Int J Interact Des Manuf (2014) 8:293–304
DOI 10.1007/s12008-013-0199-7

ORIGINAL PAPER

A context-aware augmented reality system to assist the maintenance operators

J. Zhu · S. K. Ong · A. Y. C. Nee

Received: 8 April 2013 / Accepted: 14 December 2013 / Published online: 1 January 2014

Abstract The increasing complexity and the technology advancement of equipment pose challenges to the maintenance operators nowadays. In this paper, a context-aware augmented reality (AR) system is proposed to assist the operators in routine and ad hoc maintenance tasks by providing them with context relevant information. This system integrates AR technology and context-awareness. It is able to analyze the contexts of the maintenance tasks and provide relevant and useful information to the operators through rendering the information on the real equipment and environment. In addition, the authoring of context-aware AR contents, which is essential for developing such applications, is explored. The properties of AR contents and their context-adaptation schemes are discussed, and a low-level authoring approach is proposed for constructing context-aware AR information. Finally, a case study is conducted to evaluate the system performance, and the results demonstrated that the system can adapt to dynamic maintenance situations to provide context relevant information to assist the maintenance operators.

Keywords Augmented reality · Authoring · Context-awareness · Maintenance and repair

J. Zhu · S. K. Ong · A. Y. C. Nee
NUS Graduate School for Integrative Sciences and Engineering,
National University of Singapore, 28 Medical Drive,
Singapore 117456, Singapore

S. K. Ong (✉) · A. Y. C. Nee
Mechanical Engineering Department, National University
of Singapore, 9 Engineering Drive 1, Singapore 117576,
Singapore
e-mail: mpeongsk@nus.edu.sg

1 Introduction

Maintenance plays an important role in the equipment life cycle as it updates and improves the equipment performance and safety. However, the increasing complexity and technology advancement of the equipment nowadays has led to the difficulties and challenges for the maintenance operators to retrieve relevant and correct maintenance information and complete the preventive or corrective maintenance tasks efficiently.

Augmented reality (AR) technology is the superimposition of computer graphics over real objects/scenes, viewing through Head-Mounted Devices (HMD) or handheld displays [1]. AR has been used to replace the paper-based manuals and computers to provide maintenance instructions to the technicians and it has been validated as a promising tool to assist the operators [2]. For example, Morkos et al. [3] developed a system based on mobile device that connects with the databases through mobile broadband communications, and provides information to the users in the manufacturing environments using AR technology. These reported works have demonstrated the benefits of employing AR to assist the operators in the maintenance activities.

Context is any information that can be utilized to characterize the situation of an entity, where the entity can be a place, a person, or an object that is relevant to the interaction between a user and an application, such as time, location, activities [4]. A system can be deemed as context-aware if it can collect, reason, and utilize context information, and adapt its functionality to the varying contexts [5]. For example, the operator's location can be used as the context to infer the equipment information that the operator may be interested in. When the operator is within the vicinity of a specific equipment, the information of the equipment is deemed as useful and can be rendered to him.

The integration of context-awareness and AR is a relatively new research topic, and several generic frameworks and application systems of context-aware AR have been reported recently [6–8]. In these reported systems, different information can be provided to the users according to the various contexts, and these prototype systems have demonstrated the usefulness of integrating context-awareness with AR technology. However, the information augmented is generally assumed to be available and prepared a priori by AR system developers. The properties of the AR contents and their adaptation scheme have yet to be thoroughly explored, and there is a lack of authoring approach for the system developers to create context-aware AR contents.

In this paper, a context-aware AR-assisted system is proposed to assist the maintenance operators by providing them with context relevant information. The system is an integration of AR technology and context-awareness. This research highlights the issue of providing useful and relevant information, and the proposed system is novel as it is able to collect and analyze the operator's contexts, i.e., context-awareness. A low-level authoring approach that can be used to construct context-aware AR information is proposed.

The rest of the paper is organized as follows. Section 2 reviews the state-of-the-art AR maintenance systems, and discusses the motivation and importance of the proposed research. Section 3 gives an overview of the system architecture, presents the details of the proposed concept and authoring of context-aware AR contents, and describes the adaptation scheme of authored AR contents to the contexts. Section 4 describes the system implementation and the case study conducted. Section 5 summarizes the study and presents the future work.

2 Related work

2.1 AR maintenance systems

The application of AR technology in assisting the maintenance operator started in the early 1990s, and the objectives of the reported work have changed from demonstrating the benefits of applying AR to maintenance to improving the usability of such research-based AR systems in routine and ad hoc maintenance activities. As one of the pioneering AR maintenance systems, Knowledge-based Augmented Reality for Maintenance Assistance (KARMAR) [9] was primarily designed for providing maintenance instructions to the user. The maintenance database, which consists of maintenance instructions, is developed in advance and considered as maintenance "knowledge". With the sensors pre-installed on a printer, KARMAR is able to estimate the relative position of the user's head to the printer, retrieve the corresponding instructions and augments them in the user's view through the HMD. Although rather rudimentary, this prototype system has validated the benefits AR can bring to equipment maintenance. To improve the usability of current research-based AR maintenance systems in real maintenance scenarios, a few issues have been worked on by several researchers since the mid 1990s [10–19].

Tracking is a fundamental issue in AR. Accurate and stable tracking techniques are especially important for applying AR in maintenance applications as equipment maintenance requires a high level of skill and precision. Many works have been reported [10–13,16–20] although the current status in the tracking research is still considered immature. Many AR maintenance systems [10–12] have replaced the sensor-based tracking method used in KARMAR [9] with the marker-based tracking methods, as they can provide a relatively high level of accuracy and the required software and hardware have become common items. However, marker-based tracking methods require the environment to be prepared in advance and the markers to be completely visible during the tracking process. Thus, researchers have begun to focus on markerless tracking techniques. Model-based tracking methods based on 3D models that match the point and edge features extracted from each frame captured online and those extracted from pre-built CAD models have been proposed [13,17]. These methods remove the need to attach markers in the maintenance area, but the CAD models are usually unavailable a priori and it could be costly to build them in advance. Neubert et al. [16] proposed a solution which allows rapid generation of rough, appearance-based edge models comprising group of planes. A handheld device with a camera is used to capture a video of the object of interest, and a list of edges can be annotated on a set of keyframes. The models that have been generated will be used for estimating the camera poses online through edge matching. Caponio et al. [18] proposed a similar approach that analyzes the video feed of the application environment to extract and label the feature points offline, which will be used to estimate the position and pose of the objects online.

The simultaneous localization and mapping (SLAM) technique removes the offline labeling process, and allows simultaneous map construction and tracking online. Klein and Murray [20] presented the parallel tracking and mapping (PTAM) system that is able to construct a 3D map consisting of feature points and keyframes of unknown environments. The camera pose can be estimated through analyzing the feature points in each incoming frame and matching the features with those in the 3D map. Yan et al. [19] explored vision-based SLAM using line features as they are abundant in nuclear power plants. In the proposed method, Plücker coordinates [19] are used to represent the 3D lines, and the Gaussian sum is used to approximate the initial state of the features and update the new observations. Such tracking techniques require limited offline preparation of the application scenario and provide competitive tracking performance.

Human-system interaction is another issue that hinders current AR maintenance systems in routine and ad hoc maintenance activities. There is yet a commonly acceptable interaction approach, and many useful techniques and tools have been reported [21–24]. The prevalent human-system interaction tools, such as mouse, keyboard, and touch screen, have been employed and reported in several systems [12,14,15]. These tools require less user training, but they usually distract the users from the maintenance operations as they have to halt the maintenance work temporarily and use the interaction tool to interact with the system.

Recently, several new interaction approaches have been proposed [21,22]. Goose et al. [21] developed a speech-enabled AR framework for mobile maintenance. The system enables the user to "talk" to the equipment as if he is talking to another person. Their system identifies the user's location via the markers positioned in the worksite in advance, and triggers a context-specific speech dialog. However, speech-based interaction should preferably be kept to a minimum as the application environment of AR maintenance systems, such as factories, are typically very noisy. Eye gazing based interaction has been proposed in the wearable augmented reality system [22]. By equipping a see-through HMD with an eye tracker, it is able to track the user's eye movements for interacting with the system. However, several challenges, such as the "Midas Touch" problem [25], have to be solved before gaze-based interaction can be widely adopted. Virtual Interaction Panel (VirIP) [23] has been proposed to utilize daily pen-like objects, e.g., common pens and fingers, as interaction tools. The interaction pen can be tracked in real time and used to trigger the relevant buttons in the VirIP for assembly instructions. Henderson and Feiner [24] proposed the Opportunistic Controls (OC) techniques, which were designed based on two criteria, namely, (1) limit extraneous hand, head and eye movements beyond the user's immediate vicinity, and (2) provide passive haptic feedback without modifying the application environment. An OC interface aims to enable user interaction with the system through touching natural surfaces within the task environment. For example, the system renders a virtual button on the raised geometry of the engine, and a user can press the button while receiving passive haptic feedback from the raised geometry underneath.

The third issue is the display devices that render the augmented environment to the users. HMDs [9–11,13] and handheld devices [12,15] are the prevalent display devices employed in the reported systems. HMDs are worn on the operator's heads, but it has been reported that these maintenance operators do not like to wear any type of HMD [15]. Handheld devices require the user to have at least one hand to hold and interact with the devices. It is inconvenient as the user's hands are usually engaged in maintenance tasks. Thus, a few alternative display devices, such as chest worn devices [26], have been proposed, but well accepted display devices are not available yet.

2.2 Motivations

Although significant efforts have been devoted to applying AR technology to equipment maintenance, no single AR system has yet been proven to be well accepted by the industry as a solution to the maintenance tasks [27]. An important issue that has generally been neglected is the provision of context relevant information to the users and the real-time authoring and updating of this information. In real maintenance scenarios, there can be many maintenance operators with varying levels of expertise who are carrying out different maintenance tasks, and these users may need totally different information. Thus, AR maintenance systems have to be context-aware, i.e., providing information that will be useful for each particular user by analyzing his contexts in order that these systems can be applied in routine and ad hoc maintenance activities. In addition, the authoring of context-adaptable AR contents, which is essential for developing context-aware AR services, has yet to be thoroughly explored. This research fills these gaps through exploring the development of a context-aware AR maintenance system and proposes a low-level authoring approach for constructing context-aware AR contents.

3 System overview

The proposed system consists of five modules, namely, Context Collection (CC) module, Context Management (CM) module, AR-based Visualization (ARV) module, and Virtual Content (VC) module, and Information Instance (II) module (Fig. 1). The CC module interfaces with the users and data sources, e.g., web servers and sensors, and interprets the collected raw data to form the low-level contexts. The CM module updates the collected low-level contexts in the

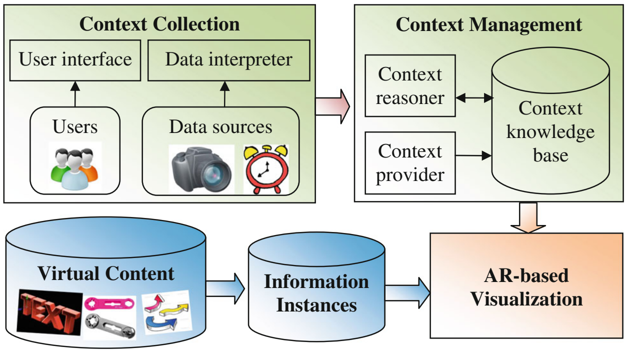


Fig. 1 System framework

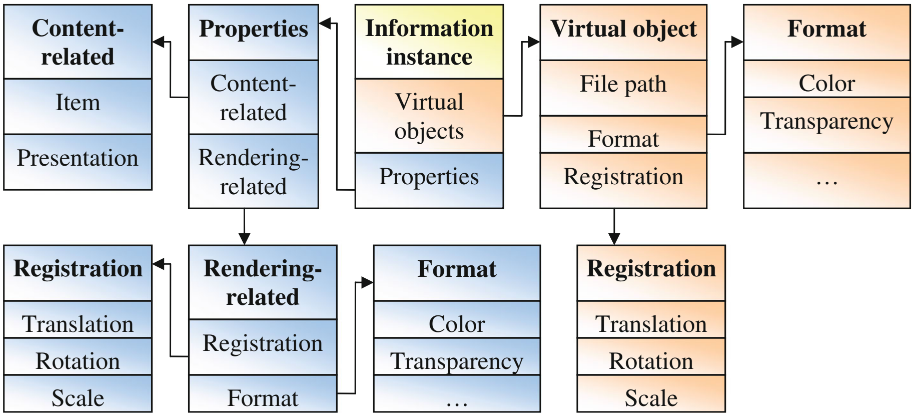


Fig. 2 Information instance

context knowledge base, derives the high-level contexts, and processes the context knowledge queries from the ARV module. The VC module stores the virtual content that can be rendered in the AR environment, including texts, images, and CAD models. These virtual objects are authored as information instances that can be adapted to the contexts and stored in the II module (Sect. 3.1). The ARV module adapts the authored information instances to the contexts provided by the CM module and renders context relevant information virtually to the user. Details of each module will be described accordingly.

3.1 Information instances module

Providing the appropriate information with suitable presentation format to the users is one of the objectives of integrating context-awareness with AR technology. To achieve this, the virtual objects stored in the VC module are authored as information instances and stored in the II module. An information instance (Fig. 2) comprises of a combination of virtual objects that are used to represent certain information, and it has the following context-adaptable properties that are used to relate the content and rendering of the information instances to the contexts:

- Content-related properties: these properties characterize the information that is described in the instance and the group of virtual objects used to describe this instance; they allow the content of an information instance to be linked to the contexts.
- Rendering-related properties: these properties characterize the rendering of the virtual objects, and relate the presentation of the information instances with respect to the contexts.

The content-related properties are item and presentation. Property item represents an instance of the domain knowledge, e.g., instructions on the lubrication of an equipment, and it can be related to the contexts of the various status of a task, the user's interest, etc. The presentation property refers to the combinations of virtual objects used to represent an item, and it can be intuitive presentation, symbolic presentation, and textual presentation. Intuitive presentation is easier for the users to comprehend, and textual presentation consumes least computing power of the mobile devices. Thus, the presentation type selected should correspond to the perceptual capabilities of the users and the computing power of the mobile device, etc.

Rendering-related properties refer to the display of the information instance in the real environment, and they are classified into two categories, namely, Format and Registration. Format manages the visual properties of augmented information, and it consists of color, transparency, etc. Based on the requirements of an application, certain properties can be defined and adapted to the contexts in this application, e.g., user's preference. Registration influences the registration of the virtual content in the AR environment, and this category of properties include translation, rotation, and scale, These properties can also be adapted to specific contexts, e.g., the scale of the virtual objects can be adapted based on the distance from the user so that the objects can appear larger when the user moves closer to the objects.

Authoring an information instance consists of adding virtual objects, defining the properties of this instance, and relating the properties to the contexts. ARToolKit [28], which provides a tracking approach based on physical markers and interfaces to insert virtual objects through extending the provided program, is employed in the system. As shown in Fig. 3, various virtual objects can be loaded into the scene, arranged spatially relative to the physical marker, and the rendering parameters, e.g., color and transparency, can be adjusted; a rectangle is added to outline the switches, an arrow is inserted to point to the relay connection, and the texts are added to describe the components. Finally, the virtual objects in the information instance and their rendering parameters are recorded and stored in the II module. The properties of an information instance and their adaptation to the contexts will be discussed in the next section.

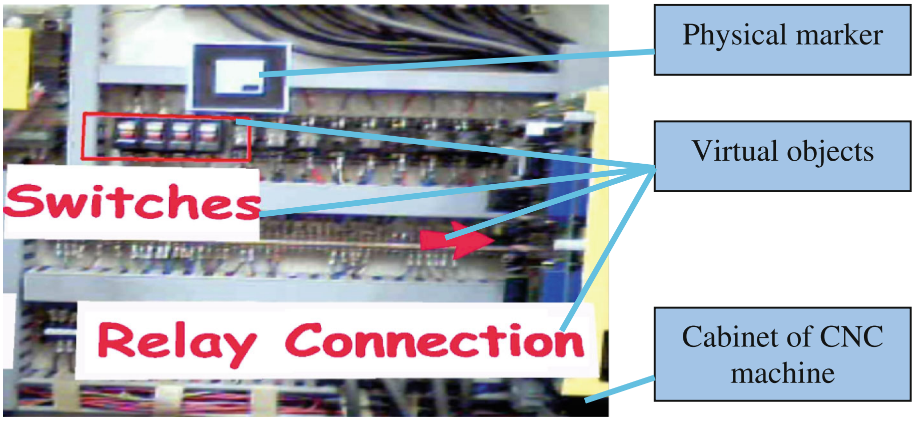


Fig. 3 Virtual objects of an information instance

3.2 Context management module

The CM module consists of the context provider, context reasoner and context knowledge base. The CM module sends real time context information to the ARV module so that context-aware AR services can be provided to the users.

3.2.1 Context knowledge base

The context knowledge base stores the context model of the application environment and the rules defining the logical relationships among the contexts. Ontology is used to model the contexts of the application environment as it is independent of programming languages and supports first-order logic context reasoning. An ontology-based model, Context Ontology for AR-assisted Maintenance Service (COARMS), is constructed based on CONON [29]. There are three concepts in the ontology model, namely, class, subclass and property. Each class can have several subclasses, and each class/subclass can have certain properties.

As shown in Fig. 4, COARMS has five classes, namely, CompEntity, Location, Person, Activity, and InformationInstances. The Activity class represents the abstraction of the various maintenance works, and it has a set of properties including Name, Status, etc. It has two subclasses, namely, CorrectiveMaintenance and PreventiveMaintenance. The Person class represents all the personnel involved in the maintenance work, and it has properties of Name, ExpertiseLevel, etc. The Location class represents the abstraction of physical locations, and it has properties, such as Name and Time. In the CompEntity class, various sensors in the context-aware environment can be registered in the SensorDevice subclass, e.g., camera. The properties Translation and Rotation encode the real time spatial information of the camera. The CompDevice subclass represents the mobile devices used, e.g., laptop and PDA, and it has the properties of BatteryLevel, CPUUsage, etc. The Equipment class represents different kinds of equipment in the maintenance environments, and it has the properties of Model, ID, etc.

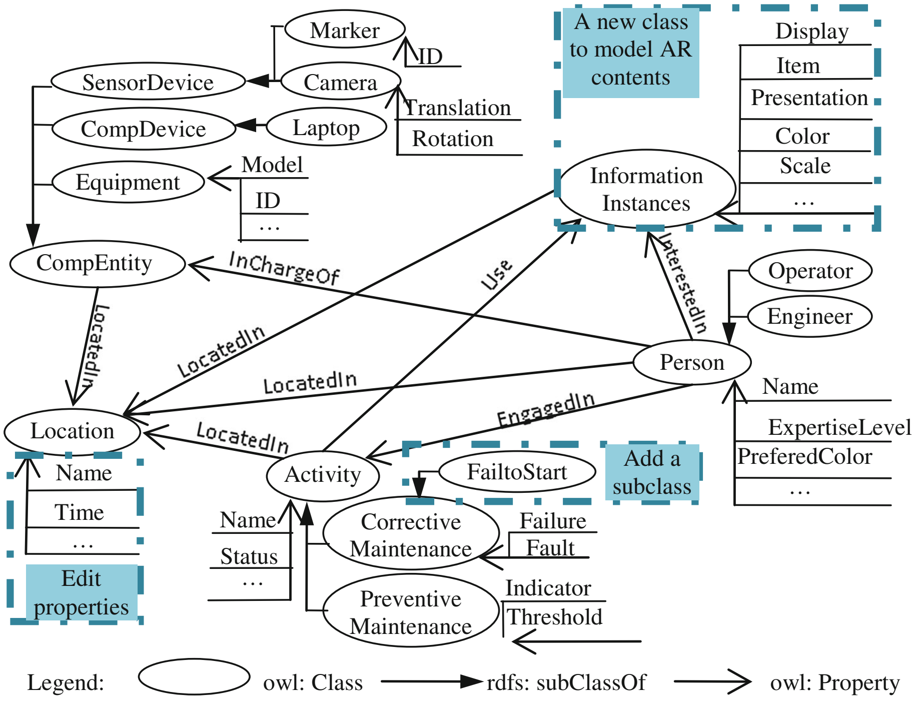
The InformationInstances class models the context-aware AR contents, i.e., each information instance authored should be registered in this class. The InformationInstances class has the properties of display and those discussed in Sect. 3.1. The display property is the property that indicates whether the information instance should be displayed in the particular contexts, i.e., value of True or False where True would mean the instance will be displayed.

Thus, COARMS specifies the intrinsic semantics of maintenance environments and it provides a basis for context management in various maintenance scenarios. To construct a context ontology for a specific maintenance environment, instances of the concepts (class, subclass, and property) can be inserted or deleted. As shown in Fig. 4, the activity to restore the equipment which fails to start can be modeled as a subclass of the CorrectiveMaintenance class, and the properties of each class/subclass can be edited.

Besides ontology, logical rules defining the logical relationships among the instances of concepts in ontology are stored in the context knowledge base. The rules are used for context reasoning and linking the properties of the information instances to the contexts. SWRL (Semantic Web Rule Language) [30], an expressive OWL-based rule language is employed in the proposed system to define the rules. A SWRL rule consists of a body (a set of conditions) and a head (a set of actions to be taken if all the conditions are satisfied). A condition and/or action in a rule is a rule atom, and the instances of the concepts defined in the ontology can be referred in the rules. The rules can be defined via the interfaces provided by OWL API, and will be encoded in OWL descriptions in ontology.

To adapt the information instances to the contexts, two sets of rules, namely, content adaptation rules and rendering adaptation rules need to be defined. The content adaptation rules filter relevant information instances according to the contexts

Fig. 4 COARMS



```
PreventiveMaintenance (Overhaul) ^ Status (Overhaul, Step2) ^
Engineer(?a) ^ EngagedIn(?a, Overhaul) -> InformationInstances
(Switch&Relay) ^ Display (Switch&Relay, True)
```

Fig. 5 Content adaptation rule

```
Engineer(?a) ^ PreferredColor(?a, ?b) ^
InformationInstances(Switch&Relay) ^ Display(Switch&Relay, True) ->
Color (Switch&Relay, ?b)
```

Fig. 6 Rendering adaptation rule

as they define the relations between the contexts and the content of information instances. Each rule consists of a body that specifies a set of relevant contexts and a head that defines the value of the display property of the information instance. For example, the content adaptation rule as shown in Fig. 5 defines the information instance Switch&Relay using texts and symbols (symbolic presentation) to outline the switches and relays that should be checked (item) to be provided to the maintenance engineer at the second step of the preventive maintenance activity of Overhaul.

Rendering adaptation rules relate the rendering-related properties of an information instance to the contexts. Adapting a particular rendering related property to the contexts comprises of two steps. Firstly, the property should be registered in the context ontology as a property of the information instance. Next, logical rules have to be defined to link the properties to the contexts. For example, to render the information instance Switch&Relay in the user's preferred color, color can be added as a property of the information instance Switch&Relay and the logical rule can be defined (Fig. 6).

3.2.2 Context provider

The context provider acquires the low-level context data provided by the CC module (Sect. 3.3) and encodes them in OWL descriptions to update the context knowledge base. For example, the values of the property ExpertiseLevel collected from user inputs and the property OperationTime of Equipment CNC1 acquired from the digital timer are encoded as Fig. 7a, b respectively. Whenever new contexts are collected, the CM module will notify the ARV module to update the AR services.

3.2.3 Context reasoner

The context reasoner derives high-level contexts from the low-level contexts that have been collected and handles context knowledge queries from the ARV module. The high-level contexts are derived based on the real time contexts and logical rules defined. For example, the display property of the information instances Switch&Relay is inferred as true when

(a)

```
<owl:NamedIndividual rdf:about="#Jack">
   <rdf:type rdf:resource="#Engineer"/>
   <ExpertiseLevel
   rdf:datatype="http://www.w3.org/2001/ XMLSchema#string">High
   </ExpertiseLevel>
</owl:NamedIndividual>
```

(b)

```
<owl:NamedIndividual rdf:about="#CNC1">
   <rdf:type rdf:resource="#Equipment"/>
     <OperationTime
     rdf:datatype="http://www.w3.org/2001/XMLSchema#integer">40
      </OperationTime>
</owl:NamedIndividual>
```

Fig. 7 Context provider

the status of the Overhaul activity is Step 2 (Fig. 5). Pellet reasoner [31], which is commonly used for ontology-based data management, is employed in the system as the context reasoner. OWL API provides interfaces with the Pellet reasoner and functions for context reasoning and knowledge queries.

3.3 Context collection module

The CC module acquires the low-level contexts from the users and other sources, e.g., web servers and sensors. A mobile user interface based on the fingertip tracking [32] and virtual panel [23] techniques have been developed for user inputs. The fingertip tracking technique tracks the user's fingertips in the image plane in real time, and the single fingertip pointing gesture is supported in the system for the user to interact with the virtual panel. Single pointing fingertip tracking is a three-step process. First, the user's hand is segmented from the background based on the color feature. Next, all the fingertips are detected from the hand contour using a curvature-based algorithm. Among all the detected fingertips, the one that is farthest away from the center of the hand region is deemed as the pointing fingertip.

A virtual panel used in the system is based on the screen coordinate, and it is a virtual display of computer augmented information, such as virtual buttons. Therefore, with the tracked pointing fingertip, the user can activate the virtual buttons to achieve the interactions with the system. As shown in Fig. 8, the user's tracked fingertip is drawn as a yellow dot, and there are two virtual buttons, Previous and Next, displayed in the virtual panel. To activate a virtual button, the user can place his fingertip on it for a predefined time range.

In addition, the data interpreter collects the context information through processing the raw data provided by various sources, e.g., web servers, applications and sensors. The data interpreter has individual functional modules to interact with each data source. For example, the data interpreter collects the local time from the digital timer, and derives the equipment operating time through subtracting the local time by the start operating time. The data interpreter has the tracking functional module which processes the real time images of the real environment captured by the camera, and outputs the desired low-level contexts in the ontology, e.g., translation and rotation of the camera, and ID of the marker in view. The acquired low-level contexts will be transmitted to the CM module to update the context knowledge base.

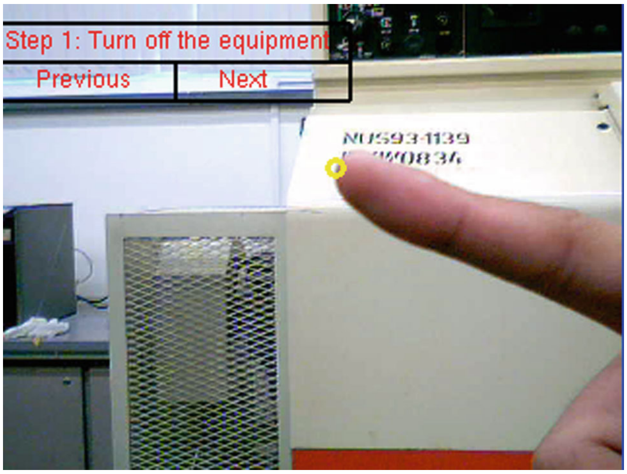


Fig. 8 Tracked fingertip and Virtual Panel

3.4 AR-based visualization module

The ARV module realizes the integration of context-awareness and AR technology. This module queries the real time context knowledge and adapts the authored information instances, so that information with the right content and presentation can be provided to the users. There can be

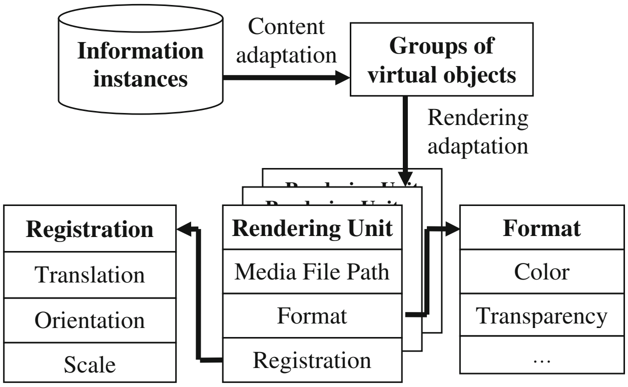


Fig. 9 Context adaptation scheme

many information instances authored and stored in the II module, and providing context relevant information involves adapting the properties of the authored information instances according to the contexts. The context adaptation scheme of the information instances consists of content adaptation and rendering adaptation that correspond to the content-related and rendering-related properties of an information instance respectively (Sect. 3.1).

In content adaptation, the information instances that should be rendered in the current contexts will be filtered through querying the value of their property display, i.e., information instances with the value of their display property as True will be filtered for rendering (Fig. 5). In rendering adaptation, the rendering parameters of the virtual objects are determined from two sources. The parameters that are context adaptable are queried from the context knowledge base, and their values are set according to the real time contexts (Fig. 6). The rest of the rendering parameters are queried from the authoring records stored in the II module (Sect. 3.1). Finally, a set of rendering units which consist of the file path of the filtered virtual objects and their rendering parameters will be generated (Fig. 9). The rendering units will be drawn in the AR environment using OpenGL.

4 System implementation and case study

A prototype system has been developed based on the methodologies presented in the previous section to provide context-aware information to the maintenance operators. The CC module and the ARV module are developed using C++ with Visual Studio 2005. The CM module has been developed using OWL API [33] and the Pellet reasoner [31] in Java, and it communicates with the other modules via socket communications. In addition, a mobile platform, comprising a Wrap 920 HMD equipped with a web camera, and a laptop, is constructed.

4.1 Case study

To validate the proposed concept, the adaptation schemes, the method for authoring context-aware AR contents and test the prototype context-aware AR maintenance system, a case study is conducted. The application scenario is the CNC milling machine in the manufacturing laboratory in the authors' institution, and context-aware AR information will be provided to the various users of the equipment, which are as follows:

- Instructions on the maintenance of the equipment will be provided to the engineers according to the task status and the engineers' expertise level;
- Alerts on the periodic preventive maintenance tasks will be provided to the equipment operators;
- Information on the equipment configuration will be provided to the trainees;
- The virtual objects will be rendered in the users' preferred colors.

The context ontology is constructed based on COARMS (Fig. 10). The person class has three subclasses, i.e., the three target users of this case study, namely, the engineer, operator and the trainee. They have the properties of ExpertiseLevel, PreferredColor, etc. There are three activities, namely, ReplaceMillingCutter, CleanWorkSpace and LearnEquipInfo. ReplaceMillingCutter is a corrective maintenance activity that requires the engineer to replace a broken cutting tool to restore equipment performance. CleanWorkSpace is a preventive maintenance activity that requires the operator to clean the workspace at 5pm daily. LearnEquipInfo is a training task that requires the maintenance trainee to understand the configuration of the milling machine. The location class has a subclass of MillingArea that represents the surroundings of the milling machine, and it has properties of name and time. In the CompEntity class, the MillingMachine is registered in Equipment, the Laptop of the mobile platform is registered in CompDevice, the camera and clock are registered in the SensorDevice, and a physical marker, kanji, is registered in Marker and attached on the CNC machine.

There are six information instances authored and to be provided to different operators involved in the various activities. ReplaceCutter_H is a brief instruction on the activity ReplaceMillingCutter, and ReplaceCutter_L1, ReplaceCutter_L2 and ReplaceCutter_L3 describe the instructions on the three sub-steps of the activity. AlertCleaning is a reminder to the operator on the activity CleanWorkSpace, and EquipInfo describes the equipment configuration to the trainee engaged in the activity LearnEquipInfo using texts, symbols and CAD models.

Several logical rules are defined for context reasoning and adaptation. For example, the Status of the activity Clean-

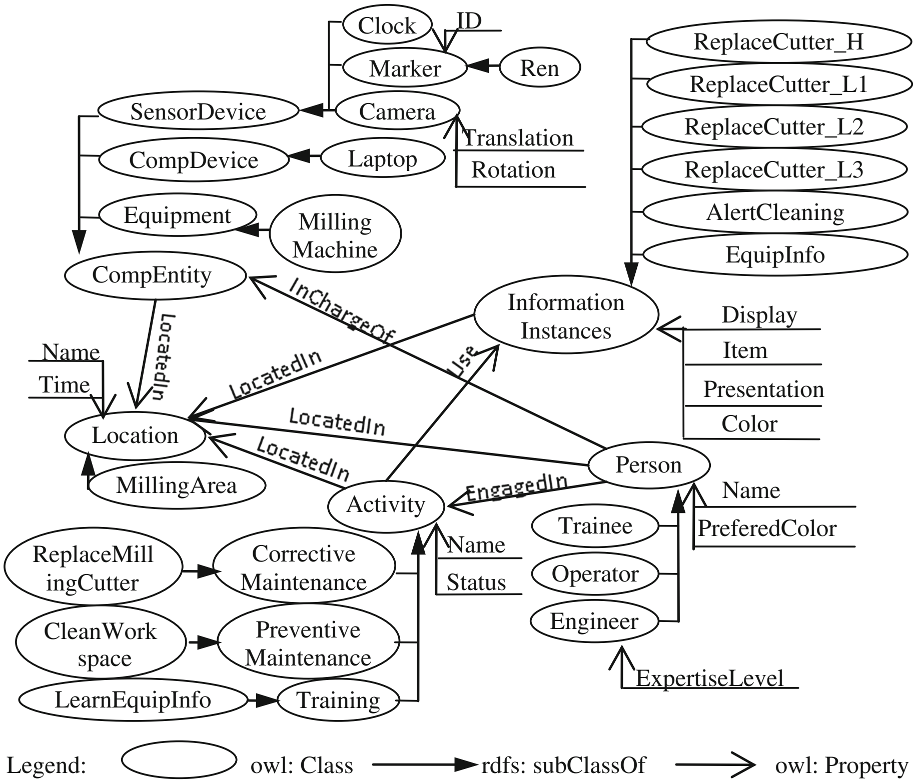


Fig. 10 Context ontology

(a)

```
PreventiveMaintenance (CleanWorkSpace) ^ LocatedIn
(CleanWorkSpace, ?a) ^ Time (?a, ?b) ^ swrlb:equal (?b,
xs:time("17:00:00")) -> Status (CleanWorkSpace, START)
```

(b)

```
Engineer (?a) ^ ExpertiseLevel (?a, High) ^ CorrectiveMaintenance
(ReplaceMillingCutter) ^ Status (ReplaceMillingCutter, START) ->
Display (ReplaceCutter_H, TRUE)
```

(c)

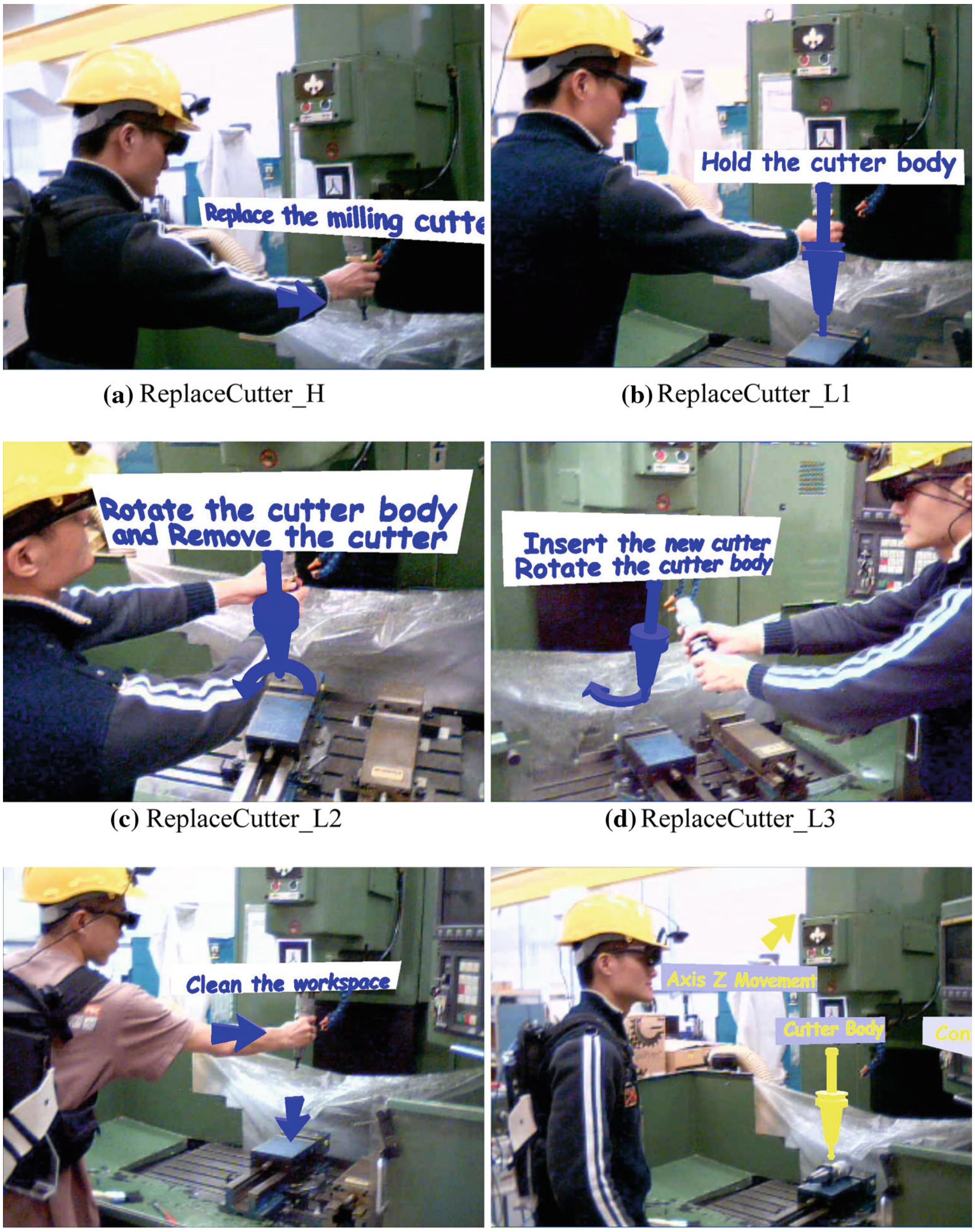
```
Person (?a) ^ PreferredColor (?a, ?b) ^ InformationInstances (?c)
Display (?c, TRUE)-> Color (?c, ?b)
```

Fig. 11 Logical rules

WorkSpace is derived from the local time obtained from the CPU timer of the mobile device (Fig. 11a), the information instance ReplaceCutter_H will be displayed when the engineer has a high level of expertise (Fig. 11b), and the color of the virtual objects will be configured according to the user's preference (Fig. 11c).

The user interface allows information of the contexts of ExpertiseLevel, PreferredColor, etc., to be obtained from the user inputs. The time property of the location class is obtained from the CPU timer of the mobile device, and the translation and rotation properties of the camera are determined through analyzing the captured images.

To validate the context adaptation of the authored information instances, on-site testing was conducted. The users carry the mobile platform and log into the system via the user interface, and different information will be rendered on the equipment. Brief maintenance instruction is provided to the experienced engineer to highlight the cutting tool and describe the operation using texts (Fig. 12a). Detailed instructions are provided to the engineer with low level of expertise (Fig. 12b–d). The reminder on cleaning the workspace is rendered to the operator to indicate the places to be cleaned when the task is due (Fig. 12e), and the information on the equipment configuration is displayed to the trainee (Fig. 12f).



(a) ReplaceCutter_H **(b)** ReplaceCutter_L1

(c) ReplaceCutter_L2 **(d)** ReplaceCutter_L3

(e) AlertCleaning **(f)** EquipInfo

Fig. 12 On-site testing

4.2 Discussion

The case study demonstrates the validity of the proposed concept, the adaptation scheme, as well as the low-level authoring method for context-aware AR contents. In addition, the developed prototype system can obtain and analyze the contexts, and adapt the authored information instances to the contexts correctly. Different information can be provided to the users according to their roles, levels of expertise, task status, preferred color, etc. Thus, the objectives of this research have been achieved.

However, a few issues that could be improved in future are identified from the case study. Firstly, the proposed authoring method requires programming skills, and domain experts without such knowledge are hindered in this process. Thus, the user interfaces that abstract the low-level programming skills will be developed in future so that more AR developers can participate in the authoring process to create context-aware AR contents easily. In addition, the proposed authoring method use physical markers as real world references to register the virtual objects. However, the on-site test conducted indicates that the virtual information jitters when the user moves further away from the marker. In future, using natural features in the environment for tracking will be explored.

5 Conclusion

The integration of context-awareness and the AR technology has been proposed to assist the maintenance personnel to perform maintenance tasks. The proposed system acquires and reasons the contexts, based on which, it renders context relevant information to the user. The feature of context-awareness can improve the usability of AR maintenance systems in routine and ad hoc maintenance activities. In addition, the concept and adaptation scheme of context-aware AR contents are explored, and a low-level authoring method is proposed.

A case study has been conducted to evaluate the proposed concepts and techniques. In the case study, various information instances are authored using the proposed approach and a prototype system is developed to validate the system performance. The case study demonstrates that the authored information instances can be adapted to the contexts correctly, and the system is able to provide context relevant AR services to the users. In future, several research issues will be explored. A user interface for authoring AR contents will be developed to abstract the low-level programming skills, so that users with no programming knowledge are able to participate in the authoring process. In addition, using natural features in the environment as real world references will be explored to replace the physical markers.

References

1. Azuma, R.: A survey of augmented reality. Presence Teleoperators Virtual Environ. **6**(4), 355–385 (1997)
2. Bottecchia, S., Cieutat, J.M., Merlo, C., Jessel, J.P.: A new AR interaction paradigm for collaborative teleassistance system: the POA. Int. J. Interact. Design Manuf. **3**(1), 35–40 (2009)
3. Morkos, B., Taiber, J., Summers, J., Mears, L., Fadel, G., Rilka, T.: Mobile devices within manufacturing environments: a BMW applicability study. Int. J. Interact. Design Manuf. **6**(2), 101–111 (2012)
4. Dey, A.K.: Understanding and using context. Pers. Ubiquitous Comput. **5**(1), 4–7 (2001)
5. Byun, H.E., Cheverst, K.: Utilizing context history to provide dynamic adaptations. Appl. Artif. Intell. **18**(6), 533–548 (2004)
6. Zhang, J.X., Sheng, Y.H., Hao, W., Wang, P.P., Tian, P., Miao, K., Pickering, C.K.: A context-aware framework supporting complex ubiquitous scenarios with augmented reality enabled. In: Proceedings of 5th International Conference on Pervasive Computing and Applications, Maribor, pp. 69–74 (2010)
7. Lewandowski, J., Arochena, H.E., Naguib, R.N.G., Chao, K.M.: A portable framework design to support user context aware augmented reality applications. In: Proceedings of 3rd International Conference on Games and Virtual Worlds for Serious Applications, Athens, pp. 144–147 (2011)
8. Shin, C., Kim, H., Kang, C., Jang, Y., Choi, A., Woo, W.: Unified context-aware augmented reality application framework for user-driven tour guides. In: Proceedings of 2010 International Symposium on Ubiquitous Virtual Reality, Gwangju, pp. 52–55 (2010)
9. Feiner, S., Macintyre, B., Seligmann, D.: Knowledge-based augmented reality. Commun. ACM **36**, 53–62 (1993)
10. Friedrich, W.: ARVIKA-augmented reality for development. Production and Service, International Symposium on Mixed and Augmented Reality, Darmstadt, , pp. 3–4 (2002)
11. Schwald, B., et al.: STARMATE: using augmented reality technology for computer guided maintenance of complex mechanical elements. In: Proceedings of eBusiness and eWork Conference (e2001), Venice, pp 17–19 (2001)
12. Savioja, P., Järvinen, P., Karhela, T., Siltanen, P., Woodward, C.: Developing a mobile, service-based augmented reality tool for modern maintenance work. In: Proceedings of the 2nd International Conference on Virtual Reality, Beijing, pp. 554–563 (2007)
13. ARTESAS: Advanced augmented reality technologies for industrial service applications. http://www.artesas.de (2012)
14. Harmo, P., Halme, A., Virekoski, P., Halinen, M., Pitkänen, H.: Etälä—virtual reality assisted telepresence system for remote maintenance. In: Proceedings of the 1st IFAC Conference on Mechatronic Systems, Darmstadt, pp. 1075–1080 (2000)
15. Didier, J.Y., Roussel, D.: AMRA: augmented reality assistance in train maintenance tasks. In: ISMAR'05: Workshop on Industrial Augmented Reality, pp. xvii–xviii (2005)
16. Neubert, J., Pretlove, J., Drummond, T.: Rapidly constructed appearance models for tracking in augmented reality applications. Mach. Vis. Appl. **23**(5), 843–856 (2012)
17. Alvarez, H., Aguinaga, I., Borro, D.: Providing guidance for maintenance operations using automatic markerless augmented reality system. IEEE International Symposium on Mixed and Augmented Reality, Basel, pp. 181–90 (2011)
18. Caponio, A., Hincapie, M., Gonzalez Mendivil, E.: lMAR: highly parallel architecture for markerless augmented reality in aircraft maintenance. International Conference on Virtual and Mixed Reality, Orlando, pp. 20–29 (2011)
19. Yan, W., Ishii, H., Shimoda, H., Izumi, M.: A feasible tracking method of augmented reality for supporting fieldwork of nuclear power plant, Lecture Notes in Computer Science, vol. 5622. LNCS, pp. 639–646 (2009)
20. Klein, G., Murray, D.: Parallel tracking and mapping for small ar workspaces. International Symposium on Mixed and Augmented Reality, Nara (2007)
21. Goose, S., Sudarsky, S., Zhang, X., Navab, N.: Speech-enabled augmented reality supporting mobile industrial maintenance. IEEE Pervasive Comput. **2**(1), 65–70 (2003)
22. Park, H.M., Lee, S.H., Choi, J.S.: Wearable augmented reality system using gaze interaction. 7th IEEE International Symposium on Mixed and Augmented Reality, Cambridge, pp. 175–176 (2008)
23. Yuan, M.L., Ong, S.K., Nee, A.Y.C.: The Virtual Interaction Panel: an easy control tool in augmented reality systems. Comput. Anim. Virtual Worlds J. (Special Issue: The Very Best Papers from CASA 2004 **15**(3–4):425–432 (2004)
24. Henderson, S.J., Feiner, S.: Opportunistic tangible user interfaces for augmented reality. IEEE Trans. Vis. Comput. Graphics **16**(1), 4–16 (2010)
25. Jacob, R.J.K.: Eye-movement-based human-computer interaction techniques: toward non-command interface, chap. 6. Advances in Human–Computer Interaction, vol. 4. Ablex Publishing Corporation, Norwood, pp. 151–190 (1993)
26. Sakata, N., Kurata, T., Kuzuoka, H.: Visual assist with a laser pointer and wearable display for remote collaboration. Trans. Virtual Real. Soc. Japan **11**(4), 561–568 (2006)
27. Ong, S.K., Yuan, M.L., Nee, A.Y.C.: Augmented reality applications in manufacturing: a survey. Int. J. Prod. Res. **46**(10), 2707–2742 (2008)

28. HITLabNZ ARToolkit: http://www.artoolkit.sourceforge.net/ (2005)
29. Wang, X.H., Zhang, D., Gu, T., Pung, H.H.: Ontology-based context modeling and reasoning using OWL. 2nd IEEE Annual conference on Pervasive Computing and Communications Workshops, Singapore, pp. 18–22 (2004)
30. SWRL: A semantic web rule language combining OWL and RuleML. http://www.w3.org/Submission/SWRL/ (retrieved 28 April 2012)
31. Pellet: OWL 2 Reasoner for Java. http://clarkparsia.com/pellet (retrieved 28 April 2012)
32. Shen, Y., Ong, S.K., Nee, A.Y.C.: Vision-based hand interaction in augmented reality environment. Int. J. Hum. Comput. Interact. **27**(6), 523–544 (2011)
33. The OWL API. http://owlapi.sourceforge.net/ (retrieved 28 April 2012)