INTERNATIONAL STANDARD

ISO/IEC 18039

First edition
2019-02

Information technology — Computer graphics, image processing and environmental data representation — Mixed and augmented reality (MAR) reference model

Technologies de l'information — Infographie, traitement de l'image et représentation des données environnementales — Modèle de référence en réalité mixte et augmentée

Reference number
ISO/IEC 18039:2019(E)

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Fax: +41 22 749 09 47
Email: copyright@iso.org
Website: www.iso.org

Published in Switzerland

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.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular, the different approval criteria needed for the different types of document should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives).

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. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents) or the IEC list of patent declarations received (see http://patents.iec.ch).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT) see www.iso.org/iso/foreword.html.

This document was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 24, *Computer graphics, image processing and environmental data representation*.

Any feedback or questions on this document should be directed to the user's national standards body. A complete listing of these bodies can be found at www.iso.org/members.html.

Introduction

This document contains annexes:

— Annex A gives examples of existing MAR solutions and technologies and how they fit into the MAR reference model.

— Annex B gives examples of representative MAR systems and how their architecture maps to the MAR reference model.

Information technology — Computer graphics, image processing and environmental data representation — Mixed and augmented reality (MAR) reference model

1 Scope

This document 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 architectural reference model establishes the set of required sub-modules and their minimum functions, the associated information content and the information models to 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 reference model is designed to apply to MAR systems independent of specific algorithms, implementation methods, computational platforms, display systems and sensors or devices used.

This document does not specify how a particular MAR application, component, system, service or specification is designed, developed or implemented. It 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 document contains a list of representative system classes and use cases with respect to the reference model.

2 Normative references

There are no normative references in this document.

3 Terms, definitions and abbreviated terms

For the purposes of this document, the following terms and definitions apply.

ISO and IEC maintain terminological databases for use in standardization at the following addresses:

— ISO Online browsing platform: available at https://www.iso.org/obp

— IEC Electropedia: available at http://www.electropedia.org/

3.1 Terms and definitions

3.1.1
augmentation
virtual object (3.1.24) data (computer-generated, synthetic) added on to or associated with target *physical object* (3.1.15) data (live video, real world image) in an *MAR scene* (3.1.9)

Note 1 to entry: This equally applies to physical object data added on to or associated with target virtual object data.

3.1.2
augmented reality
type of *mixed reality system* (3.1.13) in which *virtual world* (3.1.25) data are embedded and/or registered with the representation of *physical world* (3.1.16) data

3.1.3
augmented virtuality system
type of *mixed reality system* (3.1.13) in which *physical world* (3.1.16) data are embedded and/or registered with the representation of *virtual world* (3.1.25) data

3.1.4
display
device by which rendering results are presented to a user using various modalities such as visual, auditory, haptics, olfactory, thermal, motion

Note 1 to entry: In addition, any actuator can be considered display if it is controlled by MAR system.

3.1.5
feature
primitive geometric element (e.g. points, lines, polygons, colour, texture, shapes) and/or attribute of a given (usually physical) object used in its detection, recognition and tracking

3.1.6
MAR event
trigger resulting from the detection of a condition relevant to MAR content and *augmentation* (3.1.1)

EXAMPLE Detection of a marker.

3.1.7
MAR execution engine
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

3.1.8
MAR experience
human visualization and interaction of an *MAR scene* (3.1.9)

3.1.9
MAR scene
observable spatio-temporal organization of physical and *virtual objects* (3.1.24) which is the result of an *MAR scene representation* (3.1.10) being interpreted by an *MAR execution engine* (3.1.7) and which has at least one *physical* (3.1.15) and one virtual object

3.1.10
MAR scene representation
data structure that arranges the logical and spatial representation of a graphical scene, including the *physical* (3.1.15) and *virtual objects* (3.1.24) that are used by the *MAR execution engine* (3.1.7) to produce an *MAR scene* (3.1.9)

3.1.11
marker
metadata embedded in or associated with a *physical object* (3.1.15) that specifies the location of a super-imposed object

3.1.12
MAR continuum
spectrum spanning physical and virtual realities according to a proportional composition of physical and virtual data representations

Note 1 to entry: Originally proposed by Milgram et al.[1].

3.1.13
mixed reality system
mixed and augmented reality system
system that uses a mixture of representations of *physical world* (3.1.16) data and *virtual world* (3.1.25) data as its presentation medium

3.1.14
natural feature
feature (3.1.5) that is not artificially inserted for the purpose of easy detection/recognition/tracking

3.1.15
physical object
real object
object that exists in the real world

3.1.16
physical world
physical reality spatial organization of multiple *physical objects* (3.1.15)

3.1.17
point of interest
single or collection of target locations

Note 1 to entry: Aside from location data, a point of interest is usually associated with metadata such as identifier and other location specific information.

3.1.18
recognizer
MAR component (hardware and software) that processes *sensor* (3.1.19) output and generates *MAR events* (3.1.6) based on conditions indicated by the content creator

3.1.19
sensor
device that returns detected values related to detected or measured condition or property

Note 1 to entry: Sensor may be an aggregate of sensors.

3.1.20
spatial registration
establishment of the spatial relationship or mapping between two models, typically between *virtual object* (3.1.24) and target *physical object* (3.1.15)

3.1.21
target image
target object (3.1.22) represented by a 2D image

3.1.22
target object
physical (3.1.15) or *virtual object* (3.1.24) that is designated, designed or chosen to allow detection, recognition and tracking, and finally *augmentation* (3.1.1)

3.1.23
tracker
MAR component (hardware and software) that analyses signals from *sensors* (3.1.19) and provides some characteristics of tracked entity (e.g. position, orientation, amplitude, profile)

3.1.24
virtual object
computer-generated entity that is designated for *augmentation* (3.1.1) in association with a *physical object* (3.1.15) data representation

Note 1 to entry: In the context of MAR, it usually has perceptual (e.g. visual, aural) characteristics and, optionally, dynamic reactive behaviour.

3.1.25
virtual world
virtual environment
spatial organization of multiple *virtual objects* (3.1.24), potentially including global behaviour

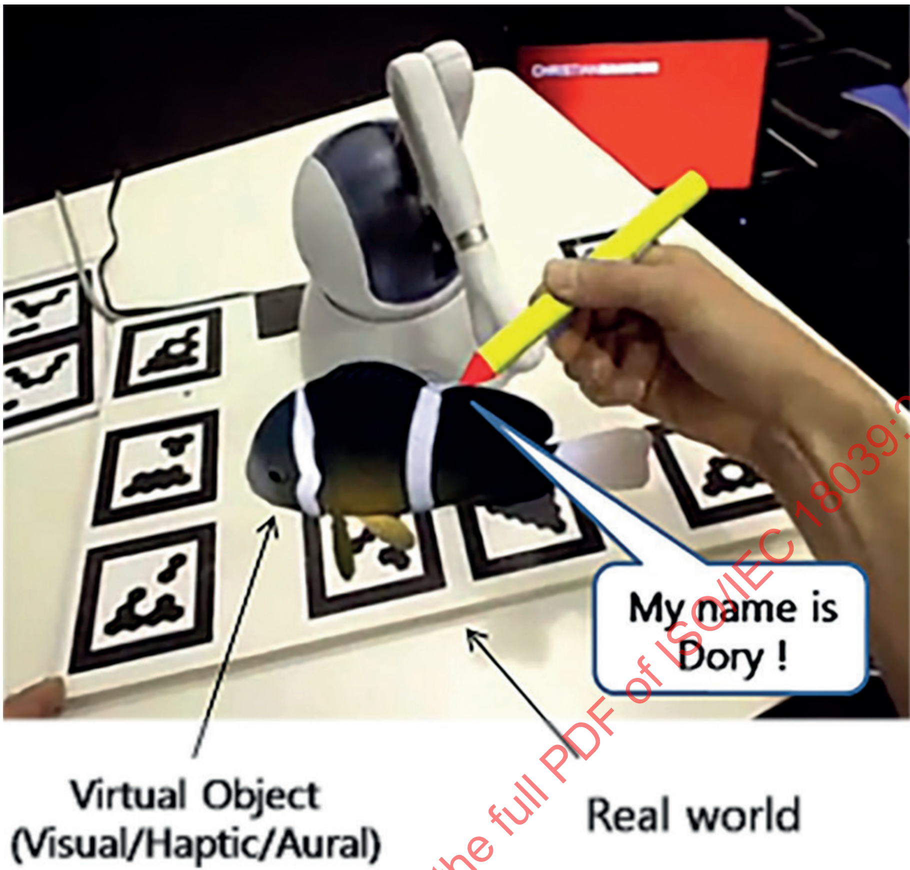
3.2 Abbreviated terms

Abbreviated term	Definition
API	Application program interface
AR	Augmented reality
AVH	Audio, visual, haptic
GNSS	Global navigation 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

4 Mixed and augmented reality (MAR) domain and concepts

4.1 General

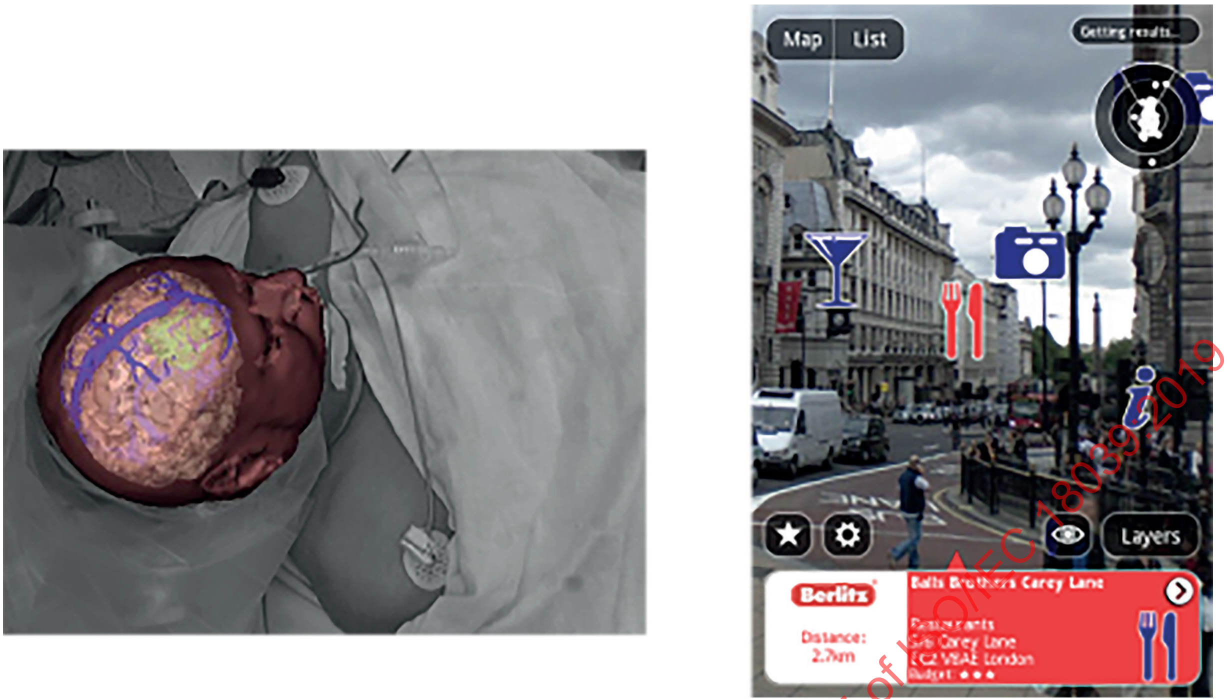
MAR refers to a spatially coordinated combination of media/information components that represent on the one hand the real world and its objects, and on the other those that are virtual, synthetic and computer generated. The virtual component can be represented and presented in many modalities (e.g. visual,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



SOURCE Magic Vision Lab, University of South Australia, reproduced with the permission of the authors

Figure 1 — The concept of MAR as a combination of representations of physical objects and computer mediated virtual objects 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 are computed and used to control the position and orientation of a virtual object. Temporal registration can also occur when the presence of a real object is detected and a virtual object is to be displayed. Registration can 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.



NOTE Virtual brain imagery tightly registered on a real human body image is shown on the left-hand side[2] and tourist information overlaid less tightly over a street scene[3] is shown on the right-hand side.

Figure 2 — The notion of registration precision at different degrees

An MAR system refers to real-time processing[4]. For example, while a live close-captioned broadcast would qualify as an MAR service, an offline production of a subtitled movie would not.

4.2 MAR continuum

Since an MAR system or its contents combines real and virtual components, an 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 VR ("All virtual, no physical") on the other end (as illustrated in Figure 3). A single instance of a system at any point on this continuum[1] that uses a mixture of both real and virtual presentation media is called an MR system. In addition, for historical reasons, MR is often synonymously or interchangeably used with AR, which is actually a particular type of MR (see Clause 7). In this document, 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 VR) are not in the scope of this document.

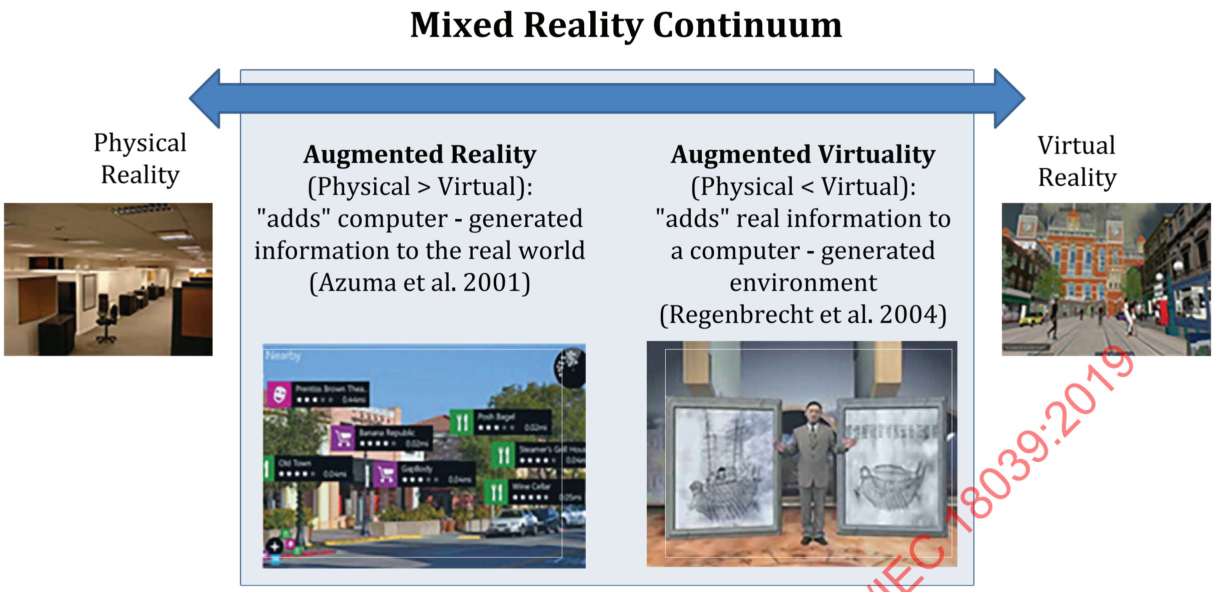


Figure 3 — The MAR (or reality-virtuality) continuum[1]: definition of different types of MR according to the relative portion between the real world representation and the virtual

Two notable types of MAR or points in the continuum are the AR and augmented virtuality. An AR 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 MR system in which the medium representing physical objects (e.g. video) is embedded into the computer-generated information (as illustrated in Figure 3).

5 MAR reference model usage example

5.1 Designing an MAR application or service

The MAR reference model is a reference guide in designing an MAR service and developing an MAR system, application or content. With respect to the given application (or service) requirements, the designer may refer to and select the necessary components from those specified in the MAR reference architecture (see Clause 6). 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 component classification scheme described in Clause 7 can help the designer to specify a more precise scope and capabilities, while the specific system classes defined in Clause 8 can facilitate the process of model, system or service refinement.

5.2 Deriving an 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.

5.3 Extending existing or creating 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 can 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, POI constructs).

6 MAR reference system architecture

6.1 Overview

An 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 an MAR system is given in Figure 4. The central pink area indicates the scope of the MAR-RM. The blue round boxes are the main computational modules and the dotted box represents the required information constructs. Arrows indicate data flow. Control signals are not explicitly shown, to avoid particular technical or implementation details in this document. Also, this reference system architecture should not to be taken as something rigid and unchangeable, but as a depiction of a typical case at the macro scale. Flexibility exists in actual application.

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); and

— managing the communication with additional services.

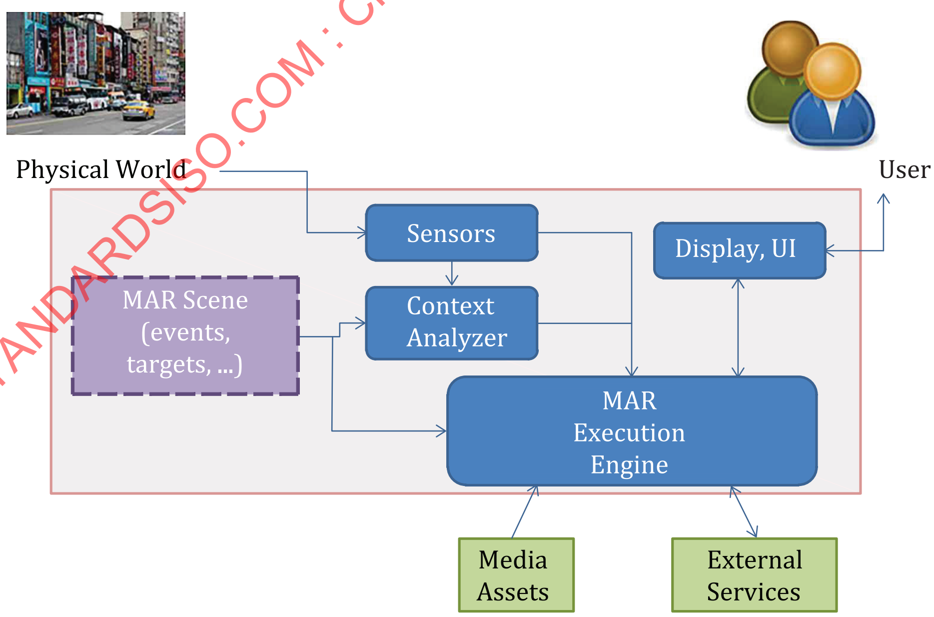


Figure 4 — Major components and their interconnection in an MAR system at a high macro level

6.2 Viewpoints

In order to detail the global architecture presented in Figure 4, the reference model considers three analysis angles, called viewpoints: enterprise, computation and information. This viewpoint-wise exposition permits readers, who can be interested in or focused on particular aspects or viewpoints, to better understand the MAR architecture. The definition of each viewpoint is provided in Table 1. In this subclause, the terms "view" and "viewpoint" are used in the context of information modelling and establishing a reference architecture. They should not be confused with the same term that refers to the location of the virtual camera in computer graphics or virtual environments.

The notion of view is separate from that of the viewpoint: a viewpoint identifies the set of concerns, representations and modelling techniques used to describe the architecture to address those concerns, and a view is the result of applying a viewpoint to a particular system[5].

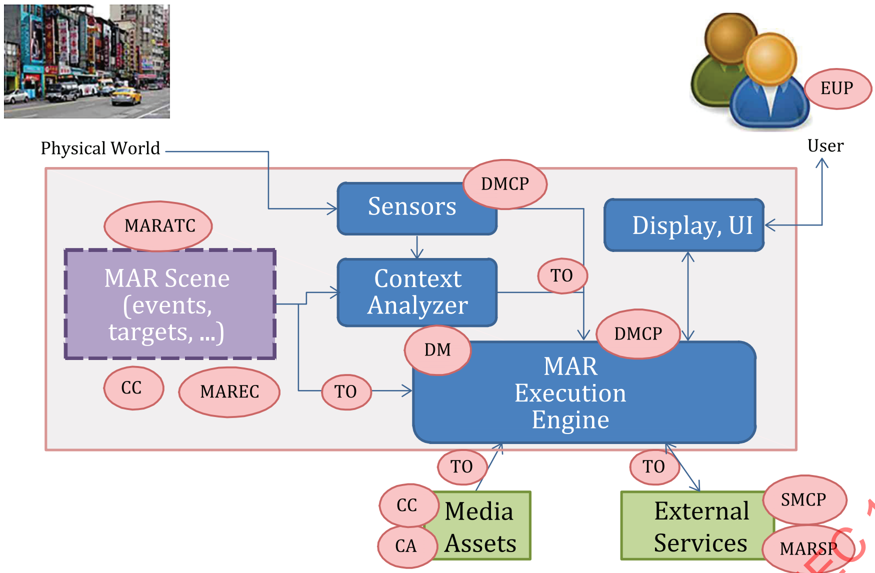
Table 1 — Definitions of the MAR viewpoints

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 each actor. Desirable characteristics for the actors at both ends of the value chain (creators and users).
Computational	Identifies the functionalities of system components and their interfaces. 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. Describes how the information is 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. Content information such as virtual objects, application behaviour and user interaction(s) management. Service information such as remote processing of the context data.

6.3 Enterprise viewpoint

6.3.1 General

The Enterprise viewpoint (see Figure 5) describes the actors involved in an MAR system, their objectives, roles and requirements. The actors can be classified according to their role. Several types of actors from the list in 6.3.2 can commercially exploit an MAR system.



Key

MARATC	MAR authoring tools creator
CA	content aggregator
TO	telecommunication operator
MAR EC	MAR experience creator
CC	content creator
DM	device manufacturer
DMCP	device middleware/component provider
EUP	MAR consumer/end user profile
SMCP	service middleware/component provider
MARSP	MAR service provider

Figure 5 — The “Enterprise viewpoint” of an MAR system and the main actors

6.3.2 Classes of actors

6.3.2.1 Class 1: providers of authoring/publishing capabilities

— *MAR authoring tools creator*: a software platform provider of the tool used to create (author) an MAR-enabled application or service; the output of the MAR authoring tool is called MAR scene representation.

— *MAR experience creator*: a person that designs and implements an MAR-enabled application or service.

— *Content creator*: a designer (person or organization) that creates multimedia content (scenes, objects); even the end user of the MAR system can be the designer of the content.

6.3.2.2 Class 2: providers of MAR execution engine components

— *Device manufacturer*: an organization that produces devices in charge of augmentation and used as end-user terminals.

— *Device middleware/component provider*: an organization that creates and provides hardware, software and/or middleware for the augmentation device, which can be one of the following modules:
  — multimedia player or browser engine provider (rendering, interaction engine, execution, etc.);
  — context knowledge provider (satellites, etc.); or
  — sensor manufacturers (inertial, geomagnetic, camera, microphone, etc.).

6.3.2.3 Class 3: service providers

— *MAR service provider*: an organization that discovers and delivers services.

— *Content aggregator*: an organization aggregating, storing, processing and serving content.

— *Telecommunication operator*: an organization that manages telecommunication among other actors.

— *Service middleware/component provider*: an organization that creates and provides hardware, software and/or middleware for processing servers, including services such as:
  — location providers (network-based location services, image databases, RFID-based location, etc.);
  — semantic provider (indexed image or text databases, etc.).

6.3.2.4 Class 4: MAR end user

The MAR consumer and end-user profile is a person who experiences the real world synchronized with digital assets and uses an MAR scene representation, an 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.

6.3.3 Business model of MAR systems

The different business models of the actors in the MAR system are as follows.

— The MAR authoring tools creator may provide the authoring software or content environment to an MAR experience creator. Such tools range 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, etc.) that may be used in the MAR experience.

— An MAR experience creator creates an MAR experience in the form of an MAR rich media representation. They can associate media assets with features in the real world, transforming them into MAR-enabled digital assets. The MAR experience creator also defines the global and/or 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 is where the MAR experience creator specifies a set of minimal requirements to 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).

— An MAR service provider supports the delivery of MAR experiences. This can be via catalogues that assist in discovering an MAR experience.

6.3.4 Criteria for successful MAR systems

The Enterprise-related requirements for the successful implementation of an 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.

6.4 Computational viewpoint

6.4.1 General

The Computational viewpoint (see Figure 6) describes the overall interworking of an MAR system. It identifies major processing components (hardware and software), defines their roles and describes how they interconnect.

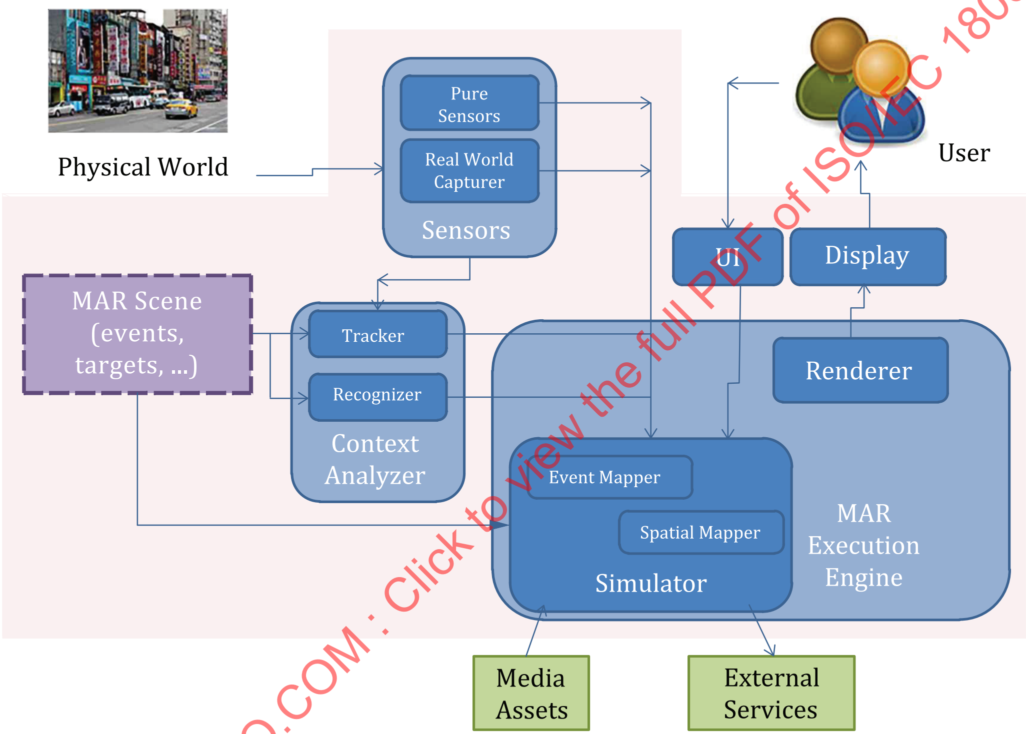


Figure 6 — Computational viewpoint: illustrating and identifying the major computational blocks in the MAR system and service

6.4.2 Sensors: pure sensor and real world capturer

A sensor is a hardware (and optionally software) component able to measure specific physical properties. 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 an 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 for use 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. The captured real world data can be in any modality, such as visual, aural, haptic. The real world data can be sensed and captured ahead of time or remotely (not of the local space) in various formats, including 3D models, point clouds and image-based (light field) models.

A sensor can measure different physical properties, and interpret and convert these observations into digital signals. The captured data can be used (1) to compute only the context in the tracker and 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; and

— output: sensor observations with or without additional metadata (position, time, etc.).

The sensors can be categorized as set out in Table 2.

Table 2 — Sensor categories

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

6.4.3 Context analyser: recognizer and tracker

The context analyser is composed of the recognizer and the tracker.

The recognizer is a hardware or software component that analyses signals from the real world and produces MAR events and data by comparing these 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).

Recognition can only be based on prior captured target signals. Both the recognizer and the tracker can be configured with a set of target signals provided by or stored in an outside resource (e.g. third-party data base server) in a consistent manner with the scene definition, or by the MAR scene description (see 6.5.7) itself.

The recognizer and the 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 target object specification data (reference target to be recognized); and

— output: at least one event acknowledging the recognition.

The input and output of the tracker are:

— input: raw or processed signals representing the physical world and target object specification data (reference target to be recognized); and

— output: instantaneous values of the characteristics (pose, orientation, volume, etc.) of the recognized target signals.

The recognizers and trackers can be categorized as set out in Tables 3 and 4.

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.)	—	—	—	—

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, etc)	Aural (intensity, pitch, etc.)	Haptic (force, direction, etc.)	Others	—
3. Place of execution	Local system	Remote system (server, cloud, etc.)	—	—	—

6.4.4 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 need 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; and

— 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).

6.4.5 Event mapper

The event mapper creates an association between an 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) can be identified as "MAR_location_event_1" while the content creator can 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; and

— output: translated event identifier for the given MAR scene.

6.4.6 MAR execution engine

The MAR execution engine constitutes the core of any MAR system. Its main purpose is to interpret the sensed data to further recognize and track the target data to be augmented, import the real world or object data, computationally simulate the dynamic behaviour of the augmented world, and compose the real and virtual data together for proper rendering in the required modalities (e.g. visually, aurally, haptically). The execution engine can require additional and/or external media assets or computational services for supporting these core functionalities. The MAR execution engine can be part of a software application able to load a full scene description [including assets, scene behaviour, user interaction(s)] for its simulation and presentation or part of a standalone application with preprogrammed 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) interpreting data provided by various mappers, user interaction(s), sensors, local and/or remote services;

(3) executing and simulating scene behaviours; and

(4) composing and synchronizing various types of multimodal media representations (aural, visual, haptics, etc.).

The input and output of the execution engine are:

— input: MAR scene description, user input(s), (mapped) MAR events and external service events; and

— output: an updated version of the scene description.

The execution engine can be categorized according to the dimensions given in Table 5.

Table 5 — Execution engine categories

Dimension	Types			
1. Space and 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	—	—

6.4.7 Renderer

The renderer refers to the software and optionally hardware components for producing, from the MAR scene description (see 6.5.7), updated after a tick (the smallest time unit) 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 or can be a combination of multiple modalities. 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; and

— output: synchronized rendering output (e.g. visual frame, stereo sound signal, motor commands).

The renderer can be categorized as set out in Table 6.

Table 6 — Renderer categories

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

6.4.8 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, AR glasses, head-mounted displays (HMD), projectors, scent diffusers, haptic devices and sound speakers. A special type of display is an actuator that does not directly stimulate the end users' senses, but can 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 can 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 (or user’s actions).

The input and output of the display are:

— input: render signals; and

— 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 having their own attributes as set out in Table 7 to Table 10.

Table 7 — Visual display categories

Dimension	Types			
1. **Presentation**	Optical see through	Video see through	Projection	—
2. **Mobility**	Fixed	Mobile	Controlled	—
3. **Number of channels**	2D (mono)	3D stereoscopic	3D holographic	—

Table 8 — Aural display categories

Dimension	Types			
1. **Number 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.)

6.4.9 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 frequently-used MAR functionalities and data models in the following ways (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;

— managing the sensors;

— detecting and recognizing markers and target objects;

— managing markers and 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 MAR events between those that are user-defined and those that are system-defined;

— manipulating the MAR scene for its dynamic simulation;

— handling user inputs; and

— making composite renderings and synchronizing for specific displays, possibly in different modalities.

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

6.5 Information viewpoint

6.5.1 General

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. For some components, standards are already available providing full data models.

6.5.2 Sensors

The sensor 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. Sensors may be used to capture the real world off-line or remotely, in which case the output of the sensors forms a real world object or scene representation (rather than a stream of instances).

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 and frequency range), extrinsic parameters (e.g. position and orientation) and 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, Table 11 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), 41 000 (sampling rate), etc.

The input and output of the sensors are:

— input: the real world (no information model is required); and

— output: sensor observations (optionally post-processed in order to extract additional metadata such as position or time, or to be processed and used as a representation for the real world/object), which depend on the type of sensor used (e.g. binary image, colour image, depth map, sound stream, force). If used as an off-line real world capture, the output is a complete real world object or scene description.

6.5.3 Recognizer

There are two types of information used by the recognizer: the output from the sensor component and the target physical object representation. By analysing this information, the recognizer outputs an MAR event.

— Input: the input data model of the recognizer is the output of the sensors. In addition, another input to the recognizer, the target physical object data, should contain an identifier indicating the event when the presence of the target object is recognized. The target physical object data may also include raw template files (such as image files, 3D model files, sound files) and/or a set of feature profiles used for the recognition and matching process.

— The types of features depend on the algorithms used by the recognizer; for instance, it can be a set of visual feature descriptors, 3D geometric features, etc.

— Output: 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, Table 12 and Table 13 illustrate possible event specifications.

Table 12 — Target physical object attribute

Sensor attribute	Values
Recognition event identifier	"Image_1", "Face_Smith", "Location_1", "Teapot3d", etc.
Raw template file and 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.

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.
Timestamp	12:32:23, 02:23:01

6.5.4 Tracker

There are two types of information used by the tracker: the sensors' output and the target physical object representation. By analysing this information, the tracker outputs an MAR event.

— Input: the input data model of the recognizer is the output of the sensors. In addition, another input to the recognizer, the target physical object data, should contain the same elements as the recognizer.

— Output: a continuous stream of instantaneous values of the characteristics (pose, orientation, volume, etc.) of the recognized target signals (see Table 15).

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, 4×4 transformation matrix, current volume level, current force level, etc.
Optional: timestamp	12:32:23, 02:23:01

6.5.5 Spatial mapper

In order to map the physical sensor space into the MAR scene, explicit mapping information should be supplied by the content or system developer. The spatial mapping information can be modelled as a table (see Table 15), with each entry characterizing the translation process from one aspect of the spatial property (e.g. lateral unit, axis direction, scale) of the sensor to the given MAR scene. There is a unique table defined for a set of sensors and an 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)

6.5.6 Event mapper

In order to map MAR events as defined by the content developer or specified within the MAR scene representation, as well as events identified and recognized by the recognizer, a correspondence table is needed. Table 16 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 an MAR scene.

Table 16 — Event mapping table example

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

6.5.7 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:

— scene description, including for real/virtual objects and spatial organization;

— scene behaviour;

— specification of how real world/object is captured or represented;

— 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;

— synchronization requirements on multimodal input or output data;

— augmentation output style and its dynamics; and

— access to remote services like maps, image databases, processing servers, etc.

The execution engine output is an "updated" scene graph data structure.

6.5.8 Renderer

The input of the AVH (aural, visual and haptic) renderer is an updated scene graph.

The output is a visual, aural and/or 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).

The MAR system can specify various capabilities of the AVH renderer (see Table 17), so the scene can be adapted and simulation performance can be optimized. For instance, a stereoscopic HMD and a mobile device can require different rendering performances. Multimodal output rendering can necessitate careful millisecond-level temporal synchronization.

Table 17 — Main types and properties of renderer

Renderer type	Capabilities dimensions
Visual	Screen size, resolution, FOV (field of view), number of channels, signal type
Aural	Sampling rate, number of channels, maximum volume
Haptic	Resolution, operating spatial range, degrees of freedom, force range

6.5.9 Display and user interface

The input to the display is a stream of synchronized visual, aural and/or haptic data, and the output, presentation display to the end user.

The input to the UI module is the user/user's action, and the output is a set of signals, as detected by the associated sensors, to be sent to the execution engine in order to update the scene.

7 MAR component classification framework

Table 18 summarizes, in the form of a classification framework, major attributes and possible values of the system components of an MAR system as seen from various viewpoints. It serves to translate abstract MAR-RM concepts into real world MAR implementations.

Table 18 — MAR component classification framework

Component	Dimension	Types				
Pure sensors	Modality	Visual	Auditory	Electromagnetic waves (e.g. GNSS)	Haptic/ tactile	Other physical properties
	Source type	Live	Pre-captured	—	—	—
Real world capturer	Modality	Visual	Auditory	Haptics properties	Other	—
	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)	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	Other
	Form of the output event	Spatial (2D, 3D, 6D, etc.)	Aural (intensity, pitch, etc.)	Haptic (force, direction, etc.)	—	—
	Execution place	Local	Remote	—	—	—
Space mapper	Space type	Spatial	Audio	Haptics	Others	—
Event mapper	Modality	Visual	Temporal	Aural	Location	Others
Execution engine	Space and 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	—
	Execution place	Local	Remote	Hybrid	—	—
Visual display	Presentation	Optical see through	Video see through	Projection	—	—
	Mobility	Fixed	Mobile	Controlled	—	—
	Number of channels	2D (mono)	3D stereoscopic	3D holographic	—	—

Table 18 *(continued)*

Component	Dimension	Types				
Aural display	Number of channels	Mono	Spatial	—	—	—
	Acoustic space coverage	Head phones	Speaker	—	—	—
Haptics Display	Type	Vibration	Pressure	Temperature	—	—
UI	Interaction type	Touch	Click	Drag	Gesture	Other

8 MAR system classes

8.1 General

This clause uses the component classification framework provided in Clause 7 to describe several higher classes of MAR application or services. Each class focuses on parts of the system architecture and provides illustrations of how to use the reference model. The possible system class examples are illustrated in Table 19. An instance of an MAR system can also be a combination of several classes. Detailed descriptions of some of the examples in the table are given in 8.2 to 8.6. While the various system classes described in this clause are illustrative, they are not exhaustive.

Table 19 — MAR system class examples

MAR Class V	Systems augmenting the 2D visual data captured by one or more real cameras
MAR Class V-R	Systems augmenting the 2D visual data continuously captured by one or more real cameras and reconstructing the 3D environment
MAR Class 3DV	Systems augmenting the 3D visual data captured by one or more cameras and/or depth cameras
MAR Class G	Systems using global position system to register synthetic objects in the real world
MAR Class A	Systems augmenting the aural modality
MAR Class 3DA	Systems augmenting the scene by using 3D audio data
MAR Class H	Systems augmenting the haptic modality

8.2 MAR Class V — Visual augmentation systems

8.2.1 Local recognition and tracking

The device detects the presence of target resources (images or their corresponding descriptors) (see Figure 7) in a video stream, optionally computes the transformation matrix (position, orientation and scaling) of the detected resources 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; and

— optionally a 2D region (in the video frame) to be considered in the recognition process and delay constraints.

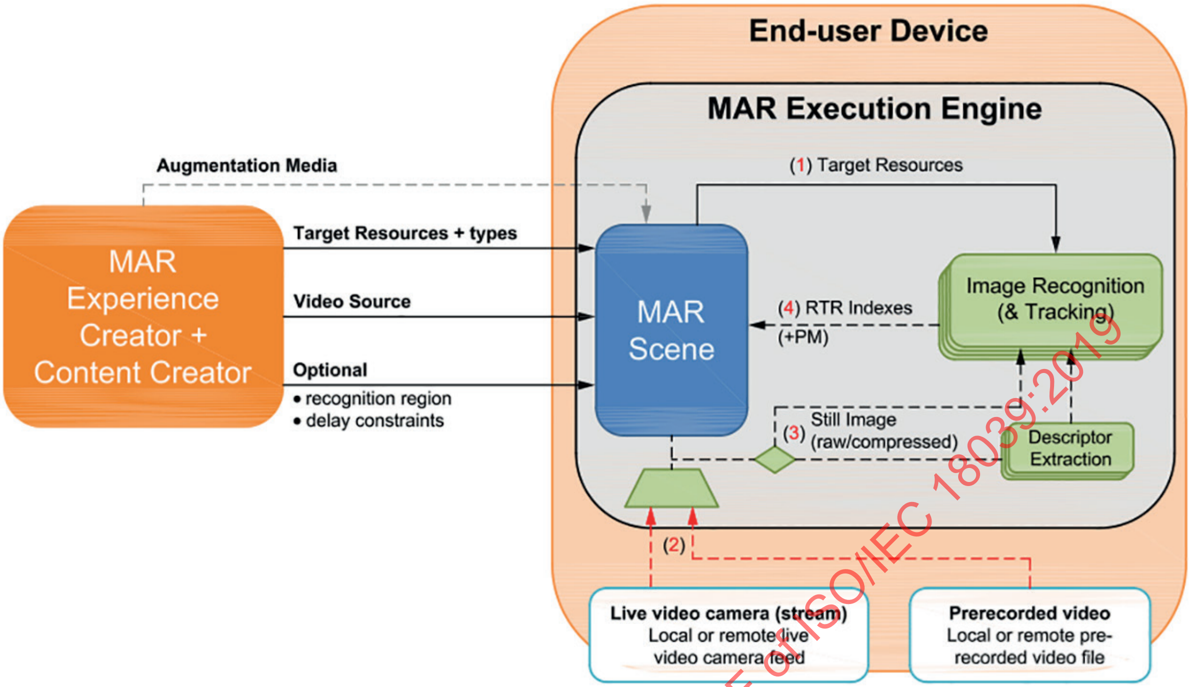


Figure 7 — Local recognition (and tracking for the pose matrix)

According to the MAR component classification scheme in Clause 7, this system class has the following characteristics:

— pure sensors – visual;

— real world capturer – visual, 2D video;

— recognizer – image patch, recognition event, local;

— tracker – image patch, spatial event, local;

— space mapper – spatial;

— event mapper – visual;

— execution engine – local, 2D + t;

— renderer – visual; and

— visual display – 2D mono.

8.2.2 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 frame rate (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 (see Figure 8). 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 are sent to the processing server (raw/compressed image or the corresponding video frame descriptors);

— media used for the augmentation;

— URL to the processing servers; and

— optionally, a 2D region (in the video frame) to be considered in the recognition process and delay constraints.

In addition, a communication protocol has to be implemented between the MAR execution engine and the processing servers.

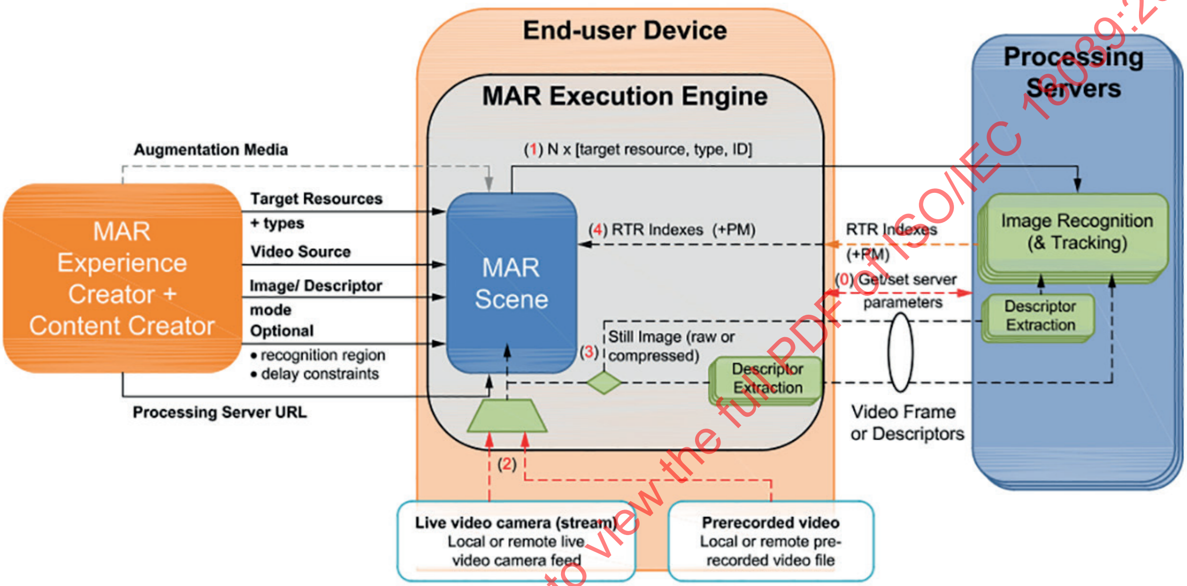


Figure 8 — Local registration, remote recognition and tracking

According to the MAR component classification scheme in Clause 7, this system class has the following characteristics:

— pure sensors – visual;

— real world capturer – visual, 2D video;

— recognizer – image patch, recognition event, remote;

— tracker – image patch, spatial event, remote;

— space mapper – spatial;

— event mapper – visual;

— execution engine – local, 2D + t;

— renderer – visual; and

— visual display – 2D mono.

8.2.3 Remote recognition, local tracking and registration

The device sends video frames in a format that can be specified by the 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 more target resources that are stored in its local database (see Figure 9). The server returns the position and size of one or more target resources detected in the frame, as well as the augmentation content (virtual objects, application behaviour). By using position and size, the device crops the target images from the frame and uses 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 are sent to the processing server (raw/compressed image or the corresponding video frame descriptors); and

— optionally, a 2D region (in the video frame) to be considered in the recognition process and delay constraints.

In addition, a communication protocol has to be implemented between the MAR execution engine and the processing servers.

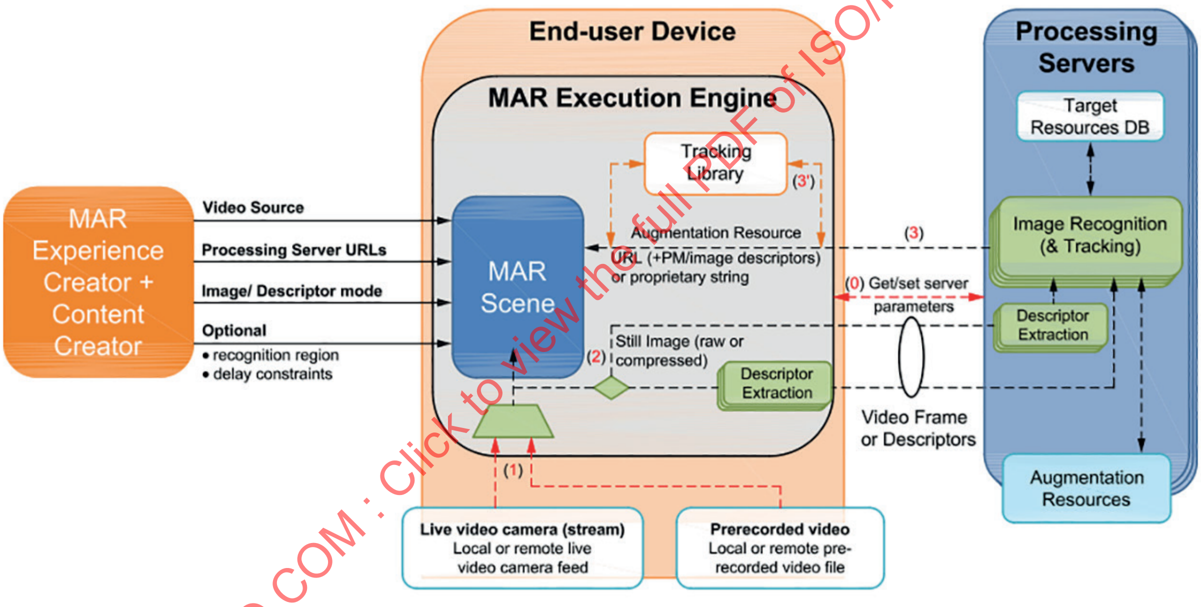


Figure 9 — Remote recognition, local tracking and registration

According to the MAR component classification scheme in Clause 7, this system class has the following characteristics:

— pure sensors – visual;

— real world capturer – visual, 2D video;

— recognizer – image patch, recognition event, remote;

— tracker – image patch , spatial event, local;

— space mapper – spatial;

— event mapper – visual;

— execution engine – local, 2D + t;

— renderer – visual; and

— visual display – 2D mono.

8.2.4 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 more target resources that are stored in its local (or remote) database (see Figure 10). 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; and

— optionally, 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.

In addition, a communication protocol has to be implemented between the MAR execution engine and the processing server.

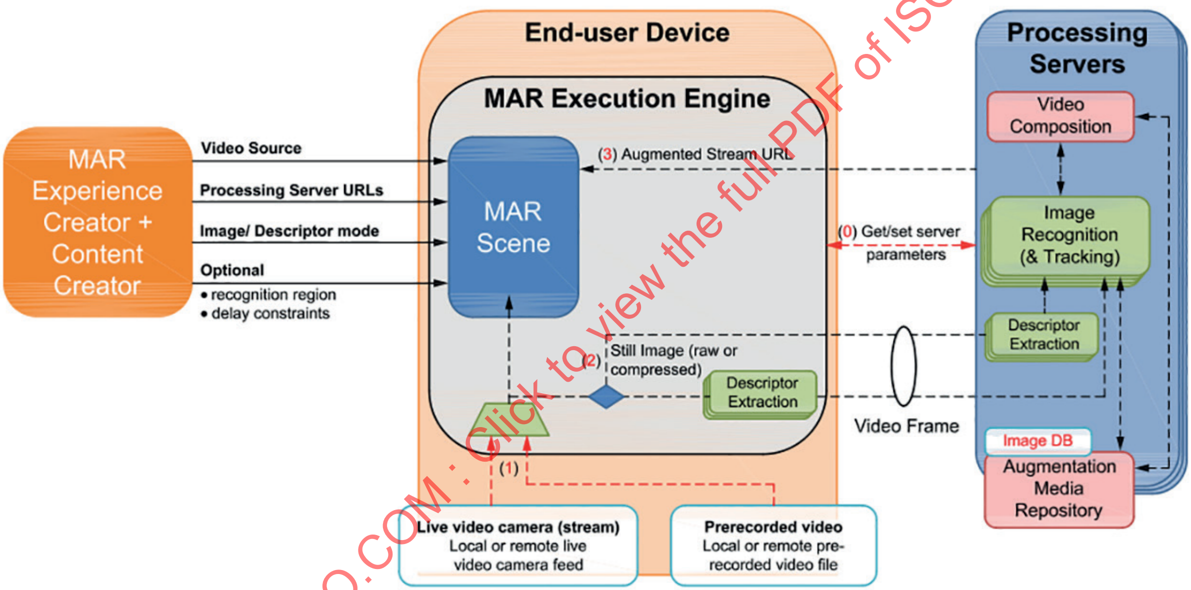


Figure 10 — Remote recognition, recognition and composition

According to the MAR component classification scheme in Clause 7, this system class has the following characteristics:

— pure sensors – visual;

— real world capturer – visual, 2D video;

— recognizer – image patch, recognition event, remote;

— tracker – image patch , spatial event, remote;

— space mapper – spatial;

— event mapper – visual;

— execution engine – remote, 2D + t;

— renderer – visual; and

— visual display – 2D mono.

8.2.5 MAR Class V-R: visual augmentation with 3D environment reconstruction

The device sends a video stream to a processing server (or local client) that is analysing the data and detects one or more target resources that are stored in its local (or remote) database. At the same time, 3D reconstruction algorithms are applied (typically a variant of SLAM: simultaneous localization and papping), reconstructing part of the environment in which the augmentation is carried out. The 3D information obtained is used to help detect, track and make the spatial augmentation possible.

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); and

— optionally, URL to the processing servers;

According to the MAR component classification scheme in Clause 7, this system class has the following characteristics:

— pure sensors – visual;

— real world capturer – visual, 2D video;

— recognizer – image patch, recognition event, remote;

— tracker – image patch , spatial event, remote;

— space mapper – spatial;

— event mapper – visual;

— execution engine – remote, 2D + t;

— renderer – visual; and

— visual display – 2D mono.

8.3 MAR type 3DV: 3D video systems

8.3.1 Real-time, local-depth estimation, condition-based augmentation

The device captures multi-view video and estimates depth. This representation is used to detect conditions imposed by the content designer (see Figure 11). Once the condition is met, the device renders the virtual object by using the scale and orientation specified by the content designer. For example, the end user has an AR experience where one virtual object is displayed on a horizontal plane detected within a ray of 10 m. The content specified in the Information viewpoint is:

— media used for the augmentation;

— the orientation and scale of the virtual object (uniform and/or isotropic scaling representing physical units); and

— the condition (e.g. horizontal plane within a ray of 10 m).

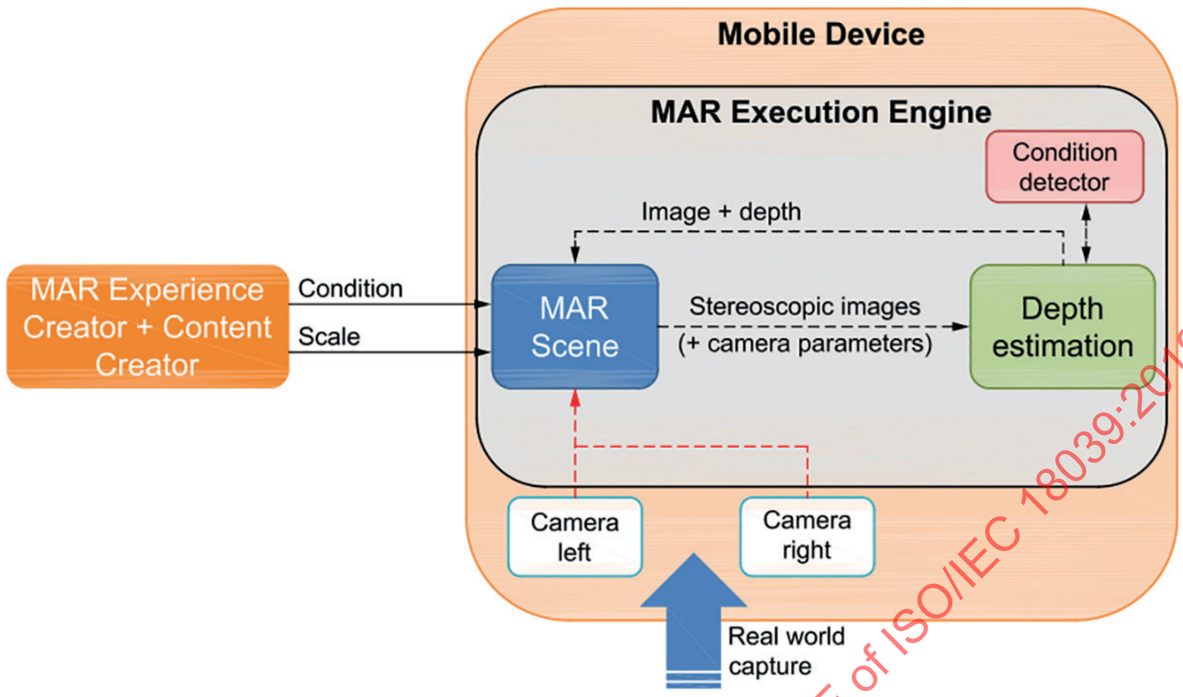


Figure 11 — Real-time, local depth estimation, condition based augmentation

According to the MAR component classification scheme in Clause 7, this system class has the following characteristics:

— pure sensors – visual, other (3D depth);

— real world capturer – visual/video, other (3D depth);

— recognizer – 3D primitives, recognition event, local;

— tracker – 3D primitives, spatial event, local;

— space mapper – spatial;

— event mapper – location, local;

— execution engine – local, 3D + t;

— renderer – visual; and

— visual display – 3D.

8.3.2 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 the author's content by introducing additional 3D virtual objects registered within an approximation of the real world (see Figure 12). 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 using the transformation matrix result. The content specified in the Information viewpoint is:

— virtual objects and their local transformations in the MAR scene experience; and

— the approximation of the 3D model of the real world.

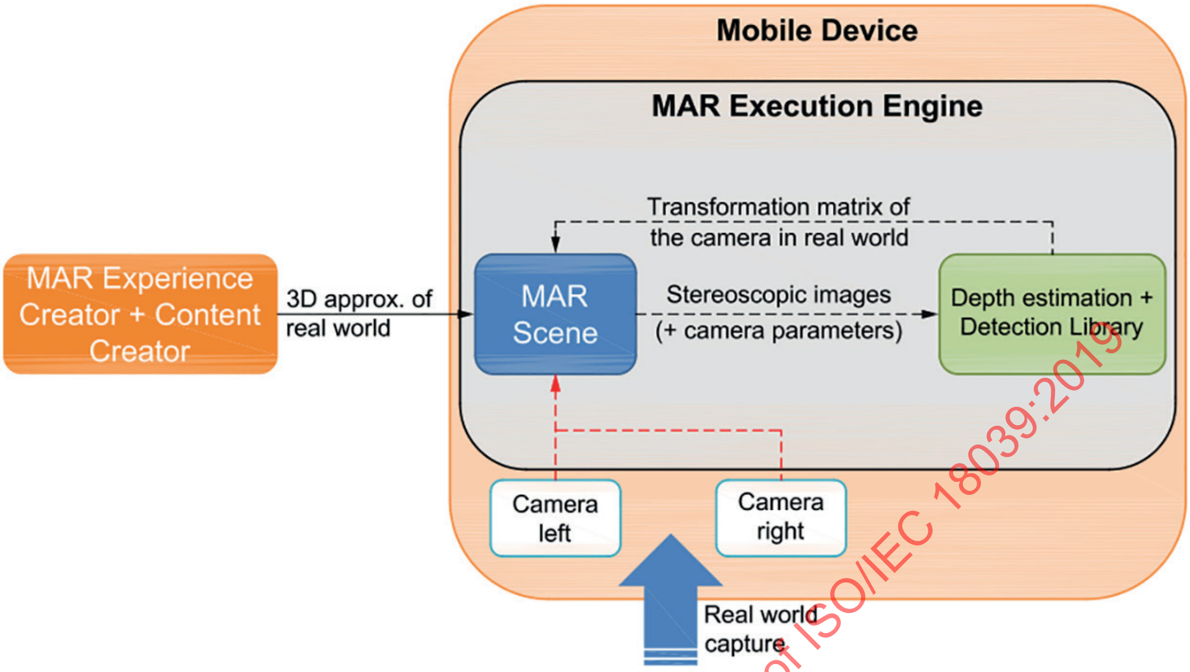


Figure 12 — Real-time, local depth estimation, model-based augmentation

According to the MAR component classification scheme in Clause 7, this system class has the following characteristics:

— pure sensors – visual, other (3D depth);

— real world capturer – visual/video, other (3D depth);

— recognizer – 3D model/primitives, recognition event, local;

— tracker – 3D model/primitives, spatial event, local;

— space mapper – spatial;

— event mapper – location, local;

— execution engine – local, 3D + t;

— renderer – visual; and

— visual display – 3D.

8.3.3 Real-time, remote depth estimation, condition-based augmentation

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

The device captures multi-view video and sends synchronized samples to a processing server that estimates the depth (see Figure 13). This representation is sent to the device and the server uses it to detect conditions imposed by the content designer. The server also sends 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 and/or isotropic scaling representing physical units);

— the condition (e.g. horizontal plane within a ray of 10 m); and

— URL of the processing server.

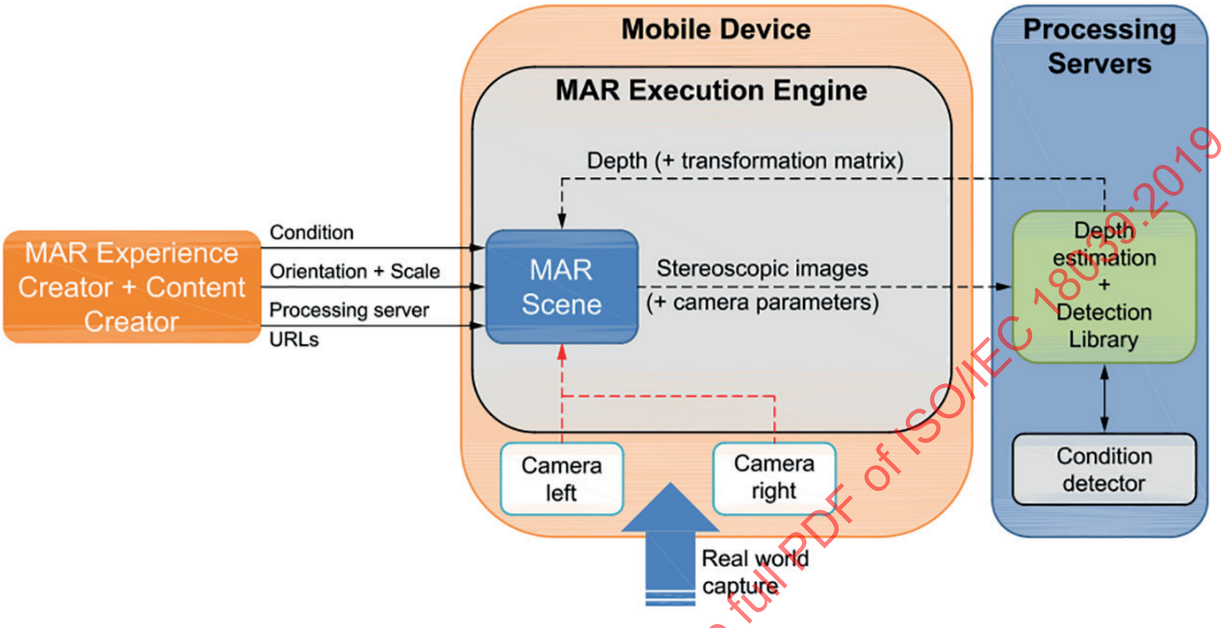


Figure 13 — Real-time, remote-depth estimation, condition-based augmentation

According to the MAR component classification scheme in Clause 7, this system class has the following characteristics:

— pure sensors – visual, other (3D depth);

— real world capturer – visual/video, other (3D depth);

— recognizer – 3D primitives, recognition event, remote;

— tracker – 3D primitives, spatial event, remote;

— space mapper – spatial;

— event mapper – location, local;

— execution engine – local, 3D + t;

— renderer – visual; and

— visual display – 3D.

8.3.4 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 (see Figure 14). 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; and

— URL of the processing server.

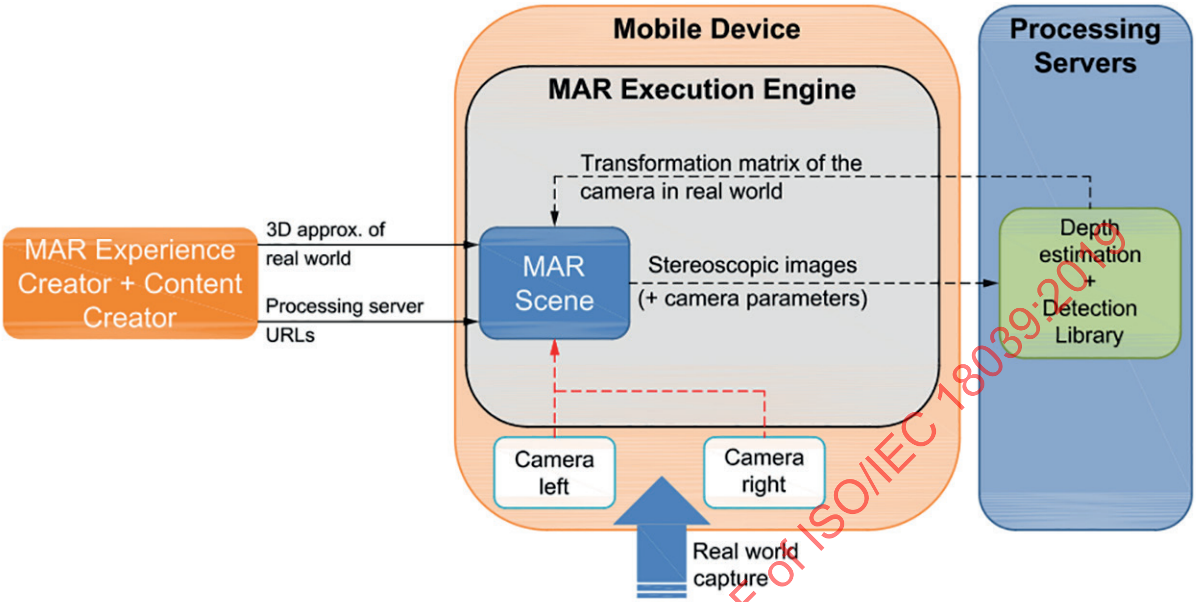


Figure 14 — Real-time, remote-depth estimation, model-based augmentation

According to the MAR component classification scheme in Clause 7, this system class has the following characteristics:

— pure sensors – visual, other (3D depth);

— real world capturer – visual/video, other (3D depth);

— recognizer – 3D model/primitives, recognition event, remote;

— tracker – 3D model/primitives, spatial event, remote;

— space mapper – spatial;

— event mapper – location, local;

— execution engine – local, 3D + t;

— renderer – visual; and

— visual display – 3D.

8.3.5 Real-time, multiple remote user reconstructions, condition-based augmentation

A content designer captures offline an approximation of the real world (or just a virtual world) as a 3D model and then authors content by adding in additional 3D virtual objects, including remotely captured users (e.g. live human models), registered within the approximation of the real world or virtual world.

The end users “co-exist”, navigate, interact and communicate with one another in the reconstructed or virtual world using various sensors. Conditions are used to make proper augmentation in a spatially consistent manner by the content designer. 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/virtual world; and

— URL of the processing server and local user capturing server.

According to the MAR component classification scheme in Clause 7, this system class has the following characteristics:

— pure sensors – visual, others (3D depth);

— real world (user) capturer – visual/video, other (3D depth);

— recognizer – recognition event;

— tracker – 3D model/primitives, spatial event, remote;

— space mapper – spatial;

— event mapper – local;

— execution engine – local, 3D;

— renderer – visual, multimodal; and

— visual display – 3D, others.

8.4 MAR Class G: points of interest (POI) — GNSS-based systems

8.4.1 Content-embedded POIs

An MAR execution engine is used by the end user to open an MAR file containing (locally in the scene) POIs from a specific region (see Figure 15). 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 and preferences are displayed. The MAR content also describes how the POIs are displayed, either on the map or in AR view, by creating “map marker” instances and using the metadata provided by the POIs. The content specified in the Information viewpoint is:

— POI data;

— the map marker shape (representation) referenced by the MAR execution engine; and

— user preferences (optional).

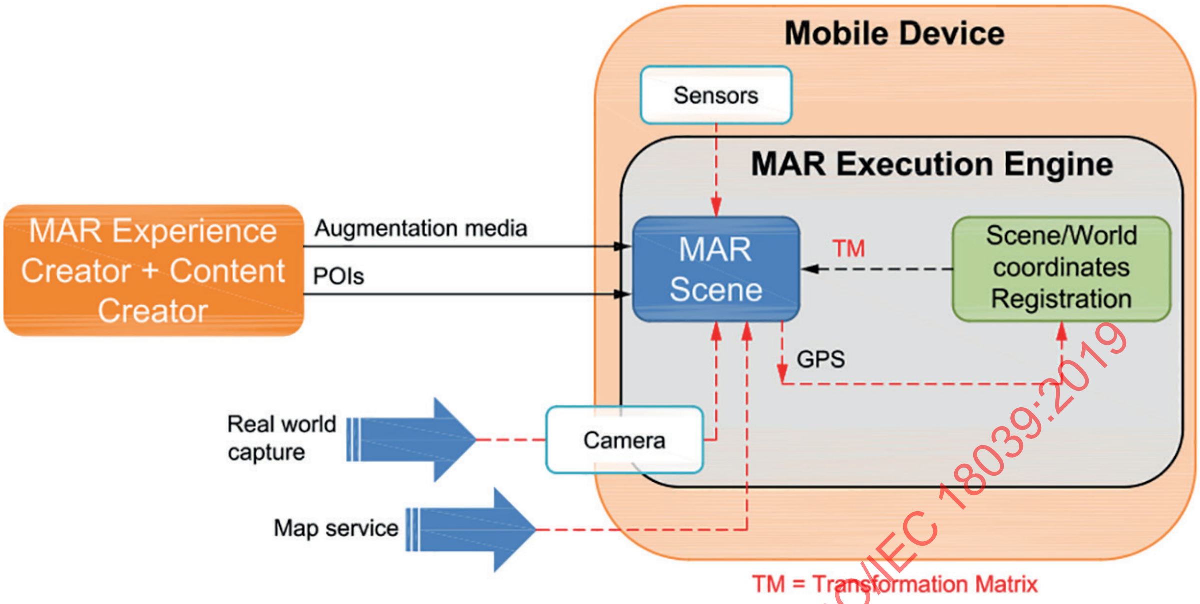


Figure 15 — Content-embedded POI

According to the MAR component classification scheme in Clause 7, this system class has the following characteristics:

— pure sensors – GNSS;

— real world capturer – visual, 2D video;

— recognizer – location, recognition event, local;

— tracker – earth reference, spatial event, local;

— space mapper – spatial;

— event mapper – location;

— execution engine – local, 2D + t;

— renderer – visual;

— visual display – 2D mono; and

— aural display – mono.

8.4.2 Server-available POIs

An MAR execution engine is used by the end user to open an MAR file (see Figure 16). 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 and 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; and

— MapMarker shape and representation.

In addition, a communication protocol should be established between the MAR execution engine and the POI provider.

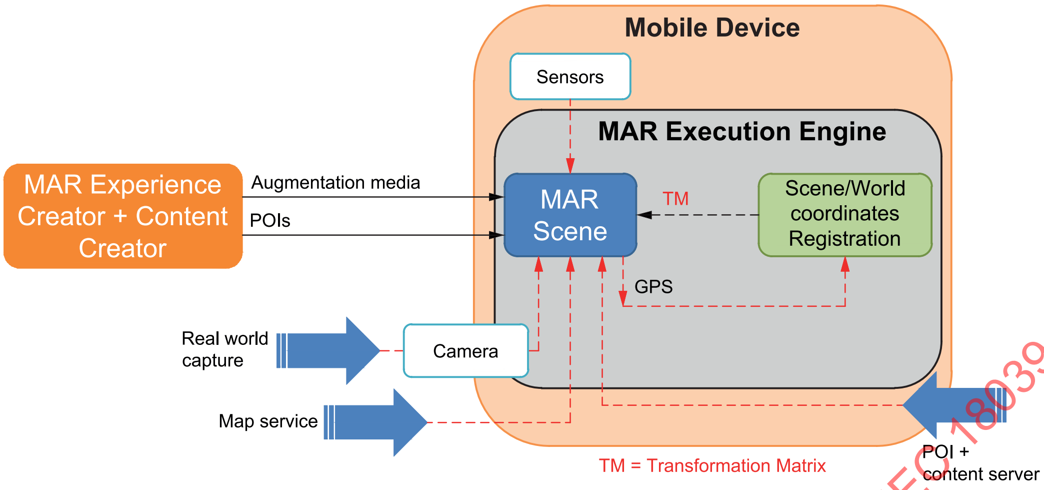


Figure 16 — Server available POI

According to the MAR component classification scheme in Clause 7, this system class has the following characteristics:

— pure sensors – GNSS;

— real world capturer – visual, 2D video;

— recognizer – location, recognition event, local;

— tracker – earth reference , spatial event, local;

— space mapper – spatial;

— event mapper – location, remote;

— execution engine – local, 2D + t;

— renderer – visual;

— visual display – 2D mono; and

— aural display – mono.

8.5 MAR type A: audio systems

8.5.1 Local audio recognition

The device detects the presence of a sound (or the corresponding descriptors) in an audio stream (see Figure 17). Audio can result from a real-time capture using a local microphone, a remote audio source or a prerecorded 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); and

— optional: the recognition frequency and the audio sequence size.

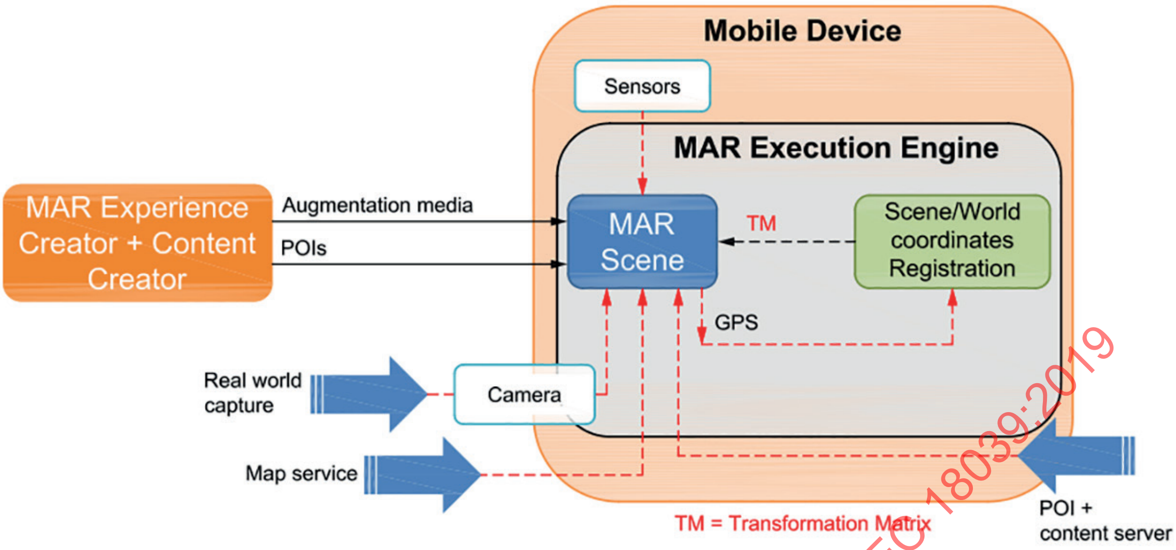


Figure 17 — Local audio recognition

According to the MAR component classification scheme in Clause 7, this system class has the following characteristics:

— pure sensors – auditory;

— real world capturer – auditory;

— recognizer – other (auditory signal), local;

— event mapper – aural, local;

— execution engine – local;

— renderer – visual, aural;

— visual display – 3D; and

— aural display – mono.

8.5.2 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 that are stored in its local (or remote) databases in the audio stream (see Figure 18). 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; and

— optionally, the recognition frequency and the audio sequence size.

In addition, a communication protocol has to be implemented between the MAR execution engine and the processing server.

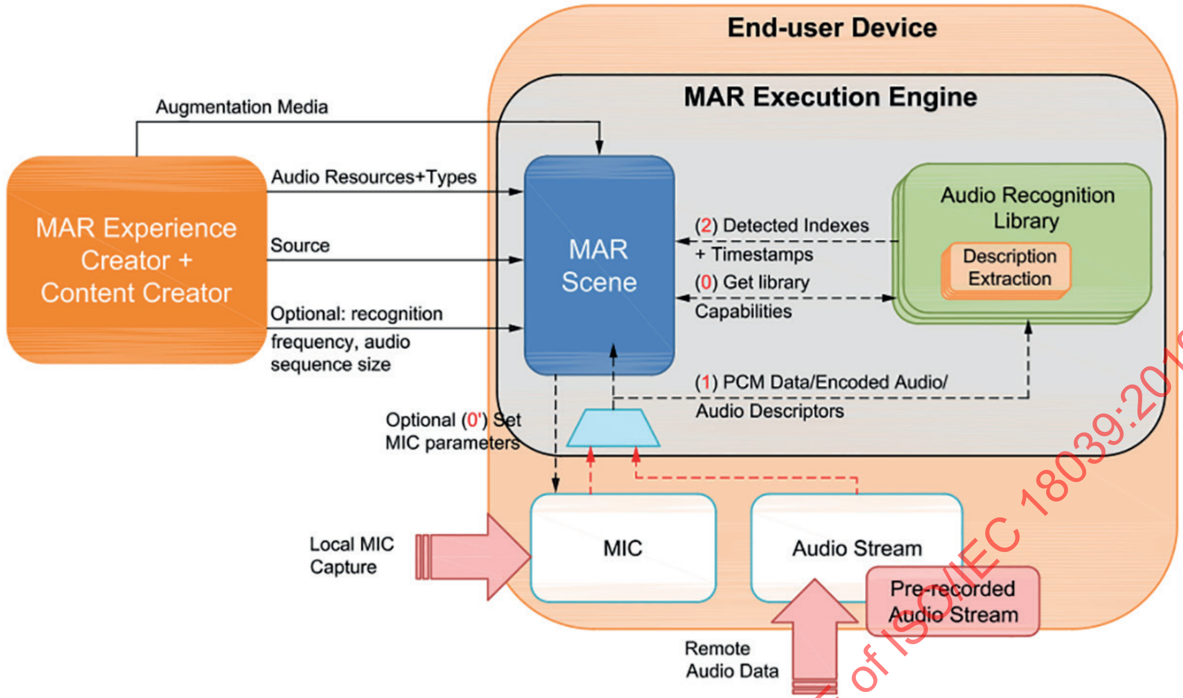


Figure 18 — Remote audio recognition

According to the MAR component classification scheme in Clause 7, this system class has the following characteristics:

— pure sensors – auditory;

— real world capturer – auditory;

— recognizer – other (auditory signal), remote;

— event mapper – aural, remote;

— execution engine – local;

— renderer – visual, aural;

— visual display – 3D; and

— aural display – mono.

8.6 MAR type 3DA: 3D audio systems

8.6.1 Local audio spatialization

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 (see Figure 19).

The content specified in the Information viewpoint is:

— target audio samples (raw or corresponding descriptors); and

— URL of the audio stream (microphone, remote audio source or local track).

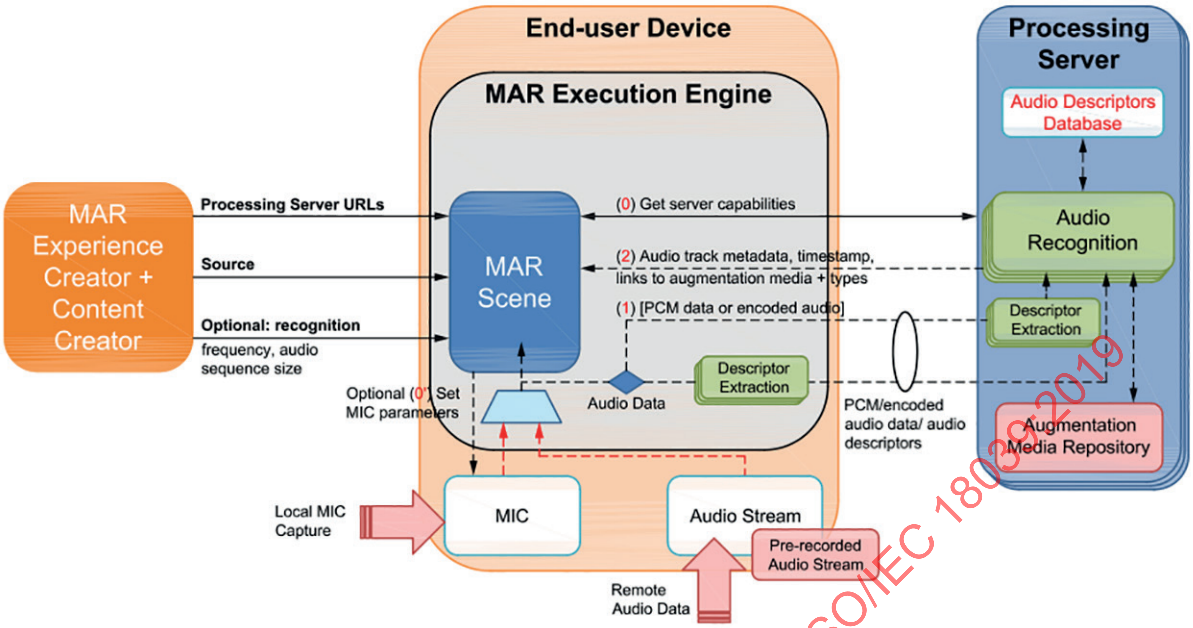


Figure 19 — Local audio spatialization

According to the MAR component classification scheme in Clause 7, this system class has the following characteristics:

— pure sensors – auditory;

— real world capturer – auditory;

— recognizer – other (auditory signal), local;

— event mapper – aural, local;

— execution engine – local;

— renderer – visual, aural;

— visual display – 3D; and

— aural display – 3D.

9 Conformance

Conformance to this reference model is expressed by describing how the aspects of an MAR implementation relate to the MAR system architecture. Conformance of MAR implementations to this document shall be demonstrated by satisfying at least the following requirements.

a) The following key architectural components, as specified in the reference model, shall be present in a given MAR implementation:

— sensors (and real world/object capturer);

— recognizer and/or tracker;

— event mapper;

— spatial mapper;

— execution engine;

— renderer; and

— display and UI.

b) A mapping may be established and evaluated between the components of an MAR implementation and the components of the reference model. The relationships among the implementation of these architectural components shall conform to those in this reference model, as specified in 6.4 and graphically depicted in Figure 6.

c) The interfaces between the architectural components of an MAR implementation shall contain and carry the information specified in 6.4 and 6.5. 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.

d) The API for an MAR implementation shall conform to the concepts specified in 6.4.8 and 6.5 in order to ensure compatibility and software interface interoperability between MAR implementations can be accomplished at least at the abstract API level.

10 Performance

The system performance guideline defines the minimum operational level of MAR systems and establishes possible conformity issues. There are several metrics that can be used to benchmark an 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 that may 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 that where it should be displayed;

— the augmentation precision can depend much on the "calibration" process for and between various MAR components;

— the latency can be measured as the total time needed to process the target object and produce the augmentation; and

— the operating conditions can include lighting conditions, mobility of the target object, sensing distance and orientation, etc.

11 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 can distract users and create potentially dangerous situations. Minimum safety guidelines are necessary to ensure that the given MAR system and content include components for safeguarding the user during the usage of the system. The issue of performance is closely related to that of safety.

The development of policies or software that increase the safety of users, assets and systems reduces risks resulting from:

— dangerous conditions that can 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 use of the MAR contents;

— 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 (such as in construction zones, traffic, operating vehicles and working at height in proximity to hazards);

— human movements necessary for operating an MAR system;

— insufficient level of performance for requirements of MAR system-assisted tasks; and

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

12 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 and the potential to mislead the user. Even as a standalone system, many MAR applications and services, by their nature, tend to deal with a lot of personal information and therefore pose an attractive target for security attacks. In general, MAR systems should exhibit a level of security (for their contents and information) comparable to other digital contents services, such as web documents and systems (http://www.w3c.org/Security/)[37] and geospatial systems (http://www.opengeospatial.org/projects/groups/securitywg)[38].

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 display output as presented by MAR systems; and

— encrypt other communications between MAR components.

13 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 are out of scope of the MAR-RM, but are 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; and

— define the duration of periods during which data access and/or storage is authorized.

14 Usability and accessibility

In any information media, including the MAR, usability of the system and contents plays a very important role beyond user acceptance. Successful commercialization often hinges on usability and safety (see Clause 12). ISO defines usability as "the extent to which a system and/or product can be used by specified users to achieve specified goals with effectiveness, efficiency, and satisfaction in a specified context of use"[6]. For a reasonable level of usability, MAR system designers should take account of the following aspects.

— Readibility/visibility: as an information medium, MAR should provide the minimum level of readability and visibility (similarly of audibility and tactility/hapticity) of the displayed MAR scene.

— Perceptual correctness: the information displayed through the MAR scene, if not designed otherwise, should not distort the original or intended appearance, e.g. in size, depth, 3D, colour and other perceptual qualities.

— Naturalness: the displayed MAR scene should, if not designed otherwise, look natural through situating the augmentation harmoniously along with its neighbouring environment.

— Learnability: how easy it is for users to accomplish basic tasks the first time they encounter the design.

— Efficiency: once users have learned the design, how quickly they are able to perform tasks.

— Memorability: when users return to the design after a period of not using it, how easily they can re-establish proficiency.

— Errors: how many errors users make, how severe these errors are and how easily they can recover from the errors.

— Satisfaction: how pleasant it is to use the design.

— Ergonomic design of hardware and interface: whether MAR devices (such as wearable sensors and displays) are carefully designed to avoid visual, aural and haptic fatigue, stress, discomfort and interferences.

— Accessibility: whether the MAR system and interfaces can be used by users with physical or cognitive impairments.

— Cultural differences: whether the MAR system and interfaces accommodate users of different cultures and speakers of languages other than English.

Similarly to the system performance, more specific guidelines and benchmark processes are needed to ensure and validate for the basic usability and accessibility of an MAR system. Such a validation should ideally be conducted under various and changing operating conditions.

Annex A
(informative)

AR-related solutions and technologies and their relation to the MAR reference model

A.1 Overview

This annex introduces examples of existing AR-related solutions and technologies and how they are related to the MAR reference model. Some of these examples are commercial, trade-marked solutions. However, this document only cites and refers to such systems as mere examples. It is not the purpose of this document to either promote or publicize them. Equivalent products may be used if they can be shown to lead to the same results.

A.2 MPEG ARAF

ISO/IEC 23000-13[7] sets out the augmented reality application format (ARAF), which can be used to formalize a full MAR experience. It consists of an extension of a subset of ISO/IEC 14496-11[8] (MPEG-4 Part 11: scene description and application engine), combined with other relevant MPEG standards [ISO/IEC 14496-1 (MPEG-4 Part 1)[9]], ISO/IEC 14496-14 (MPEG-4 Part 16)[10] and MPEG-V[11]) and is designed to enable the consumption of 2D/3D multimedia, interactive, natural and virtual content. Approximately 200 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[12] data formats for interaction devices (MPEG-V Part 5). MPEG-V[11] provides an architecture and specifies associated information representations to enable the representation of the context and to ensure interoperability between virtual worlds. Concerning MAR, MPEG-V[11] specifies the interaction between the virtual and real worlds by implementing support for accessing different input and 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 Figure A.1. 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.

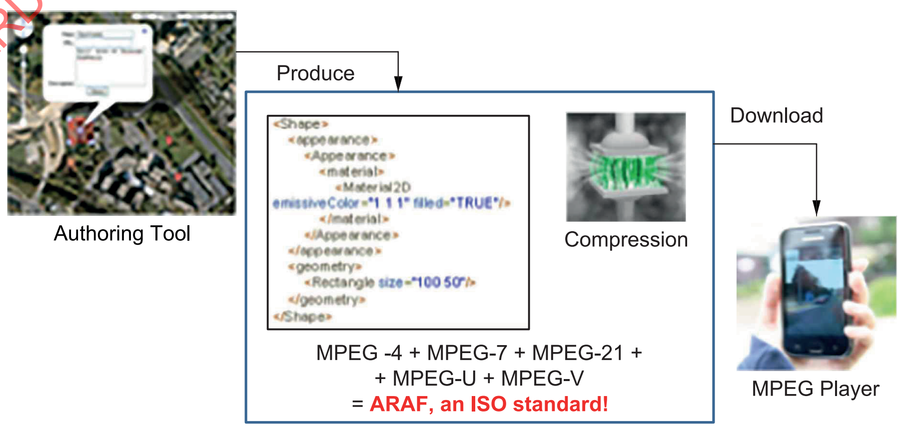


Figure A.1 — The content of AR application format[36]

By using ARAF, content creators can design MAR experiences covering all classes defined in Clause 7, 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.

A.3 KML, ARML, KARML (see Figure A.2)

KML (Keyhole Mark-up Language)[13] 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 extend 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)[14] is an extension to KML and allows for richer types of augmentation for location-based AR services. KARML[15] goes further by adding even more decorative presentation styles (e.g. balloons, panoramic images), but more importantly it proposes a method of relative spatial specification of the augmented information for their exact registration. These KML-based approaches use Open Geospatial Consortium (OGC) standards for representing GPS landmarks, but for the rest they use a mixture of non-standard constructs, albeit somewhat extensible (perhaps in an ad hoc way and driven mostly by specific vendor needs), for augmentation (e.g. vs. HTML or X3D).

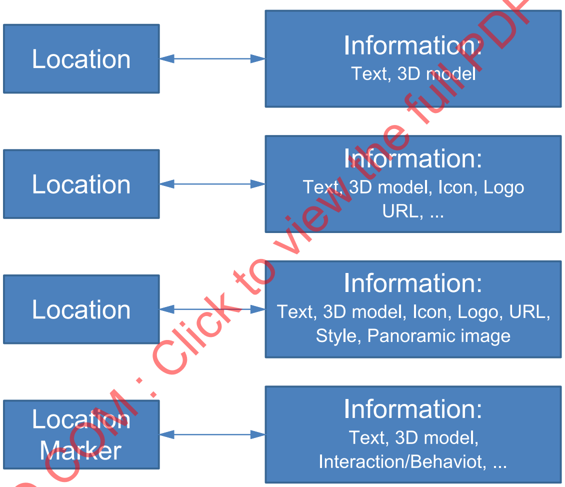


Figure A.2 — Content models of KML, ARML and KARML showing how they associate location with augmentation information

A.4 X3D

X3D[16] is a publicly available standard (ISO/IEC 19775-1) 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), 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 was originally developed to represent synthetic and 3D graphical virtual objects and scene, but can also be naturally extended for MAR because MAR systems are implemented as VR systems. For example, video see-through AR systems are implemented by 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 was 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.

A.5 JPEG AR (see Figure A.3)

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[17].

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

— interface between the sensor and AR recognizer or AR tracker, for which this document specifies information to be transmitted from the sensor to the recognizer or tracker;

— interface between the AR recognizer or AR tracker, and the event handler, for which this document specifies data and information to be composed in the recognizer or tracker and transmitted to the event handler for processing described operations according to the information;

— interface between the event handler and the content repository, for which this document specifies information and corresponding operations that the event handler and content repository manipulate; and

— interface between the event handler and renderer, for which this document specifies information to be transmitted from the event handler to the renderer for displaying composite images.

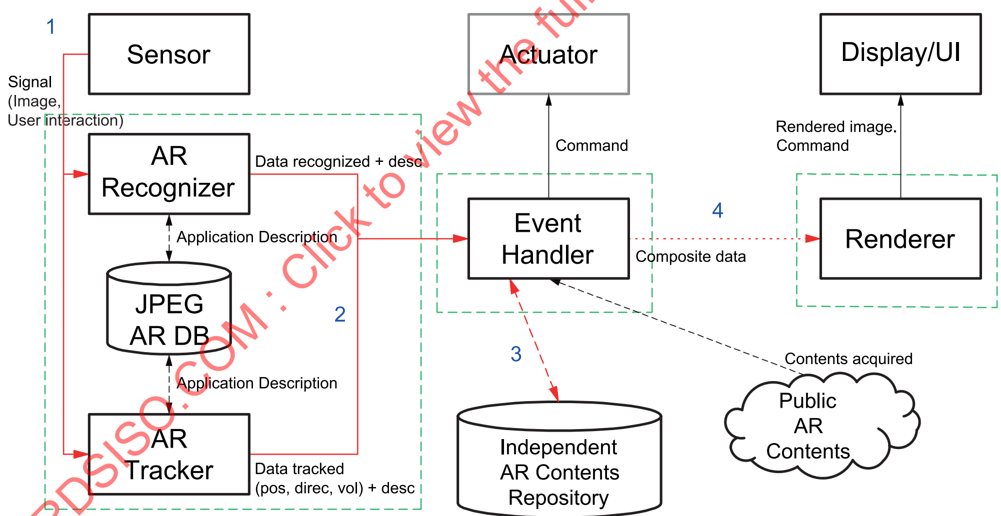


Figure A.3 — The JPEG AR framework architecture illustrating its main components[17]

A.6 ARToolKit and OSGART

ARToolKit[18] is a computer tracking library for the purpose of creating AR 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 AR: viewpoint tracking and virtual object interaction. ARToolkit, which by itself does not have any scene graph support, has been merged with an open source VR platform (with scene graph support), namely the OpenSceneGraph (OSG)[19]. This version of OSG is called the OSGART.

A.7 OpenCV and OpenVX

OpenCV (Open Source Computer Vision)[20] is a library of programming functions aimed mainly at real-time computer vision. It was developed by the Intel Russia research centre in Nizhny Novgorod and is now supported by Willow Garage and Itseez. It is free for use under the open source BSD licence. 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[21].

A.8 QR codes and 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 MAR (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 A.4 — Example of a QR code that can be used as marker for an MAR system and contents

Annex B
(informative)

Use case examples and coverage by the MAR reference model

B.1 Overview

This annex introduces use case categories and examples for MAR, and for each example provides the mapping to the MAR system architecture and corresponding viewpoints. Some of the use cases in this annex are research output and/or commercial solutions. The information is given for the convenience of users of this document and does not constitute an endorsement by ISO of the products named. Equivalent products may be used if they can be shown to lead to the same results.

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 UI for virtual objects about which a person asks one or more questions, often in a sequence. Experiences in the Guide use case category often aid users 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 are able to see, hear or feel it.

B.2.3 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, repairing and 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)

B.3.1 What it does

MagicBook[22] is a marker-based AR 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.

B.3.2 How it works

A marker is printed in a book and viewed using a video see-through display as illustrated in Figure B.1. The marker is recognized and tracked by the camera attached to the display.

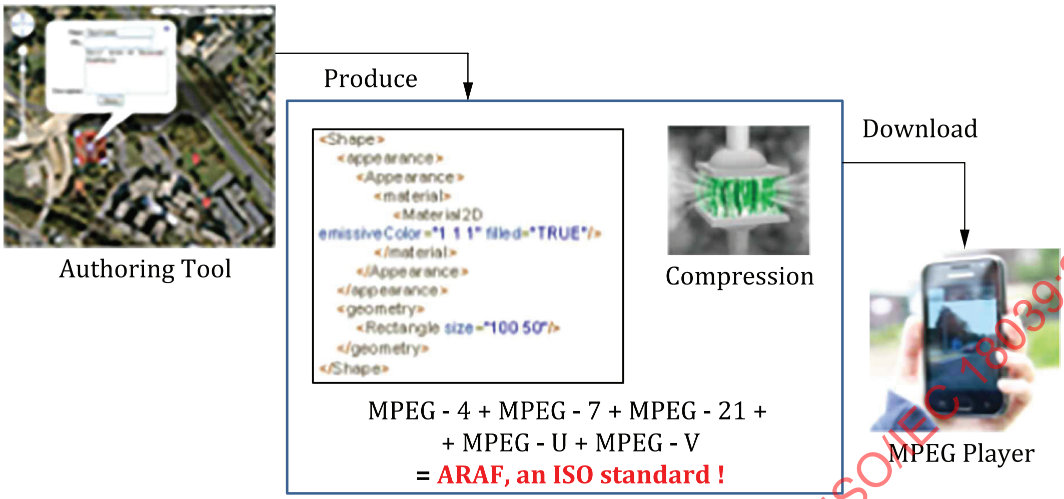


Figure B.1 — A user viewing the MagicBook using a hand-held video see-through display[22]

B.3.3 Mapping to MAR-RM and its various viewpoints

Table B.1 maps the key components in the MAR-RM to the corresponding components in the MagicBook system.

Table B.1 — Correspondence between MAR-RM components and major components in the MagicBook

MAR-RM component	Major components in the MagicBook
Sensor	Camera
Real-world capture	Live video
Target physical object	2D marker
Tracker and recognizer	Template-based recognition, homography-based 2D marker tracking
Spatial mapping	Hard coded
Event mapping	Hard coded
Execution engine	Hard coded
Rendering	OpenGL
Display and UI	Video see-through and headphone

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

B.4.1 What it does

Human Pac-Man[23] is an outdoor interactive entertainment system in which the video game Pac-Man™1) is played outdoors with humans acting as pac-men and ghosts. Virtual cookies are overlaid in the physical environment. ARQuake[24] is an outdoor interactive entertainment system developed using

1) Developed by Nanco in 1980.

markers. A marker is printed in a book and viewed using a video see-through display as illustrated in Figure B.2. The marker is recognized and tracked by the camera attached to the display.

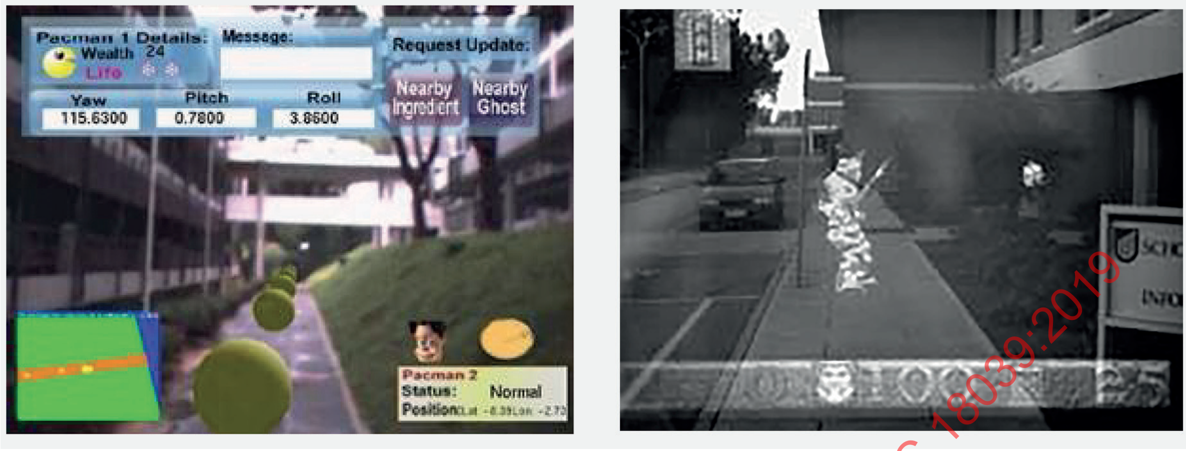


Figure B.2 — View of Human Pac-Man [8] as seen by the user

B.4.2 How it works

The user wears a head-mounted display whose location is tracked by a GPS. 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.

B.4.3 Mapping to MAR-RM and various viewpoints

Table B.2 maps the key components in the MAR-RM to the corresponding components in the ARQuake system.

Table B.2 — Correspondence between MAR-RM components and major components in Human Pac-Man and AR Quake

MAR-RM component	Major components in Human Pac-Man	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 and 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 and 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)

B.5.1 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 mannequin[25].

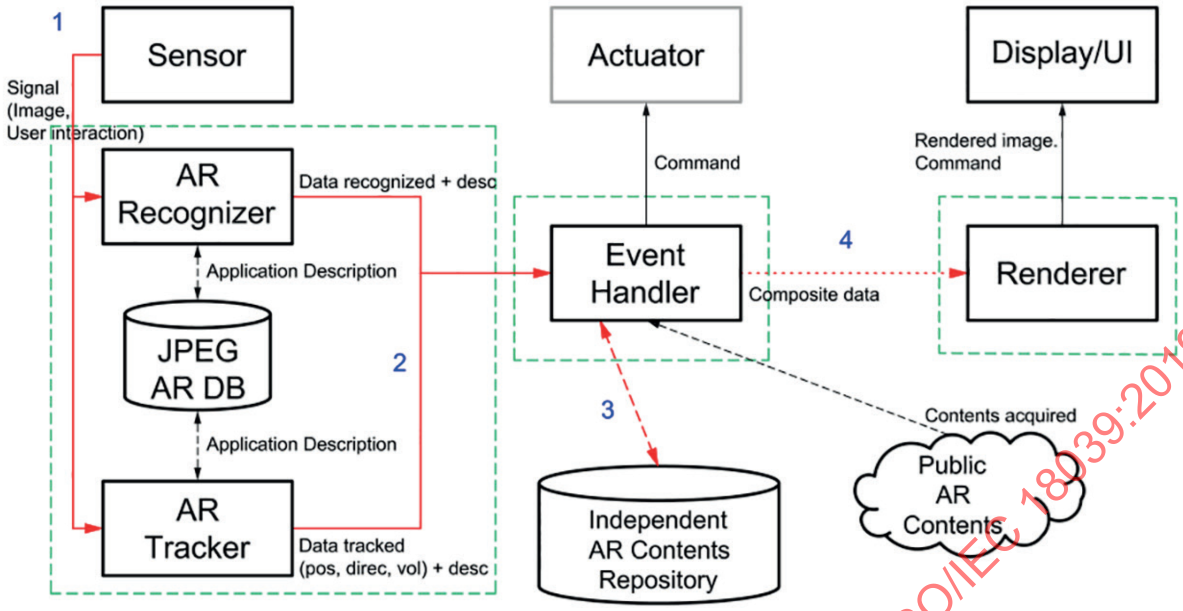


Figure B.3 — Haptic information augmented in modulated stiffness[25]

B.5.2 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 the 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.

B.5.3 Mapping to MAR-RM and various viewpoints

Table B.3 maps the key components in the MAR-RM to the corresponding components in the example augmented haptic system.

Table B.3 — Correspondence between MAR-RM components and major components in the augmented haptics

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 and UI	Haptic manipulator