Augmented Reality Visualization and Edition of Cognitive Workflow Capturing

Elizabeth Carvalho, Hugo Domingues and Gustavo Maçães
CCG - Center for Computer Graphics
Guimarães, Portugal

Luís Paulo Santos
Department of Informatics
University of Minho
Braga, Portugal

***Abstract* — The aim of the COGNITO project is to design a personal assistance system, in which Augmented Reality (AR) is used to support users in task solving and manipulation of objects. Due to its sensing and learning capability, the COGNITO system automatically creates workflow references by observing a shown task in learning mode. After the workflow has been learnt, the system can be run in playback mode, in which it explains the previously learnt task to the operator. The system compares the user activity in real-time with the workflow reference and provides adequate feedback. This system is composed by four main modules. This paper focuses on the last module – the 3D graphics engine – which is the basis to the development of both the augmented and the virtual reality player. Additionally, it also presents the template of actions editor which is an editing tool that enables non-programmers and non-3D-experts to prepare and accompany the composition of visualizations for end-users.**

Keywords: Artificial, augmented and virtual realities.

I. INTRODUCTION

The automatic capture, recognition and rendering of human sensory-motor activities constitute essential technologies in many diverse applications, ranging from 3D virtual manuals to training simulators and novel computer games. Although capture systems already exist on the market, they focus primarily on capturing raw motion data, matched to a coarse model of the human body. Moreover, the recorded data is organized as a single cinematic sequence, with little or no reference to the underlying task activity or workflow patterns exhibited by the human subject. The result is data that is difficult to use in all but the most straightforward applications, requiring extensive editing and user manipulation, especially when cognitive understanding of human action is a key concern, such as in virtual manuals or training simulators.

The aim of the COGNITO [1] project is to design a personal assistance system, in which Augmented Reality (AR) is used to support users in task solving and hand manipulation of objects. Due to its sensing and learning capability, the COGNITO system automatically creates workflow references by observing a shown task in learning mode. After the workflow has been learnt, the system can be run in playback mode, in which it explains the previously learnt task to the operator. The system compares the user activity in real-time with the workflow reference and provides adequate feedback.

Adaptive Augmented Reality represents the playback mode, in which the COGNITO system replays the previously learnt workflow to an inexperienced operator. The operator is equipped with the COGNITO system, which guides him through the task by means of interactive AR support. The system is adaptive, responding appropriately to uncertainty, difficulties and errors made by the operator, providing cognitive assistance in a timely and effective manner. The operator has the ability to pause, stop and restart the system.

Another important aspect is to have an edition tool that works as much as an assemblage environment for virtual manuals, having as basis the resulting data from monitored workflow. This tool should enable non-programmers and non-3D-experts to compose different AR visualizations for end-users.

Section 2 introduces briefly the scientific background in terms of augmented reality. Section 3 presents the architecture design, focusing on the graphics encoding, viewing and editing component of the project. Section 4 briefly resumes the present implementation state of this component, while section 5 summarizes main conclusions reached so far and future work.

II. SCIENTIFIC BACKGROUND

Recent work on Augmented Reality (AR) has made significant advances in both technology and algorithms [2]. New platforms have emerged in the form of lightweight head mounted displays and hand-held devices, equipped with miniature cameras and high resolution displays. Vision and inertial sensing enable robust camera tracking to facilitate stable content delivery. This includes marker based systems, robust model based tracking [3,4], combined vision-inertial systems [5] and more recent advances in visual SLAM – Simultaneous Localization and Mapping [6]. This progress has meant that AR is now at a stage where practical applications are a real possibility, particularly in user assistance systems such as that developed in the European project MATRIS [7].

In recent years the interest, as well as results, in research of AR technologies on Desktop environments, have increased. Several platforms have been developed with different

architectures; including AMIRE [8], ARVIKA [9], DWARF [10], DART [11]. Several software tools for development are also available, including NyARToolkit [12], StudierStube [13], SLARToolkit [14], ATOMIC [15], IrrAR [16] and OSGART[17] among others.

One of the main challenges is to obtain realistic rendered objects mixed with the real world. (figure 1). Because in the surrounding real world the light cannot be fully controlled or predicted, the synthetic model is rendered according to a lighting model that might not match the real one (light position, intensity, color, etc.). Other sensitive issue is the scale factor the synthetic model must be rendered according to the end-user's viewpoint position and the size of surrounding real objects. Occlusion of markers can cause very annoying effects and hardware processing capacity might impact severely in the overall system performance (image flickering and unsteadiness), especially when the synthetic object has a high degree of detail.

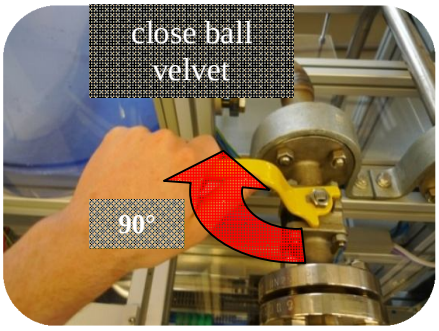

Figure 1. Example of assisted AR

Simultaneous Localization and Mapping (SLAM) is a popular map [18] building approach in autonomous mobile robotics. Because users demand faster and more effective algorithms, SLAM remains an active area of research. On the other hand, most work in visual augmented reality (AR) employs predefined markers [19] or models that simplify the algorithms needed for sensor positioning and augmentation but at the cost of imposing restrictions on the areas of operation and on interactivity. For a variety of augmented reality (AR) applications, the aim is to be able to use previously unseen physical objects as the basis for the augmentations. This demands accurate, robust and interactive systems that can position both the sensors and the scene with little prior knowledge. SLAM techniques can help a lot to handle or even solve this drawback.

There is evidence that the technology is beginning to be exploited in the market and to be accepted by end users, including industrial solutions such as AR-training and services, with major benefits being offered in terms of faster knowledge transfer and higher quality operations. However, as noted earlier, a major burden and cost factor is the creation and effective delivery of content such as assembly or repair instructions. Virtual and augmented reality instructions run in 3D and present the procedure in a continuous form. These animations are effectively created by hand and are often exported as movies, hence strongly limiting interaction and in particular making any updates or improvements impossible without completely remaking the application.

III. ARCHITECTURE DESIGN

The COGNITO system is complex and involves novel customized hardware with multiple sensors which will deliver a large amount of data at high speed. The software system will deal with streams of heterogeneous sensor data in real-time and moreover offer rendering capabilities in an augmented reality display. Last but not the least, the goal is to develop a mobile and wearable system that requires hardware miniaturization, good ergonomic and optimization of the processing load. Figure 2 illustrates the usage scenario while figure 3 presents the system overview.

Four main building blocks of the COGNITO system have been identified:

- On-Body Sensor Network (BSN) and Head-Mounted Display (HMD)
- Low-Level Sensor processing
- Workflow Recovery and Monitoring
- 3D Graphics Engine

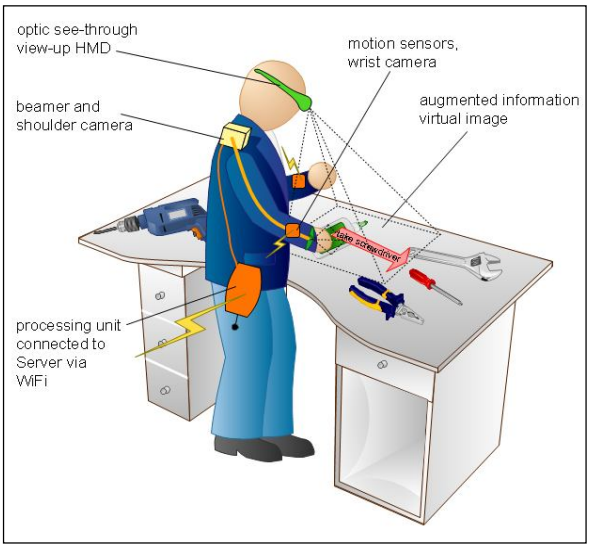

Figure 2. Wearable hardware elements

The BSN and the HMD are responsible for the inertial measurement units (IMUs), cameras, eye-tracking and miniature projectors are combined in a sensor network. A monocular head-mounted display (HMD) provides the system feed-back and user assistance information. The Low-Level Sensor processing handles the measurements from the BSN and provides estimates of the positions of the operator, his

hands, and relevant objects in the environment. The responsibility is shared between two modules: Sensor Fusion - responsible for the integration of the inertial sensors, and high level vision information; and Computer Vision - responsible for processing the raw image data from the cameras.

The Workflow Recovery and Monitoring receives a sequence of instantaneous configurations for the operator, his hands, objects and parts of objects contained within a work space from the Computer Vision. It processes this information and provides the marker (start and stop timestamp) for ongoing/future atomic events in the current workflow sequence. It also estimates the position of objects in the same workspace and provides feedback to the Computer Vision. Finally, the 3D Graphics Engine is used to produce the proper graphics for editing workflows as well as aiding the user during task execution using an augmented reality viewer.

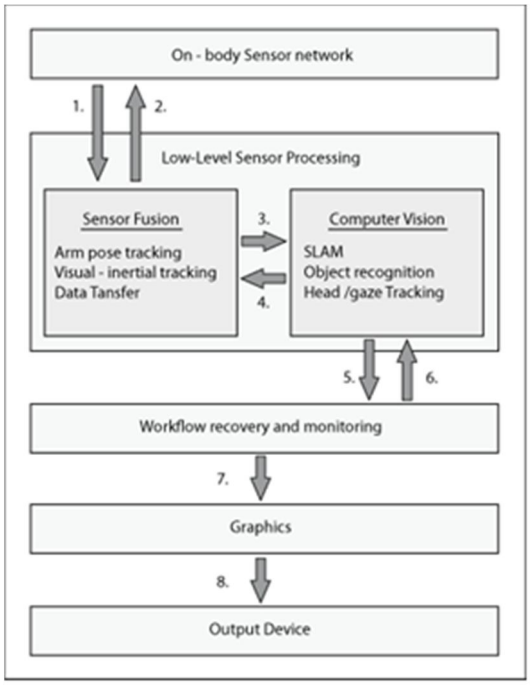

Figure 3. *System overview*

A. Graphics Encoding, Viewing and Editing

The COGNITO approach relies on well-defined templates of actions (graphics templates), which represent basic steps of the workflow. Those templates offer a minimal number of parameters, which are instantiated by the author of the graphics application. In COGNITO the same basic structures are used, but the workflow will be defined automatically as an activity graph, and each template will be instantiated using the captured sensor data. The user will have the possibility to preview and edit the results in order to correct, adapt or generate a novel workflow using the created basic elements. The workflow can then be presented in a VR or AR context either in a "step-by-step" mode or in an adaptive mode using the sensor network and recognizing in real-time the current user operation.

The Graphics Encoding, Viewing and Editing work-package is responsible for the design, development and implementation of the template of actions (TA) editor which includes a virtual reality (VR) player and the augmented reality (AR) player. The former is used on a scenario where interaction with the user is performed by conventional devices such as a display and a mouse and is used either as a verification tool while editing or as multimedia documentation of a given task. The latter requires multiple cameras and a head mounted display, is guided by user actions and based on the previously acquired workflow, involving sophisticated image preprocessing for high level information extraction, thus providing an immersive experience. Both players require reconstituting the previously acquired workflow and will be used on distinct interactive settings. The visualization will provide a true interactive scenario in the augmented reality case, because the user actions will be evaluated in real time by the previous modules. In the case of virtual reality, the user interaction will have a more restricted scope, thus being a VR version of an instruction manual.

Finally, the concept of TAs is assumed as being a formal description of atomic actions. The different combinations and sequences of these actions describe a workflow (that in turn describes how a certain task should be performed).

In resume, the main goal of this work package is to develop the required concepts, editors and visualizations components for:

- Automatic or semi-automatic composition of VR and AR visualizations, based on sensor and workflow monitoring input;
- An AR and VR system that visualizes this composition;
- A parameterizable AR and VR system that can be controlled by the end user or that reacts to the end-user behavior;
- Creation of a visualization system that enables non-programmers and non-3D-experts to prepare and accompany the composition of visualization for the end-user (e.g. mechanic).

B. The Editor

The database of action templates and their possible instantiations (filling of parameter slots) must be created with an editor that enables an immediate visualization of the instantiated template examples, to ensure an efficient editing process. Furthermore, the editor must allow manual adjustments (or changes) to the automatically recomposed visualization; other editor components will enable control of the playback and assistance session by end-users. Furthermore, it may allow defining the behavior of the system in case of deviations from the expected work plan, in the case of the exemplary AR-assistance system.

The editor will allow saving, deleting, previewing, copying and editing of the TAs. The edition will allow the insertion, deletion or edition of multimedia (sounds, video or images) and text elements that will help to support the training. These objects will be inserted in key positions along the TA. The previewing will allow the play of the template of action and its 3D visualization. Because the TAs are constructed having as basis Petri-nets graphs (inside the Workflow module), the editor will also link to an external petri-net editor, allowing specialized users to edit the workflow at this level.

Figure 4 illustrates the system usage pipeline.

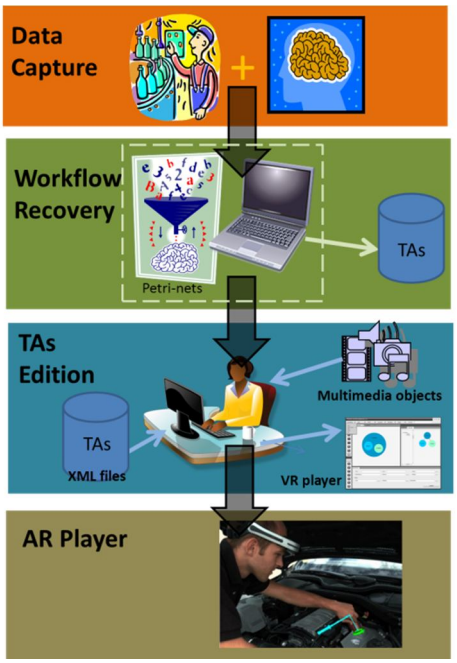

Figure 4. System usage pipeline

C. *The template of Actions*

The templates of actions that are going to be read and processed by the AR player and edited in the editor will attend a XML data schema. These schemas actually describe the characteristics of the classes of objects that are needed in order to render properly the data received from the WF module. It is considered that each workflow describes atomic actions. The proposed tags are briefly described below.

- <workflow> - It identifies the beginning and the end of a workflow. Inside of it, should be included one < scene> and one <activity > tags;

- <scene> - It identifies a group of elements that are needed in order to represent visually a task. Because this execution can usually involve several objects, this tag can hold multiple <element> tags;

- <element id=”identifier” name=”name” type=”type”> - The “name” and “identifier” are unique strings used to better identify and describe what the element holds. The “type” can be one of these: tool, avatar, part or subpart. Inside the <element> tags can be defined one <model>, one <info> and one <tool> tag, according to each scenario requirements. These are used to describe in detail what the element is composed by;

- <model filepath=”path” texture=”path” thumbnail=”path” > - This tag defines the details of the model which will be used to represent the element in 3D. The “filepath”, “texture” and “thumbnail” hold the path for the files that contain the mesh, texture and image thumbnail, respectively;

- <info> - is optional and holds all the complementary information that is going to be delivered to the user during the AR session. The following tags might be used: title, text, image, sound, video and link. Each of them might contain respectively the title, the image or video to be shown (in a part of the AR view), the hyperlink to be displayed or the texts to be written upon the AR visualization.

- <activity> - It can hold several Compound Event tags. The activity tag encompasses a group of event tags that are needed to describe the motion that represents the scene.

- <CE id=”identifier” name=”name” type=”type”> - In the Compound Event tag, the value of “identifier” and “name” are unique strings that help to identify the action itself. The value of “type” can be “error” (if the action is a corrective one) or “normal”. Each compound event contains one or more primitive events that each describe an atomic action;

- <PE id=”identifier” name=”name”> - In the Primitive Event tag, the value of “identifier” and “name” are unique strings that help to identify the action itself;

- <PEdata id=”identifier” name=”name” objects=”number of objects”> - The data relative to each primitive event is stored in data tags (one for each object), inside this one, the object attribute holds the number of objects involved in the event;

- <data id=”identifier” name=”name” total_samples=”total number of samples” visible=”boolean" opacity=”float”> - This tag contains the 3D positions and orientations for one object in the primitive event. The id and name attributes correspond to the object being animated and total_samples holds the number of positions recorded. The data should be stored in <x>, <y>, <z>, <theta_x>, <theta_y>, <theta_z> tags corresponding to the list of positions and orientations for each coordinate;

IV. CURRENT STATUS

Figure 5 illustrates a fragment of the XML that was developed and is being used as interface between the AR player module and the Workflow module.

```
<workflow>
    <scene>
        ...
        <element id='9' name='right_wrist' type='tool'>
          <model filepath ="media/models/rightHandW.obj" texture="media/models/skin75.png"
          thumbnail ="media/menu/ICO_righthand1.jpg"/>
          <info><image>media/menu/ICO_righthand.jpg</image></info>
        </element>
    </scene>

    <activity>
          <CE id="A" name="Pick and place screw baton" time="6500
            <PE id="a1" name="Lift screws baton" time="3000"/>
            <PE id="a2" name="Place screw baton within marked region" time="3000"/>
            <PE id="a3" name="Release screw baton" time="500"/>
          </CE>
          <CE id="B" name="Pick screw driver" time="3000">
            <PE id="b1" name="Lift Screwdriver" time="3000"/>
          </CE>
          ...
    </activity>
</workflow>
```

Figure 5. XML Scheme

Both the Editor and the AR player are being implemented for Linux. QT and Irrlicht are being used for development. QT is an open source framework under LGPL license for interface development and is being used to create the editor user’s interface. Irrlicht is also an open source graphics engine and is being used for the implementation of the AR and VR visualizations (in the AR player and editor). C++ is being used as the programming language. The editor is still in its early stages, while the AR player is almost completed.

Figure 6 illustrates the AR player interface. While running, the objects used in the scene are shown in the left panel, by means of icons that become active according to their use in the current activity. At the top of the window written messages are also being delivered according to what is specified in the TA. In this window, multimedia objects can also be displayed besides the 3D representation of tools and other relevant objects involved in the action.

Figure 6: VR/AR player

Figure 7 shows the overall design of the editor that is being implemented. The editor will offer a 3D graph visualization of the XML files (TA files), that will be used as a widget for interaction with the user. The editor will allow the visualization of videos, VR animation of the recorded workflow, besides the intuitive and guided edition of TAs

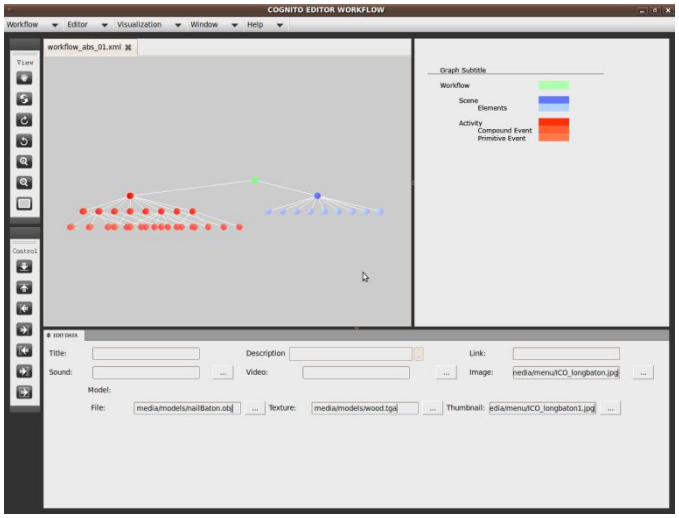

Figure 7. Workflow editor overall design

V. CONCLUSIONS AND FUTURE WORK

An experienced user will carry out an assembly task multiple times while wearing the COGNITO system. Actions and workflow are captured and learnt, and then subsequently rendered within a virtual manual application using the editor. It will also offer adaptive augmented reality visualization where an inexperienced operator, equipped with the COGNITO system, will be guided through the task based on previously learnt workflow. The system will be adaptive, responding appropriately to uncertainty, difficulties and errors made by the operative, providing cognitive assistance in a timely and effective manner.

This paper focused on the 3D graphics engine of the COGNITO system. This includes a workflow editor and two players: the VR and the AR players. Additionally, the XML schema proposed to represent templates of actions was presented.

Both the editor and the AR player were designed and are being implemented with a strong focus in usability that will have to be thoroughly tested. In the case of the editor, an interesting enhancement is to allow the end-user to have a fast and clear knowledge and notion of the contents of a TA. Presently, this content can be played in a VR window, but it is not possible to have any clue about the statistics of the atomic actions that are involved (i. e. how many times a screw was screwed or what action is more complex to execute). Further work involves the exploration of this topic with information visualization techniques.

REFERENCES

[1] COGNITO Project, FP7, ICT-2009.2.1 Cognitive Systems and Robotics, ICT – Information and Communications Technologies, number ICT–24829, http://www.ict-cognito.org/index.html (visited on 5-1-2011).

[2] Veronica Teichrieb, João Paulo Lima, Eduardo L. Apolinário, Thiago S. Farias, Márcio A. Bueno, Judith Kelner and Ismael H. Santos, "A Survey of Online Monocular Markeless Augmented Reality", International Journal of Modeling and Simulation for the Petroleum Industry, Vol. 1, no. 1, pp. 1-7, August 2007.

[3] Suya Vou Neumann, U. Azuma, R, "Hybrid inertial and vision tracking for augmented reality registration", Proceedings of Virtual Reality, 1999, Houston, Texas, USA, ISBN 0-7695-0093-5, pp. 260-267, 13-17 March 1999.

[4] Drummond, T. W. and Cipolla, R., "Real-time visual tracking of complex structures", IEEE Transactions on Pattern Analysis and Machine Intelligence, ISSN 0162-8828, Vol. 24, no. 7, pp. 932-946, July 2002.

[5] Mark Pupilli, Andrew Calway, "Real-time Camera Tracking Using Known 3D Models and a Particle Filter". 18th International Conference on Pattern Recognition, Hong Kong, ISBN 0-7695.2521-0, pp. 199-203, August 2006.

[6] Andrew J. Davison, Ian D. Reid, Nicholas D Molton, Olivier Stasse, "MonoSLAM: Real-Time Single Camera SLAM", IEEE Transactions on Pattern Analysis and Machine Intelligence, ISSN 0162-8828, Vol. 29, No. 6, pp. 1052-1067, April 2007.

[7] MATRIS – Markerless Tracking for Real-Time Augmented Reality Image Synthesis, IST- 002013, http://www.ist-matris.org/ (visited on 5-11-2011)

[8] AMIRE Project (available at http//www.amire.net, visited on 5-1-2011).

[9] ARVIKA Project (available at http://www.arvika.de, visited on 5-1-2011).

[10] DWARF Project (available at http://ar.in.tum.de/Chair/ProjectDwarf, visited on 5-1-11)

[11] DART (available at http://www.cc.gatech.edu/projects/ael/projects/dart.html , visited on 5-1-2010).

[12] NyARToolkit (available at http://nyatla.jp/nyartoolkit/wiki/index.php?FrontPage.en, visited on 5-1-2011).

[13] StudierStube Project (available at http://studierstube.icg.tu-graz.ac.at/, visited on 5-1-2011).

[14] SLARtoolkit (available at http://slartoolkit.codeplex.com/, visited on 5-1-2011).

[15] ATOMIC (available at http://www.sologicolibre.org/projects/atomic/en/, visited on 5-1-2011).

[16] IrrAR (available at http://www.irrlicht3d.org/pivot/entry.php?id=814, visited on 5-1-2011).

[17] OSGART (available at http://www.osgart.org/wiki/Main_Page, visited on 5-1-2011).

[18] Alex Kozlov, Bruce Macdonald, Burkhard Wünsche, "Towards Improving SLAM Algorithm Development using Augmented Reality", Proceedings of Australasian Conference on Robotics and Automation 2007, Brisbane, Australia, ISBN 978-0-9587583-9-0, 10-12 December, 2007.

[19] Denis Chekhlov, Andrew P. Gee, Andrew Calway and Walterio Mayol-Cuevas, "Ninja on a Plane: Automatic Discovery of Physical Planes for Augmented Reality Using Visual SLAM", Proceedings of International Symposium on Mixed and Augmented Reality (ISMAR), Nara, Japan, ISBN 978-1-4244-1749-0, pp. 153-156, 13-16 November 2007.