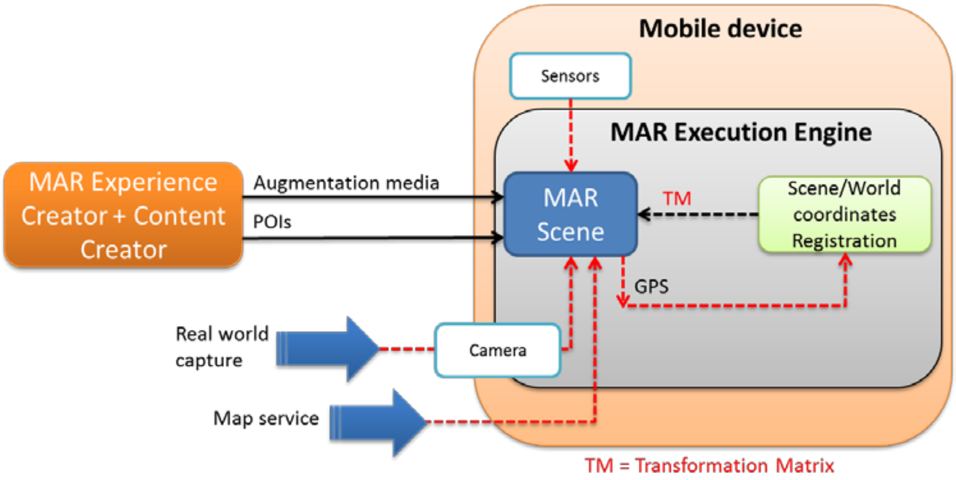

Figure 11. Content-embedded POI

7.2.2 Server-available POIs

A MAR Execution Engine is used by the end-user to open an MAR file. One or multiple URLs to POI providers are specified in the MAR content. The POIs are filtered with respect to user preferences as follows: Either the engine has access to a local resource (file) containing predefined POI-related user preferences or the engine exposes an interface allowing users to choose (on the fly) their preferences. The POIs corresponding to user selections/preferences are requested from the specified URLs and displayed. The MAR content describes how POIs are displayed, either on the map or in AR view, by creating MapMarker instances and using the metadata provided by the POIs. The content specified in the Information viewpoint is:

- URLs of the POI providers.
- The MapMarker shape/representation.

In addition, a communication protocol should be established between the MAR Execution Engine and the POI Provider.

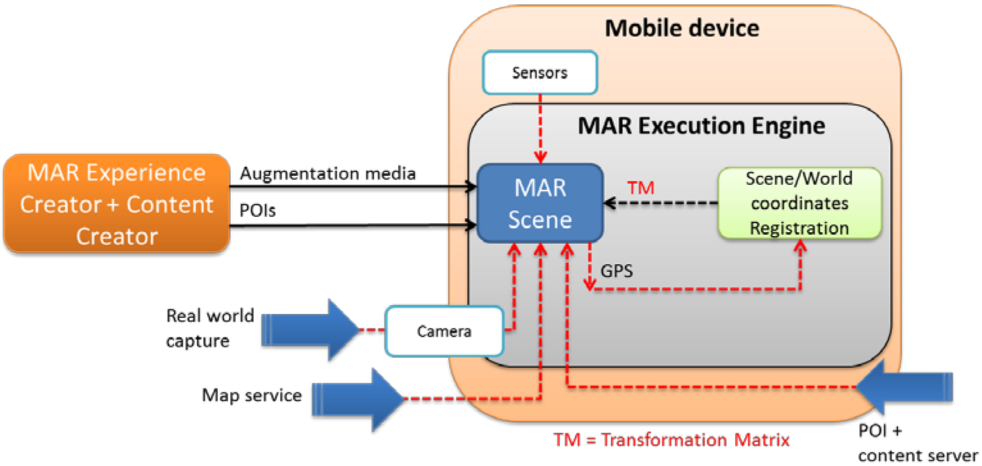

Figure 12. Server available POI

7.3 MAR Type 3DV: 3D Video Systems

7.3.1 Real Time, Local-Depth Estimation, Condition based Augmentation

Example: The end-user has an augmented reality experience where one virtual object is displayed on a horizontal plane detected within a ray of 10m.

The Device captures multi-view video and estimates depth. This representation is used to detect conditions imposed by the content designer. Once the condition is met, the Device renders the virtual object by using the scale and orientation specified by the content designer. The content specified in the Information viewpoint is:

- Media used for the augmentation.
- The orientation and scale of the virtual object (uniform/isotropic scaling representing physical units).
- The condition (e.g., horizontal plane within a ray of 10m).

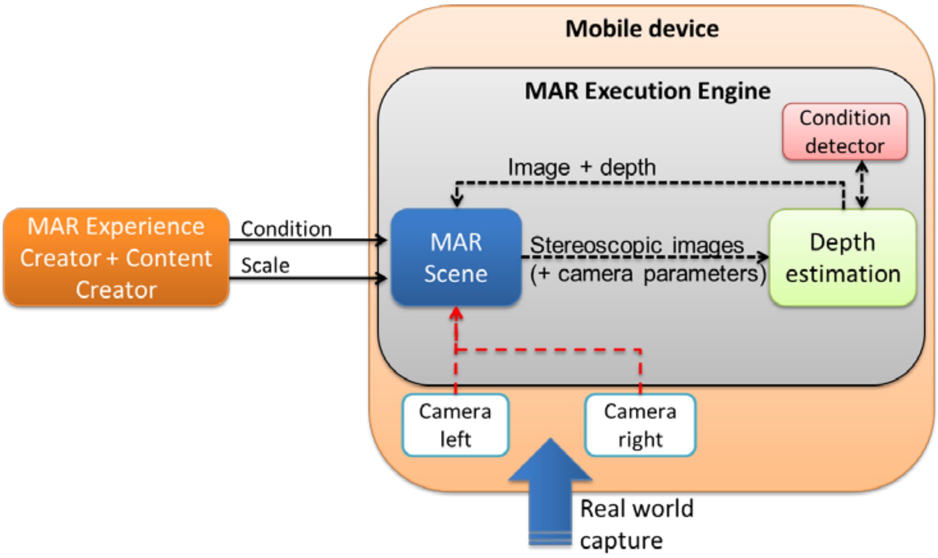

Figure 13. Real-time, local depth estimation, condition based augmentation

7.3.2 Real Time, Local-depth Estimation, Model-based Augmentation

A content designer captures offline an approximation of the real world as a 3D model and then authors content by adding additional 3D virtual objects registered within an approximation of the real world. The end-user navigates in the real world using a multi-view camera. The Device estimates the depth and computes the transformation matrix of the camera in the real world by matching the captured video and depth data with the 3D model approximating the real world. The augmented scene is therefore rendered by using the transformation matrix result. The content specified in the Information viewpoint is:

- Virtual objects and their local transformations in the scene MAR experience.
- The approximation of the 3D model of the real world.

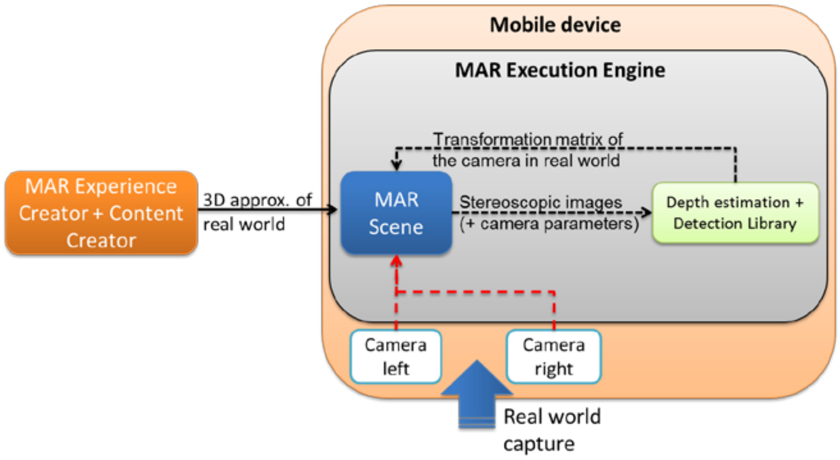

Figure 14. Real-time, local depth estimation, model-based augmentation

7.3.3 Real Time, Remote Depth Estimation, Condition-based Augmentation

Example: The end-user has an augmented reality experience where one virtual object is displayed on a horizontal plane detected within a radius of 10m.

The Device captures multi-view video, sends synchronized samples to a Processing Server that estimates the depth. This representation is sent to the device and the server uses it to detect conditions imposed by the content designer. The server sends as well the transformation matrix that the Device uses to render the virtual object by using the scale specified by the content designer. The content specified in the Information viewpoint is:

- Media used for the augmentation.
- The orientation and scale of the virtual object (uniform/isotropic scaling representing physical units).
- The condition (e.g. horizontal plane within a ray of 10m).
- URL of the Processing Server.

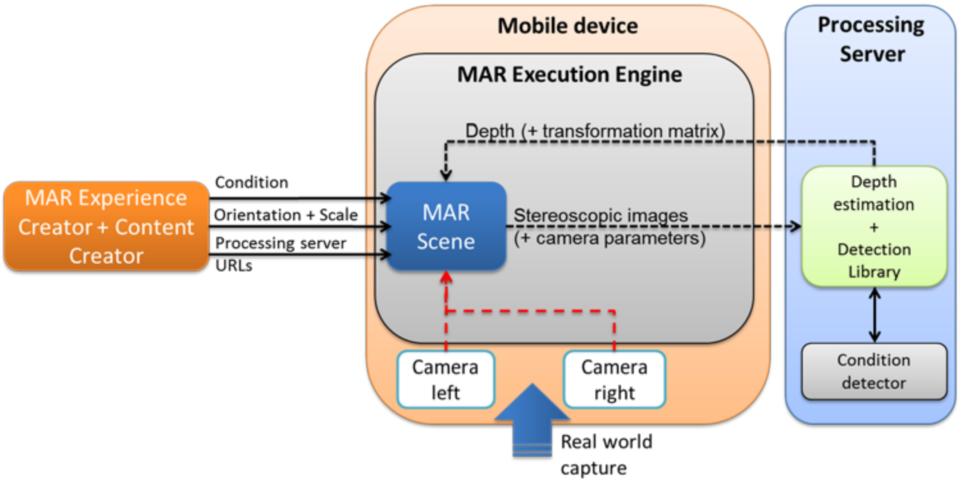

Figure 15. Real Time, remote-depth estimation, condition-based augmentation

7.3.4 Real Time, Remote-depth Estimation, Model-based Augmentation

A content designer captures offline an approximation of the real world as a 3D model and then authors content by adding additional 3D virtual objects registered within the approximation of the real world. The end-user navigates in the real world using a multi-view camera. The captured video stream is sent to the Processing Server, which computes the depth as well as the transformation matrix of the camera in the real world. Information is sent back to the Device that uses them for augmentation. The content specified in the Information viewpoint is:

- Virtual objects and their local transformations in the MAR experience.
- An approximation (3D model) of the real world.
- URL of the Processing Server.

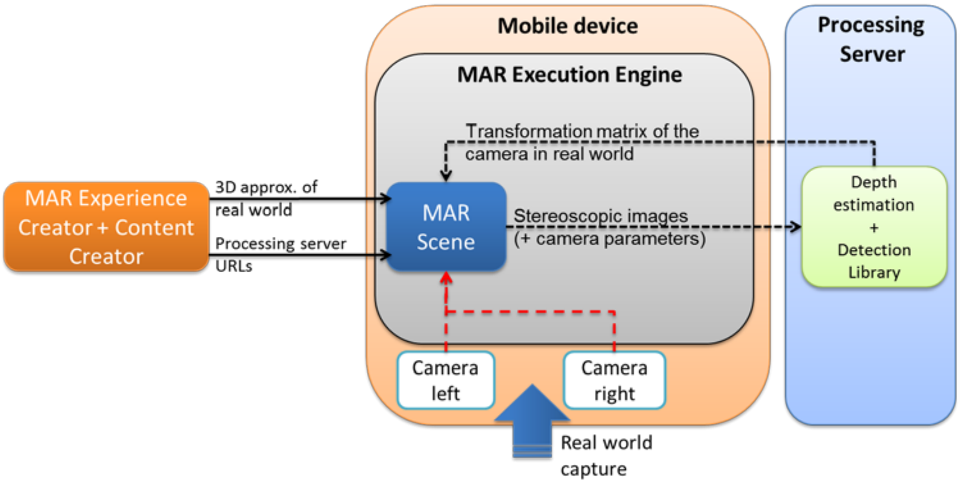

Figure 16. Real Time, remote-depth estimation, model-based augmentation

7.4 MAR Type A: Audio Systems

7.4.1 Local Audio Recognition

The Device detects the presence of a sound (or the corresponding descriptors) in an audio stream. Audio can result from a real time capture using a local microphone, a remote audio source or a pre-recorded audio stored in the device. The content specified in the Information viewpoint is:

- Media used for the augmentation.
- Target audio samples (or the corresponding descriptors).
- The URL to the audio stream (microphone, remote audio source or local track).
- Optional: the recognition frequency and the audio sequence size.

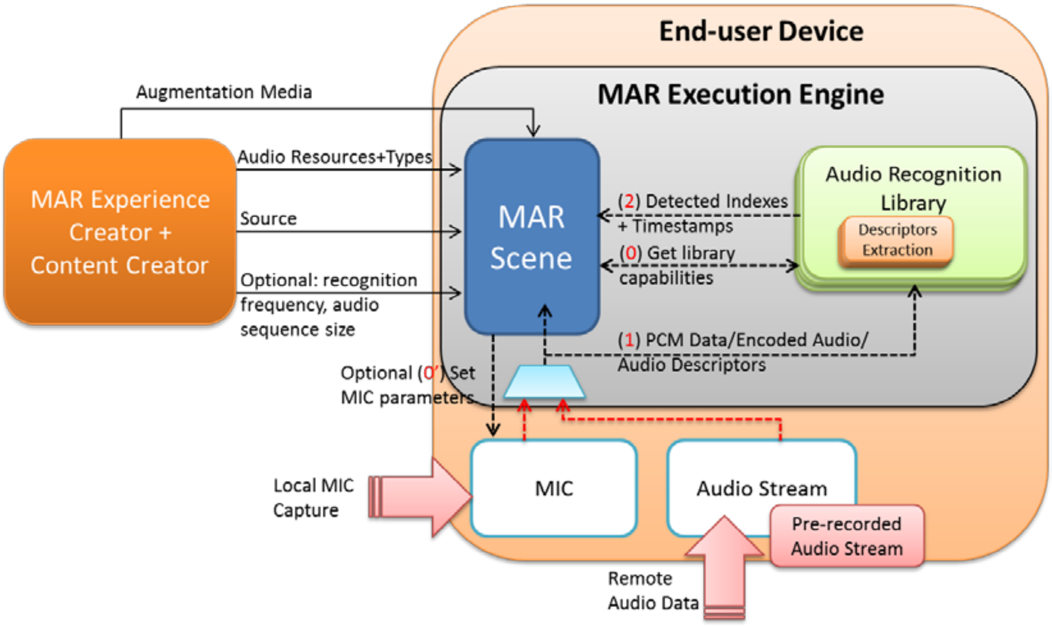

Figure 17. Local audio recognition

7.4.2 Remote Audio Recognition

The Device sends the audio stream (provided by a local microphone, a local audio track or a remote audio resource) or corresponding descriptors to a Processing Server, which detects target resources which are stored in its local (or remote) databases in the audio stream. Audio metadata, timestamps and eventually links to augmentation media of the detected resources are returned. The content specified in the Information viewpoint is:

- URL to the audio stream (local microphone, a local audio track or a remote audio resource).
- URL to the Processing Server.
- Optional: the recognition frequency and the audio sequence size.

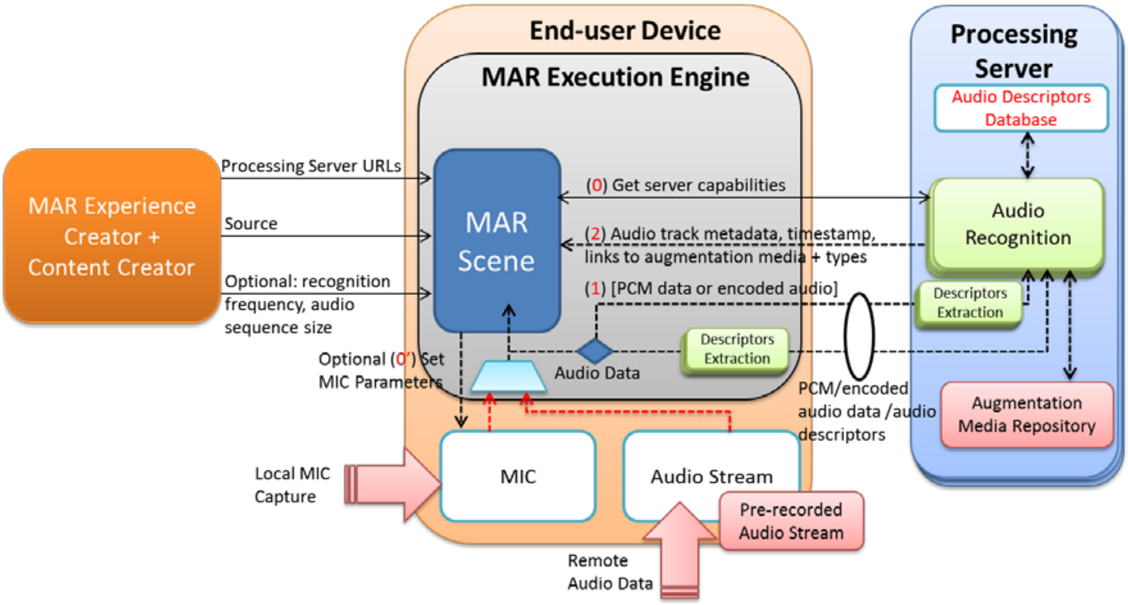

Figure 18. Remote audio recognition

In addition, a communication protocol has to be implemented between the MAR Execution Engine and the Processing Server.

7.5 MAR Type 3A: 3D Audio Systems

7.5.1 Local Audio Spatialisation

The Device computes the spatial audio data (left and right channels) by using the original audio data and the relative position between the user and the audio virtual object used for augmentation.

The content specified in the Information viewpoint is:

- Target audio samples (raw or corresponding descriptors).
- URL of the audio stream (microphone, remote audio source or local track).

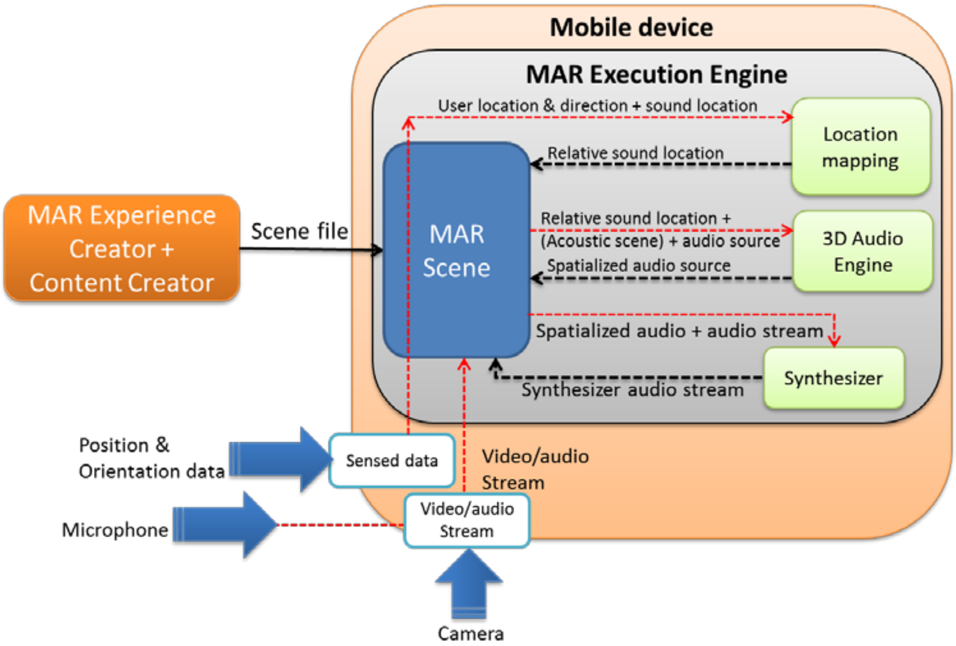

Figure 19. Local audio spatialisation

8 Conformance

Conformance to this reference model is expressed by describing how the aspects of an MAR implementation relates to the MAR system architecture. Conformance of MAR implementations to this standard shall satisfy at least the following requirements.

- The following key architectural components, as specified in the reference model, shall be present in a given MAR implementation: Display / UI, Event mapper, Recognizer, Renderer, Sensors, Execution Engine, Spatial mapper Tracker and a mapping between a MAR implementation components and the reference model components may be established and evaluated.

- The relationship between the implementation of these architectural components shall conform to those in this reference model, as specified in section 7.4 and graphically depicted in Figure 6.

- The interfaces between the architectural components of a MAR implementation shall contain and carry the information specified in sections 7.4 and 7.5. However, the specific content, format, data types, handshake, flow, and other implementation details are at the discretion of the given MAR implementation to meet its specific needs.

- The API for a MAR implementation shall conform to the concepts specified in sections 7.4.8 and 7.5 in order to ensure compatibility and software interface interoperbility between MAR implementations can be accomplished at least at the abstract API level.

9 Performance

The system performance guideline defines the minimum operational level of MAR systems and establishes possible conformance issues. There are several metrics that can be used to benchmark a MAR system, defined at various component levels or at the global level. For the latter case, augmentation precision and speed in different operating conditions are the most relevant. Specifying performance metrics is outside the scope of the MAR-RM, however, several examples are provided to be used by other benchmarking systems:

- The augmentation precision can be measured by the error between the virtual camera parameters, estimated by the tracker and the correct ones or by the distance (in pixels) and angular distance (in degrees) between the place where the virtual object is displayed and the one where it should be displayed.

- The latency can be measured as the total time needed to process the target object and produce the augmentation.

- The operating conditions may include lighting conditions, mobility of the target object, sensing distance and orientation, etc..

10 Safety

MAR systems are used by human users to interact in the real world and entail various safety issues. For example, most MAR systems require the use of special displays which may steer users attention away and create potentially dangerous situations. Minimum safety guidelines are necessary to ensure that the given MAR system and content includes components for safeguarding the user during the usage of the system. Note that the issue of performance is closely related to that of safety.

Development of policies or software that increases the safety of users, assets and systems, will reduce risks resulting from:

- Obstruction of dangerous conditions that could lead to injury of humans during MAR system use.

- Hardware necessary for MAR system operation that has not been safety certified for specific environments.

- Lack of sufficient instructions, presentations, and highlighting of information for safe and proper usage of the MAR contents,

- Non-ergonomic design of hardware (especially wearable sensors and displays) and display design (feedback) causing visual, aural and haptic fatigue, stress, discomfort and interferences.

- Distraction of attention from potential hazards in the real world.

- Temporary disconnection of the network service causing false confidence in the currently presented information.

- Not considering special operational safety and health (OSH) requirements (e.g. such as in construction zones, traffic, operating vehicles, working at height in proximity to hazards, etc.).

- Human movements necessary for operating a MAR system.

- Insufficient level of performance for requirements of MAR system-assisted tasks.

- Accessibility issues for users with impairments deserve special consideration

- Sickness from mismatched stimuli to the human vestibular system. restricted field of view, and other potential factors. Disruptive effects may in turn lead to disorientation, nausea, blurred vision, loss of spatial acuity, and multiple other symptoms. These symptoms may last even after a user is no longer immersed in the MAR systems and services.

11 Security

Most MAR system services and implementations, like many other modern information systems, often rely on network based solutions and are prone to the usual information security problems. Even as a stand-alone system many MAR applications and services, by nature, tend to deal with a lots of personal information, therefore pose as an attractive target for security attacks. In general, MAR systems should exhibit a level of security (for its contents and information) comparable to other digital contents services such as web documents/systems (http://www.w3c.org/Security/) and geospatial systems (http://www.opengeospatial.org/projects/groups/securitywg).

In particular, the MAR-RM should outline the minimum set of features and components for architects and developers to consider for the sake of general security:

- Encrypt digital assets.
- Encrypt sensor readings captured by MAR systems.
- Encrypt other communications between MAR components.

12 Privacy

Personal privacy and potential exposure of personal information to unauthorized systems or third parties via cameras or other sensors on the MAR-assisted device is out of scope of the MAR-RM but is highly relevant to the adoption of MAR systems. Developers may consider how to use existing or new systems and include components in their MAR systems that:

- Authenticate user identity (e.g., registration with an account).
- Authorize system access to users' personal data.
- Define the duration of periods during which data access and/or storage is authorized.

Annex A
(informative)

Patent Statements

The International Organization for Standardization and the International Electrotechnical Commission (IEC) draw attention to the fact that it is claimed that compliance with this part of ISO/IEC 18039 may involve the use of patents.

ISO and IEC take no position concerning the evidence, validity and scope of these patent rights.

The holders of these patent rights have assured the ISO and IEC that they are willing to negotiate licences under reasonable and non-discriminatory terms and conditions with applicants throughout the world. In this respect, the statements of the holders of these patents right are registered with ISO and IEC. Information may be obtained from the companies listed below.

Attention is drawn to the possibility that some of the elements of this part of ISO/IEC 18039 may be the subject of patent rights other than those identified in this annex. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

The contributors to this standard informally have declared that they are not aware of any patent nor royalty related materials used in this standard (as indicated in the following table). Recipients of this draft are invited to submit, with their comments, notification of any relevant patent rights of which they are aware and to provide supporting documentation.

Company	**Address**
none	

Annex B
(informative)

Use Case Examples[3] and Coverage by the MAR reference model

B.1 Introduction

This section introduces use case categories and examples for MAR, and for each example provides the mapping to the MAR system architecture and corresponding viewpoints.

B.2 Use Case Categories

B.2.1 Guide Use Case Category

The simplest and most fundamental use case category is Guide. In the Guide type of experience, a user points sensors at a target physical object (or in a direction) and queries the system. The system provides a user interface for virtual objects about which a person asks one or more questions, often in a sequence. Experiences in the Guide use case category often leads user in learning, completing a task or arriving at a destination (navigation).

B.2.2 Publish Use Case Category

The Publish use case category permits a user to “author” a new virtual object in the form of text, image, video or audio and to attach this user-generated information to a real physical object target. The user expresses an opinion, provides additional thoughts or asks questions, and other people with permissions to access the virtual object will be able to see, hear or feel it.

12.1.1 Collaborate Use Case Category

The Collaborate Use Case category encompasses all use cases in which there is the physical world, digital assets and two or more users interacting with one another in real time. In Collaborate, there is no prior limit to where users are in the physical world with respect to one another.

A specific Collaborate use case can specify the distance between users (proximity) in meters. Other use cases can specify categories of objects that constitute the focus of attention. For example, there are use cases in this category involving manufacturing, repair, maintenance of machinery, infrastructure or some stationary, man-made object. Other use cases in this category are multi-player AR-assisted games.

B.3 MagicBook (Class V, Guide)

12.1.2 What it Does

MagicBook [4] is a marker-based augmented reality system. Animated 3D models and other types of virtual objects are added to the printed book content. It helps convey information that is difficult to express solely with print. In addition, it allows a transition into a pure VR mode.

[3] Some of the use cases are commercial solutions and may have related intellectual property rights.

Figure 20. A user is viewing the MagicBook using a hand-held video see through display.

12.1.3 How it Works

A marker is printed in a book and viewed using a video see-through display as illustrated in the Figure above. The marker is recognized and tracked by the camera attached to the display.

12.1.4 Mapping to MAR-RM and its Various Viewpoints

MAR-RM Component	**Major Components in the MagicBook**
Sensor	Camera
Real-world capture	Live video
Target physical object	2D Marker
Tracker / Recognizer	Template-based recognition/homography-based 2D marker tracking
Spatial mapping	Hard coded
Event mapping	Hard coded
Execution Engine	Hard coded
Rendering	OpenGL
Display / UI	Video see-through and headphone

B.4 Human Pac-man (Type G, Collaborate) and ARQuake (Class V and G, Collaborate)

12.1.5 What it Does

Human Pac-man [5] is an outdoor interactive entertainment system in which the video game Pac-man (developed by Namco in 1980) is played outdoors with humans acting as pacmen and ghosts. Virtual cookies are overlaid in the physical environment. ARQuake [6] is an outdoor interactive entertainment system developed in 2000 using markers.

Figure 21. The view of Human Pac-man as seen by the user is shown: (left) AR Quake: a first-person outdoor and (right) AR game using a marker instead.

12.1.6 How it Works

The user wears a head-mounted display whose location is tracked by a GPS. In Pacman, virtual cookies appear properly registered in the real world and are also mapped by their GPS coordinates. Users interact with the virtual cookies and other users (e.g., ghosts) and have a similar behaviour as in the conventional "Pac-Man" game.

12.1.7 Mapping to MAR-RM and Various Viewpoints

MAR-RM **Component**	Major **Components** in Human Pacman	Major **Components** in AR Quake
Sensor	Camera/GNSS	Camera/GNSS/compass
Real-world capture	Live video	Live video
Target physical object	Location	Location/direction/marker
Tracking/recognition	GNSS	Camera/GNSS/compass
Spatial mapping	Hard-coded earth referenced	Hard coded
Event mapping	Hard coded	Hard coded
Simulation Engine	Hard coded	Hard coded
Rendering	Generic graphic subsystem	Quake game ported
Display/UI	Video see-through, head phone, hand-held keyboard and mouse and other touch sensors	Video see-through, head phone, button device

B. 5 Augmented Haptics – Stiffness Modulation (Class H, Guide)

12.1.8 What it Does

In this use case, a user feels the response force of an object as well as the augmented response force of a virtual object. It can be used for instance in training for cancer palpation on a dummy mannequin [7].

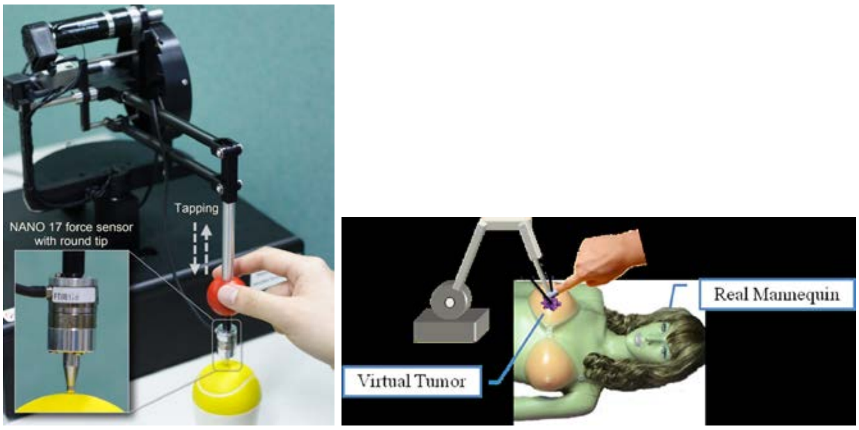

Figure 22. The stiffness is modulated in which haptic information is augmented.

12.1.9 How it Works

A manipulator-type haptic device is used to sense and capture the force from a real object. Both the haptic probe and user's hand are mechanically tracked. A collision with a virtual object is simulated and its added reaction force is computationally created and displayed through the haptic probe.

12.1.10 Mapping to MAR-RM and Various Viewpoints

MAR-RM Component	**Major Components in the Augmented Haptics**
Sensor	Force and joint sensors on the haptic manipulator
Real-world capture	Force sensor
Target object	Any 3D physical object
Tracker	Joint sensor on the haptic manipulator and kinematic computation
Recognizer	No recognition
Spatial mapping	Hard coded
Event mapping	Hard coded
Simulation Engine	Hard coded
Rendering	In-house force rendering algorithm
Display / UI	Haptic manipulator

B. 6 Hear Through Augmented Audio (Class A, Guide)

12.1.11 What it Does

Composition of real world sound and computed generated audio [8].

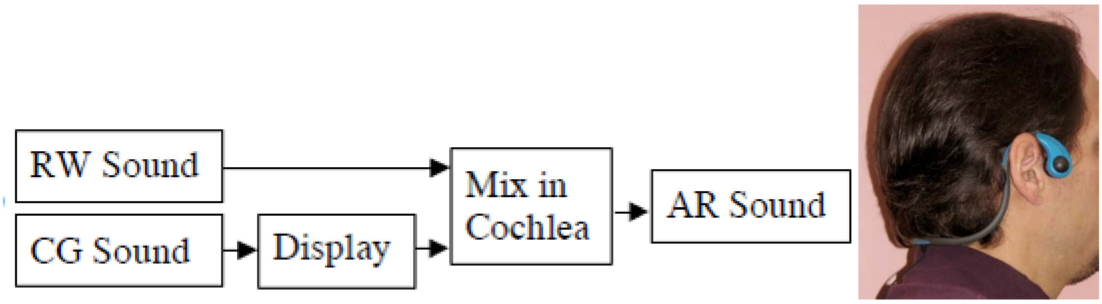

Figure 23. Hear-through augmented audio uses a bone-conducting headset.

12.1.12 How it Works

A bone-conduction headset is used to add augmented sound to real-world sound. It is considered a “hear through” because the augmented media is merged and perceived by the human rather than as a result of a computed composition.

12.1.13 Mapping to MAR-RM and Various Viewpoints

MAR-RM Component	Major Components in the Augmented Audio
Sensor	None
Real-world capture	Direct capture by human ear
Target object	Real-world sound
Tracking/Recognition	Hard coded
Spatial mapping	Hard coded
Event mapping	None
Simulation Engine	None
Rendering	HRTF-based rendering of 3D sound
Display	Bone-conduction headset

B.7. CityViewAR on Google Glass (Class G, Guide)

12.1.14 What it Does

CityViewAR [9] is a mobile outdoor AR application providing geographical information visualization on a city scale. It was developed in Christchurch, New Zealand, which was hit by several major earthquakes in 2010 and 2011. The application provides information about destroyed buildings and historical sites that were affected by the earthquakes.

Figure 24. CityViewAR as seen through optical see through Google glass is shown.

12.1.15 How it Works

Geo-located content is provided in a number of formats including 2D map views, AR visualization of 3D models of buildings on-site, immersive panorama photographs, and list views. GPS-based tracking is implemented on Android-based smartphone platforms and is displayed through an optical see-through Google Glass.

12.1.16 Mapping to MAR-RM and Various Viewpoints

MAR-RM Component	Major Components in the CityViewAR
Sensor	GNSS and compass
Real-world capture	None
Target physical object	Location
Tracker/recognizer	GNSS and compass
Spatial mapping	Absolute earth reference
Event mapping	Hard coded for location and direction
Simulation Engine	Hard coded
Rendering	Text and image
Display	Optical see through

B. 8 Diorama—Projector-based Spatial Augmented Reality (Class 3DV, Publish)

12.1.17 What it does

The Diorama [10] is a spatially augmented reality system for augmenting movable 3D objects in an indoor environment by using multiple projectors. The augmentation is made directly on the target physical object.

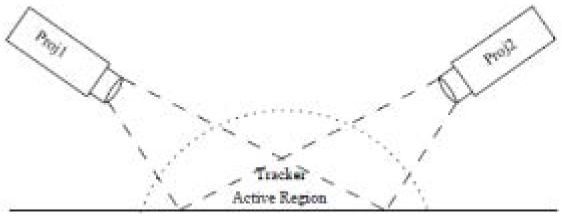

Figure 25. The user interacts with a house-shaped object, which is augmented with a different painting.

12.1.18 How it Works

A projector illuminates augmentation on an object that is tracked by an optical tracker. A user applies a paintbrush that is also tracked. The paintbrush is used to create an image that is projected on the physical object after a calibration process. The geometry of the target physical object is known in advance.

12.1.19 Mapping to MAR-RM and Various Viewpoints

MAR-RM Component	Major Components in the Diorama

Sensor	Optical tracker
Real-world capture	None
Target object	A preconfigured 3D physical object
Tracking/Recognition	Optical tracker
Spatial mapping	Hard coded
Event mapping	Hard coded/user input(s)
Simulation Engine	None
Rendering	Image
Display	Projector

B. 9 Mobile AR with PTAM (Class 3DV, Guide)

12.1.20 What it Does

PTAM [11] is a mobile AR application that augments the environment, based on tracking and mapping natural features such as 3D points.

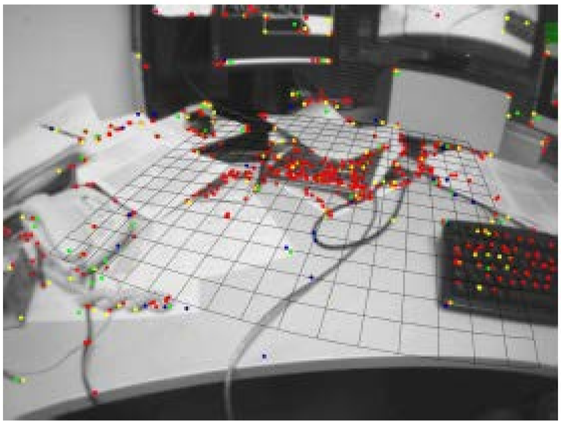

Figure 26. PTAM tracks and maps 3D point features and augmenting on a virtual plane.

12.1.21 How it Works

A simplified single camera based the SLAM [12] algorithm is used. The tracking and mapping tasks are split in order to operate in parallel threads. One thread deals with the task of robust tracking of the mobile device, while the other constructs a probabilistic 3D point map from a series of video frames through cycles of prediction and correction.

12.1.22 Mapping to MAR-RM and Various Viewpoints

MAR-RM Component	**Major Components in the PTAM**
Sensor	Camera

Real-world capture	Live video
Target object	Particular 3D points in the environment
Tracker/Recognizer	Image-processing software
Spatial mapping	Relative to a calibration object
Event mapping	None
Simulation Engine	Hard coded
Rendering	OpenGL ES
Display	Mobile phone screen

B. 10 KinectFusion (Class 3DV, Guide)

12.1.23 What it Does

KinectFusion [13] is a system for accurate real time mapping of complex and arbitrary indoor scenes and objects, using a moving depth camera. Using reconstructed 3D information about the environment, a more effective augmentation is possible for solving the occlusion problem and enabling physical simulations (e.g., rendering augmentation behind real objects).

Figure 27. Examples show: capturing the depth image of a scene, performing its 3D reconstruction (above) and carrying out physical simulation with thousands of virtual balls.

12.1.24 How it Works

The depth data streamed from a movable Kinect depth sensor are fused into a single global surface model of the observed scene in real time. At the same time, the current sensor pose is obtained by tracking the live depth frame relative to the global model using a coarse-to-fine iterative closest-point algorithm.

12.1.25 Mapping to MAR-RM and Various Viewpoints

MAR-RM Component	**Major Components in the KinectFusion**
Sensor	Kinect (depth sensor and video camera)
Real-world capture	Live 3D video
Target object	3D physical object
Tracking/Recognition	Depth image processing software
Spatial mapping	None
Event mapping	None
Simulation Engine	Hard coded
Rendering	Generic graphics engine
Display	Any

B. 11 Use Case ARQuiz

12.1.26 What it Does

ARQuiz [14] is an augmented reality location-based game for smartphones and tablets, based on the MPEG ARAF standard. The game consists of answering questions related to physical places that are being visited by players. Each question has an associated hint. A hint is a short story that eventually helps the player to find the correct answer to a given question. All hints are placed on a specific route that the user has to follow in order to advance in the game. The hints are visible on a map (MAP view). As every hint is represented by the same icon on the map, the player does not know which hint corresponds to his current question. This is the reason why the player has to use the AR view where the hints are represented by numbered 3D spheres. Once the correct number has been identified (it is visible in the AR view), the player has to “catch” that sphere by getting closer to it (5 meters or less). When close enough to the hint, the corresponding story is displayed on the device’s screen. By reading the story, the player eventually finds the correct answer of the question but may also discover more details about the place he or she is visiting at that specific moment.

Figure 28. Snaps of from the AR Quiz game are shown.

12.1.27 How it Works

A smartphone on which an ARAF Browser is installed needs to be used to open and run the game file. A GPS[4] signal and an Internet connection are needed in order to play the game. The game is an MP4 file that encapsulates the textual data (ARAF-compliant file), as well as additional media that is linked in the game file. The ARAF-compliant file consists of BIFS content in which the interface and the logic of the game are described. Being an AR location-based game, the ARAF Browser needs to access GPS data, as well as the orientation sensors of the device in order to place the virtual objects at the correct positions in the augmented reality (camera) view.

12.1.28 Mapping to MAR-RM and Various Viewpoints

MAR-RM Component	Major Components in the ARQuiz
Sensor	GNSS and orientation
Real-world capture	Live video
Target physical object	Location
Tracker/recognizer	GNSS and orientation
Spatial mapping	Absolute earth reference
Event mapping	Hard coded for location
Simulation Engine	Hard coded
Rendering	Text, images and 3D graphics
Display	Mobile phone screen

[4] The term GPS is specific to the United States' GNSS system, the NAVSTAR Global Positioning System. As of 2013, the (GPS) is fully operational GNSS.

B.12 Use case Augmented Printed Material

12.1.29 What it Does

Augmented Printed Material [15] is an AR application that enriches (physical) printed material with any digital media-like videos, images, sounds or 3D graphics. The application presents the user additional information related to the printed material that he or she is reading. An Augmented Printed Material application can enrich anything from a simple book to a tourism guide, a city map or a newspaper.

Figure 29. Examples show of augmenting printed contents.

12.1.30 How it Works

The application uses ARAF-compliant content that can be read by any ARAF Browser. The user opens the application and starts pointing the camera to the pages of the printed material that he or she is reading. Digital representations (e.g., screenshots) of the printed material pages to be augmented and their corresponding augmentation media are used as input to the application. As long as the application's scanning mode is active, camera frames are processed and matched against target images submitted by the Content creator. Once an image has been recognized, a tracking algorithm is then used to compute the relative position (pose matrix) of the recognized target image in the real world. The application is programmed to overlay the associated augmentation media on top of the recognized physical material as long as the physical printed material is being tracked.

12.1.31 Mapping to MAR-RM and Various Viewpoints

MAR-RM Component	**Major Components in the Augmented Printed Material**
Sensor	Live sensed camera data
Real world capture	Live video
Target physical object	Printed material
Tracker/recognizer	Image processing software (image recognition and tracking)
Spatial mapping	Relative to a recognized object

Event mapping	Hard coded
Simulation Engine	Hard coded
Rendering	Text, images and 3d graphics
Display	Mobile phone screen

B.13 Augmented Reality for Laparascopic Surgery (Medical Application)

12.1.32 What it Does

Laparoscopic surgery is a preferred way of surgical operations due to the minimal incisions made to the body. However, it is at the same time a difficult procedure due to the limited view the doctor is able obtain through the miniature camera inserted through the incision hole. The presented use case developed by researchers from the University of North Carolina, Chapel Hill, uses augmented imagery, that is, graphically reconstructed internal part of the body spatially registered into the patient's body (near the incisino hole) to provide better guidance for doctors in performing the laproscopic surgery (seen by a head mounted display).

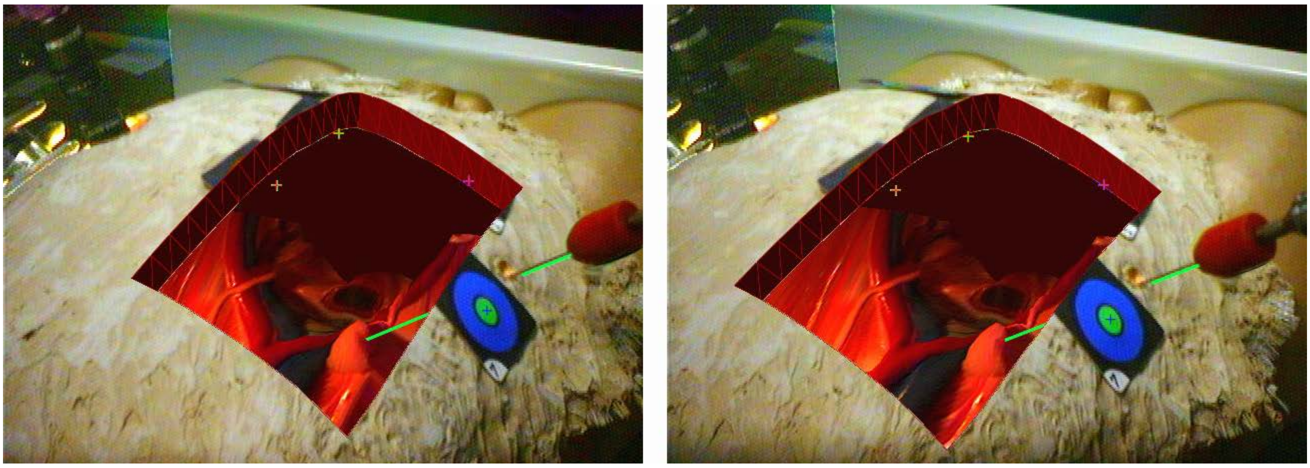

Figure 30. Patient's internals is augmented near the incision point for guiding the laparoscope operation [Fuchs].

12.1.33 How it Works

In order to acquire the 3D models of the patients internal organs near the incision point, a 3D laproscope has been developed. It is equipped with a light projector on its end and uses the structured light method to extract and recover the 3D structure. Note that this 3D laparoscope is equipped with a 6DOF tracker so that the extracted 3D model is also registered with respect to the patents body and doctor's view point/direction.

12.1.34 Mapping to MAR-RM and Various Viewpoints

MAR-RM Component	Major Components in the Augmented Printed Material
Sensor	Live sensed camera data
Real world capture	Live video, 3D reconstruction from the structured light on the laparoscope, Live laparoscope

	video camera.
Target physical object	(Dummy) patient
Tracker/recognizer	Optical tracker (installed on the ceiling) and fiducials
Spatial mapping	Relative to a recognized object
Event mapping	Hard coded
Simulation Engine	Hard coded
Rendering	Text, images and 3D graphics
Display	HMD

[Fuchs] Augmented Reality Visualization for Laparoscopic Surgery, Fuchs et al, MICCAI, 1998

Annex C
(informative)

Existent AR-related solutions/technologies and their Application to the MAR reference model

C. 1 MPEG ARAF

The Augmented Reality Application Format (ARAF) is an ISO standard published by MPEG and can be used to formalize a full MAR experience. It consists of an extension of a subset of the MPEG-4 Part 11 (Scene Description and Application Engine) standard, combined with other relevant MPEG standards (MPEG-4 Part 1, MPEG-4 Part 16, MPEG-V) and is designed to enable the consumption of 2D/3D multimedia, interactive, natural and virtual content. About two hundred nodes are standardized in MPEG-4 Part 11, allowing various kinds of scenes to be constructed. ARAF refers to a subset of these nodes. The data captured from sensors or used to command actuators in ARAF are based on ISO/IEC 23005-5 data formats for interaction devices (MPEG-V Part 5). MPEG-V provides an architecture and specifies associated information representations to enable the representation of the context and to ensure interoperability between virtual worlds. Concerning mixed and augmented reality, MPEG-V specifies the interaction between the virtual and real worlds by implementing support for accessing different input/output devices, e.g., sensors, actuators, vision and rendering and robotics. The following sensors are used in ARAF: Orientation, position, acceleration, angular velocity, GPS, altitude, geomagnetic and camera.

The ARAF concept is illustrated in the figure below. It allows the distinction between content creation (using dedicated authoring tools) and content “consumption” (using platform-specific AR browsers). Authors can specify the MAR experience by only editing the ARAF content.

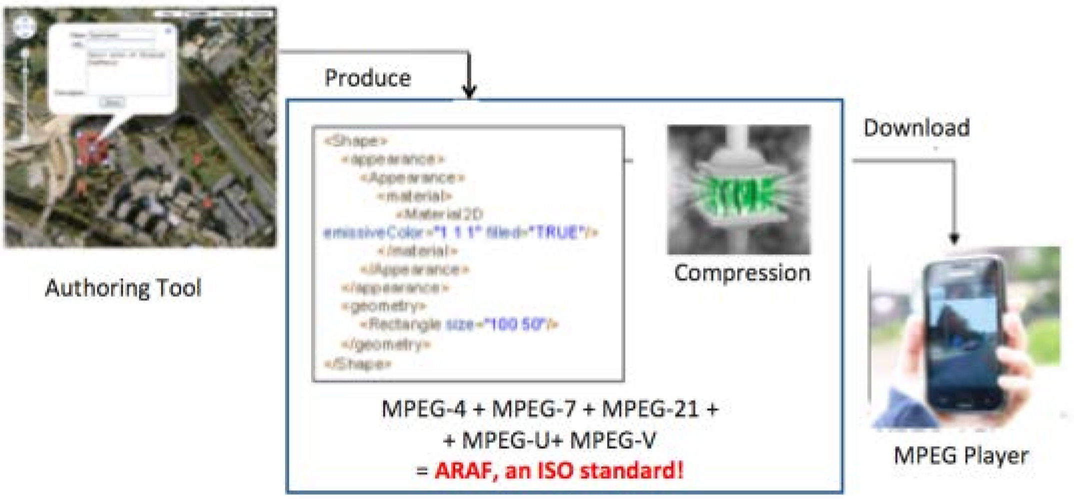

Figure 31. The concent of AR application format as proposed by the SC 29 WG 11.

By using ARAF, Content creators can design MAR experiences covering all classes defined in Section 9, from location-based services to image-based augmentation, from local to cloud-assisted processing. ARAF also supports natural user interaction(s), 3D graphics, 3D video and 3D audio media representation, as well as a variety of sensors and actuators.

C.2 KML/ARML/KARML

KML (Keyhole Mark-up Language) [16] offers simple XML-based constructs for representing a physical GPS (2D) location and associating text descriptions or 3D model files to it. KML has no further sensor-related information, and thus the event of location detection (whichever way it is found by the application) is automatically tied to the corresponding content specification. KML is structurally difficult to be extended for vision-based AR (which requires a 3D scene graph-like structure) and more sophisticated augmentation can be added only in an ad hoc way.

ARML (AR Mark-up Language) [17] is an extension to KML and allows for richer types of augmentation for location-based AR services. KARML [18] goes a bit further by adding even more decorative presentation styles (e.g., balloons, panoramic images, etc.), but more importantly, it proposes a method of relative spatial specification of the augmented information for their exact registration. These KML-based approaches use OGC standards for representing GPS landmarks, but for the rest, they use a mixture of non-standard constructs, albeit being somewhat extensible (perhaps in an ad hoc way and driven mostly by specific vendor needs), for augmentation (e.g., vs. HTML or X3D).

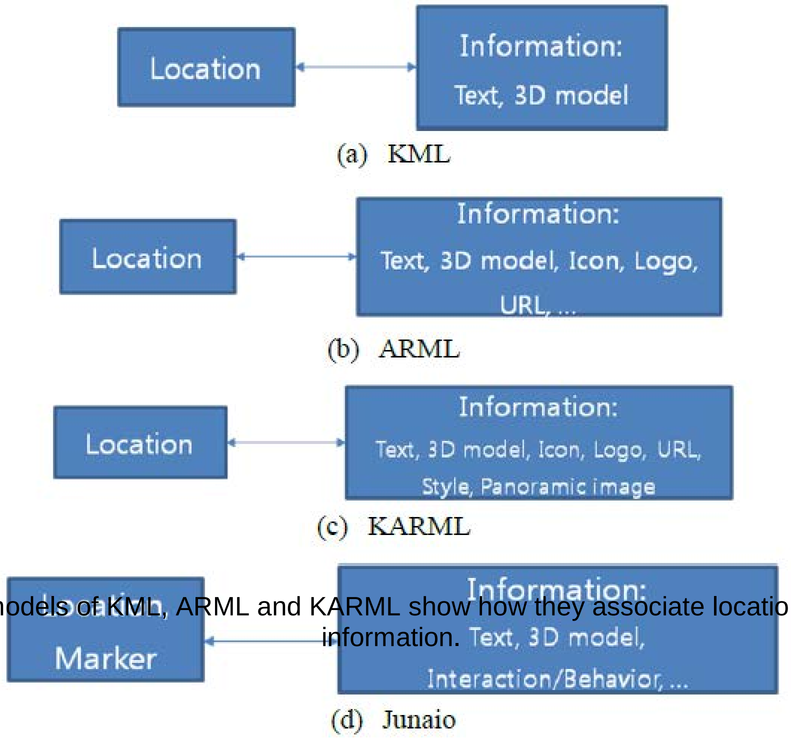

Figure 32. Content models of KML, ARML and KARML show how they associate location with augmentation information.

C.3 X3D

X3D [19] is a royalty-free ISO standard XML-based file format for representing 3D computer graphics. It is a successor to the Virtual Reality Modelling Language (VRML). X3D features extensions to VRML (e.g., CAD, Geospatial, Humanoid animation, NURBS, etc.), the ability to encode the scene graph using an XML syntax as well as the Open Inventor-like syntax of VRML97, or binary formatting, and enhanced APIs. In essence, it can be used to represent a 3D virtual scene with dynamic behaviours and user interaction(s).

X3D is originally developed to represent synthetic and 3D graphical virtual objects and scene, however, can also be naturally extended for MAR, because MAR systems, are implemented as a virtual reality systems. For example, video see-through AR systems are implemented with designating the virtual viewpoint as that of the background (real world) capture camera and rendering the augmentation objects and background video stream (as a moving texture) in the virtual space.

In 2009 an X3D AR working group has been set up to extend its capability for MAR functionalities. These include additional constructs and nodes for representing live video, physical and virtual camera properties, ghost objects, MAR events and MAR visualization.

C.4 JPEG AR

The JPEG AR describes a mechanism of JPEG image-based AR at an abstraction level, without specifying the syntaxes and protocols. Currently, there are three interest points in JPEG AR frameworks: interface, application description and JPEG file format.

For the interface, there are four main perspectives that are taken into account:

- Interface between the Sensor and AR Recognizer/AR Tracker
  - For this interface, this International Standard specifies information that needs to be transmitted from the Sensor to the Recognizer/Tracker.
- Interface between the AR Recognizer/AR Tracker and the Event Handler
  - For this interface, this International Standard specifies data and information that needs to be composed in the Recognizer/Tracker and transmitted to the Event Handler. This transmitted data and information is necessary for the Event Handler to process described operations according to the information.
- Interface between the Event Handler and the Content Repository
  - For this interface, this International Standard specifies information and corresponding operations that the Event Handler and Content Repository manipulate.
- Interface between the Event Handler and Renderer
  - For this interface, this International Standard specifies information that is transmitted from the Event Handler to the Renderer for displaying composite images.

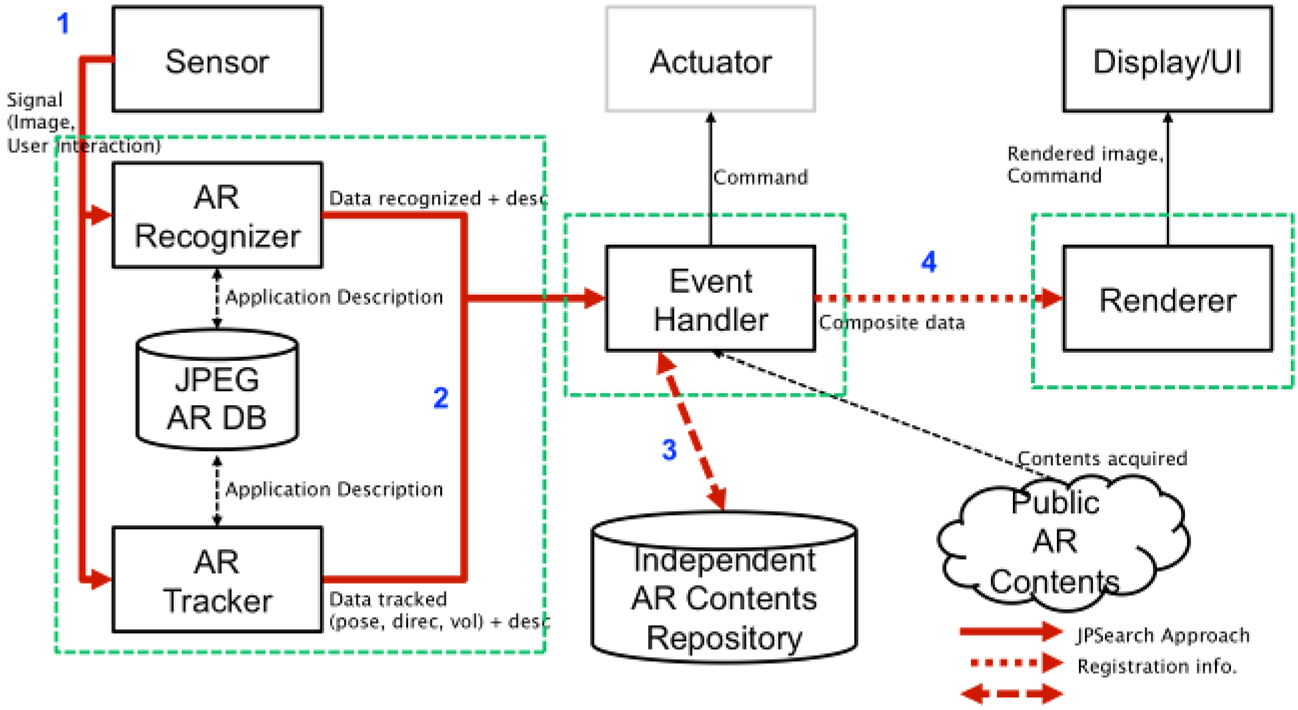

Figure 33. The JPEG AR Framework Architecture illustrates its main components.

C.6 ARToolKit/OSGART

ARToolKit [20] is a computer tracking library for the purpose of creating augmented reality applications that overlay virtual objects in the real world. It uses video tracking capabilities that calculate the real camera position and orientation relative to square physical markers in real time. Once the real camera position is known, a virtual camera can be positioned at the same point and 3D computer graphics models can be precisely overlaid on the real marker. ARToolKit provides solutions for two of the key problems in augmented reality: viewpoint tracking and virtual object interaction. ARToolkit, which by itself does not have any scene graph support, has been merged with an open source virtual reality platform (with scene graph support), namely the OpenSceneGraph (OSG) [21]. This version of OSG is called the OSGART.

C.9 Opens/OpenVX

OpenCV (Open Source Computer Vision) [22] is a library of programming functions mainly aimed at real time computer vision, developed by the Intel Russia research center in Nizhny Novgorod, and now supported by Willow Garage and Itseez. It is free for use under the open source BSD license. The library is cross-platform. It focuses mainly on real time image processing. If the library finds Intel's Integrated Performance Primitives on the system, it will use these proprietary optimized routines to accelerate itself. As a basic library for computer vision, it is often used as a means for implementing many MAR systems and contents.

The KHRONOS group has developed a similar standard for such a computer vision library called OpenVX, which lends itself to hardware acceleration and higher performance.

C.10 QR Codes / Bar Codes

QR code (abbreviated from Quick Response Code) is the trademark for a type of matrix barcode (or two-dimensional barcode) first designed for the automotive industry in Japan. A barcode is a machine-readable optical label that contains information about the item to which it is attached. A QR code uses four standardized encoding modes (numeric, alphanumeric, byte / binary, and kanji) to efficiently store data; extensions may also be used.

The QR Code system became popular outside the automotive industry due to its fast readability and greater storage capacity compared to standard UPC barcodes. Applications include product tracking, item identification, time tracking, document management, and mixed and augmented reality as well (as a marker).

A QR code consists of black modules (square dots) arranged in a square grid on a white background, which can be read by an imaging device (such as a camera) and processed using Reed–Solomon error correction until the image can be appropriately interpreted. The required data are then extracted from patterns present in both horizontal and vertical components of the image.

Figure: An example of a QR code can be used as marker for a MAR system and contents.

13 Bibliography

[1] Ferenc A Jolesz, Image-guided procedures and the operating room of the future, Harvard Medical School, Boston, MA, available online (Nov. 2015) at http://www.spl.harvard.edu/archive/spl-pre2007/pages/papers/horizon/horizon.html

[2] Azuma, Ronald T. A Survey of Augmented Reality. Presence: Teleoperators and Virtual Environments 6, 4 (August 1997), pp. 355 - 385. Earlier version appeared in Course Notes #9: Developing Advanced Virtual Reality Applications, ACM SIGGRAPH 1995 (Los Angeles, 6-11 August 1995), 20-1 to 20-38.

[3] Milgram, Paul; H. Takemura, A. Utsumi, F. Kishino (1994). "Augmented Reality: A class of displays on the reality-virtuality continuum". Proceedings of Telemanipulator and Telepresence Technologies. pp. 2351–34.

[4] Mark Billinghurst, Hirokazu Kato, Ivan Poupyrev, The MagicBook: A transitional AR interface, Computers & Graphics, Volume 25, Issue 5, October 2001, Pages 745-753.

[5] Adrian David Cheok, Kok Hwee Goh, Wei Liu, Farzam Farbiz, Siew Wan Fong, Sze Lee Teo, Yu Li, Xubo Yang, Human Pacman: a mobile, wide-area entertainment system based on physical, social, and ubiquitous computing, Personal and Ubiquitous Computing, (2004) 8: 71–81, 2004.

[6] Bruce Thomas, Ben Close, John Donoghue, John Squires, Phillip De Bondi, Michael Morris, Wayne Piekarski, ARQuake: An outdoor/indoor augmented reality first person application, Proc. of ISWC, 2000.

[7] Seokhee Jeon and Seungmoon Choi, Real Stiffness Augmentation for Haptic Augmented Reality, Presence: Teleoperators and Virtual Environments, vol. 20, no. 4, pp. 337-370, 2011.

[8] Robert Lindeman, Haruo Noma, Paul de Barros, An empirical study of hear-through augmented reality: using bone conduction to deliver spatialized audio, , Proc. of IEEE VR, 2008

[9] Gun Lee, Andreas Dunser, Senugwon Kim and Mark Billinghurst, CityViewAR: a mobile outdoor AR application for city visualization, Proc. of ISMAR, 2012.

[10] Deepak Bandyopadhyay, Ramesh Raskar, Henry Fuchs, Dynamic Shader Lamps: Painting on Movable Objects, Proc. of ISAR, 2001.

[11] George Klein and David Murray, Parallel tracking and mapping for small AR workspaces, Proc. of ISMAR, 2007.

[12] A. Davison, I. Reid, N. D. Molton, and O. Stasse, MonoSLAM: Realtime, single camera SLAM. IEEE Trans. Pattern Analysis and Machine Intelligence, 2007.

[13] Richard Newcombe, Shahram Izadi, Otmar Hillinges, David Molyneaux, and David Kim, KinectFusion: Real time dense surface mapping and tracking, Proc. of ISMAR 2011.

[14] T. Lavric, V. Scurtu, M. Preda: Create and play augmented experiences, 104th MPEG meeting, Inchon, April 2014.

[15] MPEG ARAF content: augmented magazine and printed content available online (Nov. 2015) at https://www.youtube.com/watch?v=CNcgxEOt_rM

[16] Keyhole Markup Language / Open Geographica Consortium, www.opengeospatial.org/standards/kml

[17] Mobilizy, ARML (Augmented Reality Markup Language) 1.0 Specification for Wikitude, available online (Nov. 2015) http://www.openarml.org/wikitude4.html

[18] Hill, MacIntyre, Gandy, Davidson and Rouzati, Khamra-KML/HTML Augmented Reality Mobile Architecture, https://research.cc.gatech.edu/polaris

[19] Extensible 3D Graphics, Web3D Consortium, available online (Nov. 2015) at http://www.web3d.org/x3d/what-x3d

[20] Kato, H., Billinghurst, M. "Marker tracking and hmd calibration for a video-based augmented reality conferencing system.", *In Proceedings of the 2nd IEEE and ACM International Workshop on Augmented Reality (IWAR 99)*, October 1999.

[21] Martz, P.: OpenSceneGraph Quick Start Guide. Louisville, USA, 2007

[22] OpenCV Developer Site available online (Nov. 2015) at http://code.opencv.org

[23] IEEE 1471 / ISO 42010:2007