

Augmented and Mixed Reality Features and Tools for Remote Laboratory Experiments

<http://dx.doi.org/10.3991/ijoe.v12i07.5851>

Mark Smith, Ananda Maiti, Andrew D Maxwell, Alexander A Kist
University of Southern Queensland, Toowoomba, Australia

Abstract— Augmented Reality (AR) is the process of overlaying meaningful interactive information in a live video stream for creating an enriched visual experience for users. Within Remote Access Laboratories (RAL) this enables users to gain design experience along with gaining knowledge about the particular experiment in question and potentially collaborate on design experiences. This paper focuses on the issues related to the applications of AR in RAL, the levels of AR in context of RAL and their effect on the learning tools. This paper also discusses the challenges of integrating a Natural User interface into the AR for RAL experiments. Finally it presents two example applications for AR in RAL experiment - Virtual Objects Creation and Object Identification and Tagging.

Index terms—augmented reality; remote laboratories;; digital image processing; object tracking; computer vision

I. INTRODUCTION

Remote Access Laboratories (RAL) are online environments that allow users to communicate with remote measurement and scientific instruments over the internet. The users employ a User Interface (UI) that provides information regarding the experiment's inputs and outputs. The UI is sometimes referred to as a client. For an effective learning experience, UIs must be well designed. Several technologies have been used to create appropriate UI for RAL including specialized versions for desktops [1] and mobiles [2]. This means the experiment makers must create software tools to integrate special features like Augmented Reality into the UI for the experiment they design[3].

Augmented Reality (AR) is broadly a technology to enhance users visual and auditory perception by providing information for which the users senses cannot directly detect [4]. Mostly, AR embeds additional information in video/audio streams to create a richer interactive User Interface (UI). The embedded AR components can comprise of simple text or highly complex graphics and visuals. AR technology typically reacts to the changing surrounding environment i.e. responds with AR components depending upon the visual or tracked inputs into the system. The AR components typically need to update in real time by recognizing the input video frames' contents, processing it according to a pre-determined logic, and then produce the AR response components. Several other factors and input components within the target environment such as image tags, room telemetry, internal GPS, inertial, magnetic tracking, etc. are can also collected and analyzed to determine the AR output to produce an immersive experience for the users. Apart from objects, gestures from the users themselves which are also part of the environment. AR systems can

also take in other positional, inertial, or tracked inputs, in addition to conventional inputs such as a mouse and keyboard. AR is used in many areas of science and technology including computer games for recreational purposes, sports and entertainment, navigations and tourism. AR has also been used in education [5].

In this paper, the key aspects of integrating AR into RAL are discussed. This includes practical scenarios of augmented and virtual objects contents in the final video feedback utilising web-based technology. It also includes the AR framework and issues with implementing natural user interfaces for a RAL system and its vital components. Finally an example implementation in a web-based environment is discussed.

The rest of the paper is organized as follows: Section 2 discusses the current status of AR and it's applications in education and in particular RAL. Section 3 describes the application areas of AR in RAL, and types of AR and constraints of applying them to RAL systems. The key aspects of AR and RAL are presented in Section 4. Section 5 discusses the implementation methods of the tools described.

II. RELATED WORK

A. Augmented Reality

Augmented Reality can be perceived in multiple varied manners. Most commonly it is defined as a mixed environment that blends both digital information with real world objects in a meaningful and structured way [6]. Consequently the digital information should augment and support the viewable real world objects, where the amount of real world objects in the environment should be more than the digital overlay information for it to be considered AR [7]. AR has become widely popular and well known with gaming apps such as Pokémon Go¹ which was recently released. This emboldens the popularity of such technology and their acceptance and integration into traditional human activities.

There are different classes of AR environment based on how *immersive* they appear. One common form of device is a head mounted display system, possibly with hand gloves incorporating feedback [8]. These are fully immersive environments that enable users to experience the whole of the reality environment with augmented features. It also allows more accurate interaction with the AR environment. Fully immersive AR is achieved by using a wearable device such as smart glasses or head mounted displays. These devices have cameras mounted on them which are capable of running applications to process the video view within the visual area of the

¹ <http://www.pokemongo.com/en-au/>

wearer. This view is then enhanced with overlaid information.

The other type is desktop based AR [9] which only considers partial spaces around the user, in particular possibly being confined to the desktop computer screen. In this instance, the view is limited and interaction with the environment occurs through regular input devices e.g. mouse, keyboard, webCamera, etc. While full immersive AR is more attractive and advantageous, they have many problems:

- Expensive: The hardware required is expensive, and is not yet a commonly available tool available for broad educational purposes.
- Requires high precision to recreate the augmented feature. It requires expertise to setup and maintain the system, and is prone to tracking errors [10].
- It works by augmenting the local visual environment with digital virtual objects. This by itself takes considerable cost, processing power and technology. It is more difficult and unnecessary to recreate a remote real environment completely and then augment it with virtual objects. The ability to view remote real environment is much reduced and only a fixed set of views are available through cameras. Thus these video streams can be projected directly onto screens. However, a Virtual Reality (VR) environment can be used to provide an alternative interface where the users can have a immersive experience in a virtual world with some real components.

B. AR in Education

AR in education broadly aims to provide a rich educational experience. Such systems are usually implemented using desktop AR models or mobile devices due to their ready availability. Traditionally, some systems use visual markers for identifying a fixed position in the real world video stream, which is to be replaced with augmented information or visual objects. The augmented objects are typically stored in a database, against a unique identifier and reproduced when a desired marker appears within the captured stream. This also requires accurate computer vision models to correctly identify the marker and the encoded identifier within it.

This type of technology helps in understanding the functionality of lesson material through the use of enhanced models of the objects that replace the in-place real world learning materials. They can present the users with a quick in-depth augmented multimedia experience during their interaction with the real world environment.

C. RAL and AR

Augmented Reality features have been added to RAL experiments before [5, 7, 11]. Usually the AR is desktop based where the augmentation comprises of overlaid virtual components, such as switches, that can be manipulated by the user. In [7] the virtual elements of field-programmable gate array (FPGA) boards allow the user to remotely interact with the real and virtual devices. The real devices are viewed through a camera providing video feedback. The virtual components are a few 3D buttons overlaid over the real buttons in the video feedback. This approach provided a very realistic environment, as a majority of what the users see as part of the user interface was the real object i.e. the FPGA

Board, enabling to view all the changes and events clearly. Only small portions of the video feedback was overlaid with other information and graphics—that takes users inputs.

Within the context of web environments for RAL, desktop AR provides an opportunity for wider use and access to AR experiments, as this approach minimises the need for specialised hardware or software at the users end. Our peer-to-peer experiment [12] utilizing the SNAP [13] platform demonstrates the potential capabilities and usability of AR within a RAL framework. Other current JavaScript implementations provide options for utilizing WEB based implementations. For better and more engaging learning experience, special input devices can be used to take more complex inputs.

D. Natural User Interface

Natural User Interfaces (NUI) are a mechanism to take input from users without using a fixed position (i.e. immobile) or dimensionally restricted input devices (e.g. keyboard) which is restricted in movement and only allows basic text or button inputs or a mouse which allows only limited degrees of freedom. The NUI aims to take inputs from the natural movement of the users, particular gestures of their body parts. Typically an NUI aims to incorporate computer vision processing to identify and track whole body, or hand and finger movements as gestures for input to the system.

NUIs utilising 2D Cameras have been investigated for a long time, but with recent advances incorporating depth sensors (e.g. Kinect [14]) and other devices such Leap Motion [15], it has become easier to capture complex body gestures with high accuracy. NUIs can be easily integrated with AR (or VR) for inputs such that the user does not have to rely on conventional means for input. They can directly interact with the objects in the AR. NUI is very common in the computer game industry and has helped deliver very cost effective and commonly available solutions [16, 17].

III. AUGMENTED REALITY AND RAL

This section describes the application areas of AR in RAL, the types of AR, and constraints in applying them.

A. AR Application Areas in RAL

Augmented Reality (AR) can be used in many ways [6]. AR in RAL can be used primarily for two purposes:

- *Induced visibility*: In certain experiments some objects/entities/properties may not be visible to the camera. For example, magnetic fields that attract magnetic materials generated and studied by using different electromagnets [18]. These entities which are part of the experiment, may be embedded into correct positions by using animations. This involves re-drawing certain objects such as arrows over the region to indicate the presence and orientation of the entities.

- *Overlaid Information*: Another set of objects that needs to be presented is text information relating to certain real objects in the video. It is best to draw the text onto the video feedback close to the associated object. To do this however, the objects must be identified and tracked in real-time during the experiment. Overlaid text information must be updated in real time as well to reflect the change in the state of the object.

B. Levels of AR

In the current context, AR is the process of overlaying virtual objects including scalar images or vector animations or both onto the video feedback. The video feedback has a definite frame rate and resolution and thus a fixed number of pixels (P) for each frame. Milgram [19] shows a scale of virtuality, such as 2, where there is a continuous spectrum from reality (on the left) and a fully virtualised environment on the right. For AR, a pixel p in the feedback visuals may contain a real-object, a virtual object or fractionally both. Thus two measurements within a frame can be defined -

1. Virtual Pixels (P_V): the average number of pixels that relate to a virtual object. Then, ΔP_V and ΔR_V are two parameters that signifies average change in the P_V and R_V over time in the video feedback where R_V is the matrix representing the position of the virtual pixels.
2. Real Pixels (P_r): the number of pixels that relate to real objects. ΔP_r and ΔR_r signifies average change in the P_r and R_r over time in the video feedback where R_r is the matrix representing the position of the real pixels.

This allows for different degrees of virtual and real objects to be blended in the virtual continuum. In RAL, the AR may be implemented by having

Case 1. More virtual components with more active behaviour than that of the real objects. i.e.

$$P_V > P_r \text{ and } (\Delta P_V)(\Delta R_V) > (\Delta P_r)(\Delta R_r)$$

Case 2. More virtual components than the real objects but less activity in behaviour than the real objects. i.e.

$$P_V > P_r \text{ but } (\Delta P_V)(\Delta R_V) < (\Delta P_r)(\Delta R_r)$$

Case 3. Lesser virtual components than the real objects but more or equal active in behaviour than the real objects.

$$P_V \leq P_r \text{ but } (\Delta P_V)(\Delta R_V) \geq (\Delta P_r)(\Delta R_r)$$

Case 4. Lesser virtual components than real objects. i.e.

$$P_V < P_r \text{ and } (\Delta P_V)(\Delta R_V) < (\Delta P_r)(\Delta R_r)$$

The first case is in the space of augmented virtuality [20] where both virtual visibility and associated information are high and the real objects themselves do not change their orientation much. In the second case real objects change their orientation more often compared to (or equally to) the induced visibility/information. In the third case the user have less virtual components and the real world objects, both can be equally active. In the fourth case, the users' interact largely with the real components and only supporting information is displayed as visible information.

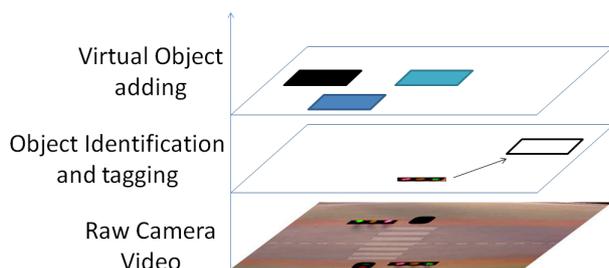


Fig 1. Video streaming combining live video, object detection and image overlav

IV. AR FRAMEWORK FOR RAL

This section presents the key aspects of the AR framework for RAL

A. Key Aspects of an AR System

Augmented Reality systems require several core components in order to provide the augmented visual services. The minimum set of services may vary from installation to installation, but for *visual* AR systems the set can be broadly defined as:

- *Video Capture of Live Video Streams*: real-time video stream interception is an essential component of all AR systems. Without proper processing of the video stream, numerous errors become introduced into the resultant feedback. Azuma [4] has detailed the primary difficulties, including many hardware and software induced issues. The capture of live video streams requires timely and accurate processing to avoid as many introduced errors as possible. Computer vision systems undertake analysis of the individual frames, either as discrete instance or as sequence.
- *Real Object Identification (ROI)*: Without the ability to interpret the scene from a video stream, AR capabilities become ineffectual. Interpretation of objects within the video scene comprises the bulk of the image processing workload. Computer vision techniques, such as probability density functions [21], frame subtract and clustering, are used to extract meaningful information from the sequence of live video frames. Object identification hinges on techniques that filter all but the items of interest. Sensing moving or changing scene objects through algorithms such as frame subtraction, fail to appreciate the complexity in the image parameters between one frame and the next. Single point solutions only solve problems for specific experiment configurations.
- *Object Tracking*: Identifying an object is only part of the AR key aspects, and identification must work with object tracking. Without the ability to track the identified object, recognize it if it should move in and out of the scene, or adjust its visible surface area, then the results are also ineffectual. Object tracking must continue to recognize the object and changes its position or orientation between frames.
- *Image Overlays*: Visual AR systems provide the augmentation through image overlays, combining the real and virtual dataset into one seamless visual information stream. Overlay data services need to ensure that important objects with the video stream are not occluded by any virtual objects draw to the frames. Occlusion affects both the real and virtual data, and strict rules are necessary to keep tracked objects, virtual feedback elements and secondary data objects clear from each other. Fig 2 shows an example of separating different data sets into layers, aiding the AR services to provide the correct data at the correct location.
- *Interactive Feedback*: In [4], the key aspect of AR is the interactive nature of the technology. Feedback to the user, in visual AR systems, requires images or text to be placed over the existing live video. Image

overlays form a source of feedback, but also in a form that allows interaction.

B. Key Aspects of an AR System in RAL

Integrating AR services into RAL requires system interfaces allowing the key aspects of AR to interact seamlessly. Fig 4 demonstrates the level of cohesion necessary for this to occur. Specifically how AR models interface with RAL is not within the scope of this paper, but the key interfaces, and their crucial requirements, are defined below:

- **Interfaces:** Remote Access Laboratories are both sources and sinks of data. The live video stream, the key aspect of RAL systems, is served to the AR services. This is the primary interface between the two systems, and requires careful management, as described further below. RAL user platforms, such as SNAP, require no modification to use the video stream, whereas more complex visual processing is required for three dimensional content of engineering and science subjects [9].
- **Timings/Delays:** The discrepancy between the live video data appearing from the RAL experimental rig, and the image overlays supplied by the AR services, are finite and influence the users immersion in the AR experience. Any delays introduced by the AR services must be kept to a minimum so as to maintain the AR synchronisation.
- **Registration:** Understanding a video scene is the purveyance of the AR services, but any virtual objects or images created by the AR services, must align with the real objects within the stream [22]. Delays (above) will produce obvious discrepancies between the real and virtual data. Misaligned virtual objects can appear in the wrong location within the users view.

C. NUI and gestures

Apart from real object tracking and virtual object creations, NUIs can be an integral part of RAL. Many experiments require operating instruments without using a keyboard type interface with buttons only. This involves complex hand movements to manipulate physical objects and observe the resultant behaviour of the objects. While implementing such experiments, the hand movements are replaced with some form of automations and the users never get the real hand-on-experience as they would have in an on-site laboratory. The NUI has the potential to address this issue. The issues that need to be considered to implement NUI and AR in an experiment for RAL are as follows:

- **Space Requirements:** NUI based experiments require a limited amount of space to operate. The type of the

experiment will determine what sort of space is required for movement. But most experiments require hand based gestures only, thus requiring a table type surface for creating the virtual objects.

- **Suitable Gestures:** NUI requires a suitable gesture library for each experiment that defines what can be recognized [14, 23] as a valid set of movements with respect to the experiment.
- **Gestural Control:** A suitable devices needs to be chosen for capturing NUI data. This could be wearable devices e.g. Sixsense Hero, Razer Hydra [24], remote recognitions devices e.g. Leap Motion [15] for hand gestures or Microsoft Kinect for full or upper body gestures [14].
- **Types of Display:** The output of the gestures along with the graphical representations of the gestures can be displayed either on a wearable display device e.g. smart glass or on a computer desktop as well.
- **Virtual objects:** As with any AR application, the NUI also requires virtual objects which are manipulated according to the users' input. These objects can be stored locally or downloaded during runtime.

Integrating the NUI with the AR involves the following issues:

- The virtual objects may be stored locally or downloaded and the user interacts with them locally on their computing devices. If a certain gesture (for e.g. turning on a switch) requires actions on the remote experiment rig, then the UI sends the request to the rig and waits for a reply. Once a response is received, the corresponding changes are reflected in the video output. This method is effective if there are distinct short gestures in term of time in the experiment e.g. flipping a switch or pushing a button. Longer gestures may require larger number of request/response between the NUI and the rig causing delays in updating the NUI and the virtual objects. The user may not be able to hold the hand or body positions for large intervals while waiting for a response to be received, thus creating a jittery learning experience.
- The virtual object may be operated remotely where the users' inputs from the NUI are constantly streamed to the remote experiments rig. The rig then determines any gesture and makes changes to the real rig and the virtual components as well. The status of the virtual rig and real instrument are constantly streamed back to the NUI. This type of remote NUI can be used to record and process long gestures as the gestures are streamed to the remote location.

V. IMPLEMENTATIONS

This section discusses the various methods of implementing AR tools within a RAL framework. Generic aspects of implementation have been discussed in the last section and demonstrated here in context of the Peer-to-Peer RALfie system [3]. It uses the SNAP platform [3] with some added AR functionalities. The SNAP is a web-based programming platform used for creating experiments programs. It is based on JavaScript and runs within a web browser. It is capable of transmitting commands to and consuming incoming data from an experiment. The SNAP enables smooth transition from

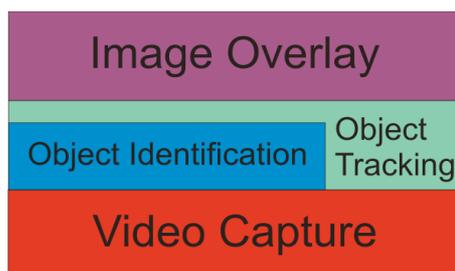


Fig 2. Augmented Reality visual interface model

the experiment makers interface to the learners interface with respect to a Peer-to-Peer RAL [3].

A. Video Capture

The first stage of implementing visual sensory AR services, is the processing of the live video stream from the RAL cameras. Typically, video processing techniques for non-AR applications, are performed off-line as some sort of post-processing procedure. Key aspects of AR dictate that the system must be interactive; as such, any sort of post-processing is automatically precluded. Ignoring *Case 1* as a form of AR for live experiences of remote labs, the remaining cases require the superimposing of virtual overlays.

Live video streams from RAL assemblies are broken down to individual frames for processing by other AR services. Tracking the frame numbers is necessary to ensure that video overlays are matched to the frame that they are derived from.

B. Object Detection and Tracking

ROI occurs through Computer Vision (CV) object detection methods. All computer vision models are based on the assumption that something significant changes between one frame and the next, or from one frame to the reference point. Expanding the CV models to interface to RAL systems is not a straightforward process, as most CV algorithms require either extensive training (probabilistic models require historical data [25] for baseline purposes), or processing periods longer than standard video frame rates.

Data associated with objects within the video stream are required to match the action appearing to the user, but also to the results of the ROI services. Sensor data plays an important role in producing the immersive interactive AR experience, and must coordinate with ROI, tracking and feedback services.

C. Object Identification and Tagging (OIT)

The approach to object identification is varied due to the methods utilized to extract the scene data. Successfully identifying an object in one frame does not necessarily ensure the object will appear in the next from in the same way.

Due to variations in lighting, subtle changes in camera image translation and other factors, a group of pixels, consisting of a shape (S) and volume (V) will change from frame-to-frame. Taking colour variation in isolation, for example; Fig 3 shows an example of the colour variation found in pendulum experimental rig, shown in Fig 6. The extent of a single pixels RGB colour variation over 214 frames, is substantial. The graph demonstrates the variation in the red and green colour channels for a pixel deemed to be a static background pixel.

For motion static rigs, such as the traffic light experiment shown below in Fig 4, object identification is simpler, as the location for the object detection is also static. When the object is detected, it has to be the object previously detected.

Once an object has been detected, the experiment maker must select the object as an item of interest. Several objects may be detected, but not everything is important to the experiment. Selecting the object *tags* the

item, supplying a unique identifier which can be associated with a sensor, for further use by the AR/RAL system.

In any rig, the actuators move causing a change in the rig's position. The magnitude of the change in the position can be measured by a sensor. For e.g. in Fig 6 (b) showing a pendulum experiment, if the user has to drop the ball, then the corresponding actuator is rotated by a certain degree. This ball then changes position in the video stream. The length of the drop is a function of the rotation. This change in the balls position can be display in real time as augmented texts pointing towards the ball. The aim of the experiment is to find the *acceleration due to gravity* by swinging the ball at different heights.

The experiment maker can attach the sensor values x as a function $f(x)$ which is constantly updated on the screen. The experiment maker can also designate a particular area of the screen where the text is displayed. This should ideally be a space that does not have any meaningful object and the text should not overlap such objects. However, in certain cases where, the rig has massive change in position, no such suitable space on the screen may be available for the entire duration of the experiment. The AR engine must determine a suitable space to put the text. The user is also able to switch on and off the AR components to make them visible or invisible.

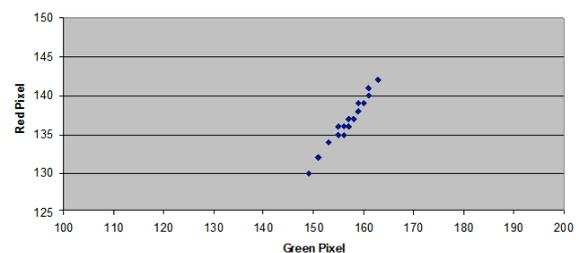


Fig 3. Red and Green colour variation of pixel (190,51) for pendulum experimental rig

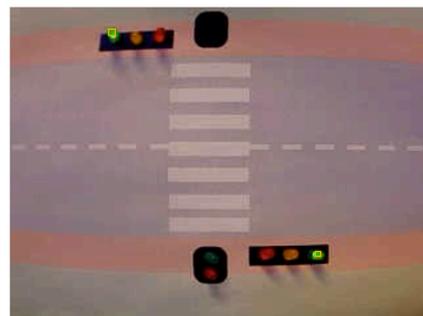


Fig 4. Traffic light experiment, with green traffic light detected by AR services

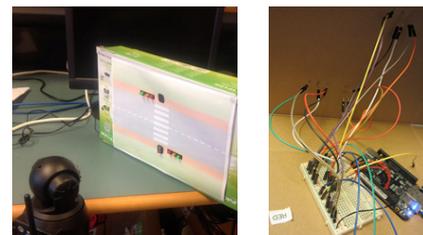


Fig 5. A P2P RAL experiment using everyday objects and MCU

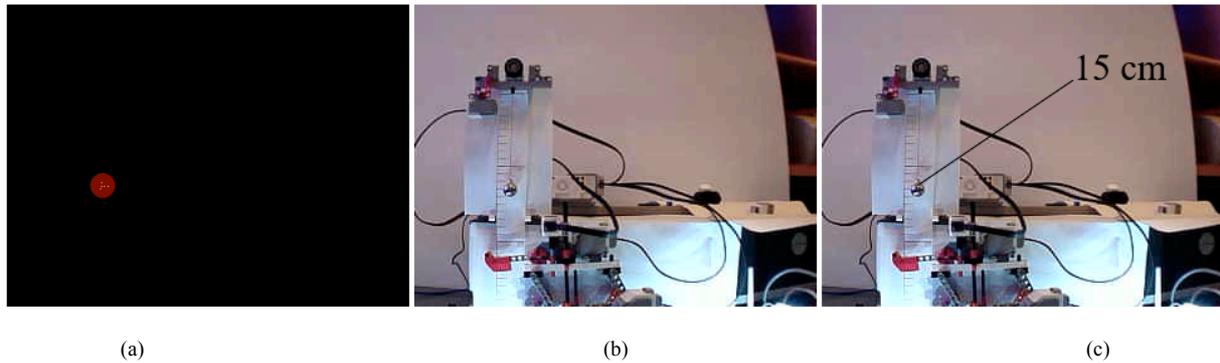


Fig 6. The pendulum Experiment. (a) The difference in frames to identify the moving object (i.e. the ball) (b) The original video feedback of the pendulum experiment (c) The final video feedback with the sensor value as shown to users.

A prototype ROI mechanism has been developed in RAL [12] as follows:

Step1. The initial image of a video stream, when a session starts, is stored as the background image (B).

Step2. Once any object moves in image F_i , it is isolated by subtracting $F_i' = F_i - B$. A residue of the object is left in B, which is identified using subsequent frames, as the residue will always remain static. The pixels of the object residue in B are replaced with the corresponding pixels in the current frame F_i .

Step3. A clustering mechanism is used to remove noise and get the actual objects in the frame F_i' . The clustering mechanism takes into account the potential radius (as specified by the experiment maker) of the target object that needs to be tagged. Thus any object that is larger than the size is automatically put off the list. Fig 6 (a) depicts an identified object.

These steps allow identifying and extracting an object from the video feedback. Once the object has been identified:

Step4. The experiment makers can then select the object(s) that need to be stored permanently and will be used for AR.

Step5. The experiment maker then associate a sensor value with the desired object and also mention its x, y coordinates on the stage.

Step6. Each object is stored in a database and marked with a unique identifier. The object does not need to be identified in real world as what it actually is, but only matched relatively in each experiment session.

It may be noted that the actual algorithms to realize each part can be implemented in multiple ways. For example, the DBSCAN algorithm [26] is used for creating clusters of right size.

While the steps 1-6 are designed to work as a generic tool to identify objects, they can be modified and used to identify objects specific to an experiment by the its makers in a centralized RAL.

When the users run the experiment, the AR module checks if any of the objects, stored in the database while creating the experiment is in the frame. If there is any such object, then the corresponding, sensor values are shown if AR tools are activated. Fig 6 (c) shows the final output of the Object Identification and Tagging (OIT) process where the ball is tagged with the value of the sensor measuring its height. This image frame is placed in the stage of the SNAP environment.

D. Super-Imposing the Camera View

Methods of augmenting the live video stream received from the RAL rig depends on the user platform. Correctly combining computer generated images with the live video stream, where the virtual objects react in synchronisation with the real objects, is a complex process.

The final output from the AR video services needs to be a new composite video stream, containing the combination of real and virtual objects of cases 2, 3 and 4. Graphic engine packages such as Unity [27], provide the frame work for comprehensive video processing. Unity is a cross-platform game engine that utilises object-orientated programming through its C# language support. Video and 3D image overlay are combined, under Unity, and issues such as timing and registration can all be controlled, resolved or minimised through the underlying software.

Performing image overlay within the SNAP user [3] environment, is both easier and harder. The convenience of the SNAP system mitigates some of the intricate processes necessary to add AR capabilities. It is important to note that the live video stream is not modified in the SNAP platform.

In RALfie, the SNAP platform has a designated area of screen that is called the stage. The stage is where all the objects of animation and other output data are displayed. The simplest form of AR is to super-imposing the cameras view below the stage. This is done by connecting the stage's background directly to the camera stream. The camera steam may be resized to any size and placed at any position on the stage or a full screen mode can be applied.

Experiment makers must include a command to start the AR. If the AR is not started then the SNAP environment behaves like a typical non-AR setup. Once the AR mode is started, the camera is visible. Then the experiment makers can include any object they wish on the stage that will appear on top of the video stream. The experiment makers can make precise movements according to the underlying changes in the camera feedback.

Fig 7 shows an example of this type of AR. The experiment activity concerned is a traffic light system as shown previously in Fig 4 and Fig 5. The rig has some LEDs that must go 'on' and 'off' and virtual objects (sprites) are the cars which must stop and move according to the LEDs status. The LEDs are connected to the ports on the MCU which is a Beaglebone and controlled with

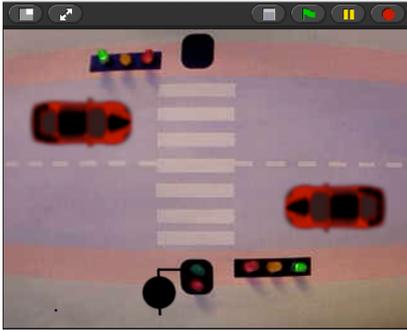


Fig 7. A traffic light example in SNAP with real LEDs and virtual cars

commands/instructions from the SNAP UI to the MCU. This particular experiment acts as a practice for experiment makers to create program for the experiments including AR. Obviously, in context of a centralized RAL, super-imposing of a virtual object on video stream can be done to enhance the user experience.

The video feedback shows the LEDs and the background roads etc. and the cars' movements are controlled through SNAP depending upon the data received from MCU. This type of AR presents two kinds of issues:

1. *Stability*: The camera view is assumed to be static. The problem may arise if and when the camera view changes due to the camera getting moved accidentally, in which case, the changed view can be detected and the experiment makers can be notified to duly put the camera back in place. However, if the AR objects are very precisely programmed or stored in a library, they may require re-calibration of coordinates to ensure correct UI interactions. There is currently no object identification procedure to re-align the virtual objects accordingly.

2. *Response time*: The response time is the time taken to retrieve any data/video from the experiment maker node to the user node. On the internet this may be high. The animation frame rate will be typically faster than the frames from the video, thus it will create a lag in user interaction if every frame of the stage is attached to a new video frame. For this purpose, the video is handled by a separate process that runs parallel to the snap execution platform running the virtual objects activity. Whenever the video frame is retrieved, the stage background is updated accordingly. Thus the users' interaction with the UI components remains normal.

A second problem is the difference between the arrival rate of data and the SNAP animation frame rate. Due to response time, the data may arrive at a later time than the relevant frame where the data was supposed to have any effect. Thus, all SNAP execution including animation is suspended when a message or instruction a is issued from the SNAP to the MCU at time T_a . During this period the virtual objects do not move or operate thus creating a paused state until the data is received. With higher latencies, the number of paused states will increase in an interval of time, thus affecting quality of experience. But the data and SNAP animation will remain synchronized thus not affecting the learning objectives.

The last problem is the arrival time difference between the data and video stream. Typically, the command and sensor data exchanged between the nodes is very small

and delivered at faster rate than the video as well. This causes the problem of de-synchronization between the video frame and the virtual objects. Thus, the SNAP execution engine is paused until a new video frame T_a^v is received after receiving a new data T_a^d after an instruction a is issued. With stable internet conditions, there will be least effect on the performance and interaction of the users. Thus the paused time after an instruction a is issued to the experimental rig from the user interface is:

$$\text{Paused Time}(a) = \max\{T_a^v - T_a, T_a^d - T_a\}$$

Improving the immersion factor for users, can mean allowing the AR services to run without the need for a data stream. Detecting the lights changing does not require a signal from the MCU to the remote SNAP terminal. Vehicles may continue to move until the lights change. User interaction (to switch the lights) is then only affected by the latency in one direction, to the experimental rig. This delay is unlikely to annoy the user.

VI. LIMITATIONS AND FUTURE WORK

The prototype system for ROI (or its implementation - OIT) is successful in principle to provide a generic tool to create AR interfaces for multiple experiments using the steps described earlier. But it is not able to support all types of experiments. The major performance shortcomings that can be refined are:

1. The JavaScript based SNAP environment runs in a web-browser. The OIT AR tools increase the CPU rate to more than 33% on a 2.5GHz, Intel i5 processor using the web browser Firefox version 41. This indicates that AR tools require considerable computational power which may not be available on mobile devices.
2. In *Step 2*, basic frame subtraction methods can only work in relatively static experimental rigs. Large movements in the rig, other than the items of interest, cause significant flooding of the data. To reduce the loss of signal, more comprehensive subtraction methods are required, adding to the processing power requirements and delays.
3. In *Step 3*, a clustering algorithm the DBSCAN algorithm is used which has 2 inputs ϵ and p where ϵ signifies the radius around a point and p is the minimum number of points (or pixels) around a given point. The resultant clusters will have each point in the cluster surrounded by a minimum of p clusters within a radius of ϵ . This is an ideal way to determine objects and reduce noise in the video input. However, this also adds to parameters that need to be altered to an extent to identify the desired objects correctly. Further improved implementations of this have to either automatically adjust this or the users choose the desirable values.
4. In *Step 4*, the objects properties - average colour for red, blue and green along with a range of minimum and maximum heights and widths of the detected clusters are stored. While this is sufficient to identify small and mobile objects with uniform colour, it may fail on some scenarios with larger objects. .

One of the inherent limitations of AR implementations in RAL includes the need for static camera positions. Ambient lighting changes can also affect the outcome. It will create larger differences between the background (B)

and any subsequent frames. Thus the experiments with AR tools must be setup in a well-defined environment. Further work will look into minimizing the effect of response time on the performance of AR tools. There are also issues with identical objects in the same experiment setup that needs to be addressed.

The OIT in this work assumes that there are fewer moving or changing components compared to static objects in an experiment view. At the moment issues such as occlusion are not addressed, i.e. when an object is covered by other objects. The OIT described here can be used only for moving or largely changing objects visible to the camera. Future work will research computer vision techniques to achieve reliable OIT in a variety of experiment types with minimal impact on computer resources.

VII. CONCLUSIONS

AR tools are based on the activity of the virtual and real components in the experiment video stream. Such services can help users of the experiments to quickly identify the changing parameters of the experiment and help developers become acquainted with the relationship of those parameters and the RAL rig operation. In short, augmented reality tools can help users and makers to recognize the most important learning concepts in the experiment. It can highlight important data and help user to understand the experiment. It provides a higher level of immersion in the experiment, such that the user does not feel disconnected from the resources and increases authenticity of the learning experience.

REFERENCES

- [1] Z. Nedic, J. Machotka, and A. Nafalski, "Remote Laboratories Versus Virtual and Real Laboratories," in *33rd ASEE/IEEE Frontiers in Education Conference*, Boulder, 2003. <http://dx.doi.org/10.1109/fie.2003.1263343>
- [2] J. Garcia-Zubia, D. López-de-Ipiña, and P. Orduña, "Mobile Devices and Remote Labs in Engineering Education," in *Eighth IEEE International Conference on Advanced Learning Technologies (ICALT'08)*, 2008, pp. 620-622.
- [3] A. Maiti, A. D. Maxwell, A. A. Kist, and L. Orwin, "Joining the Game and the Experiment in Peer-to-Peer Remote Laboratories for STEM Education," presented at the exp.at 2015, Ponta Delgada, São Miguel Island, Azores, Portugal, 2015. <http://dx.doi.org/10.1109/expat.2015.7463268>
- [4] R. T. Azuma, "A Survey of Augmented Reality," *Presence*, vol. 6, pp. 355-385, 1997. <http://dx.doi.org/10.1162/pres.1997.6.4.355>
- [5] A. Cardoso, M. T. Restivo, M. Quintas, F. Chouzal, M. Rasteiro, J. Marques, et al., "Online Experimentation: Experiment@Portugal 2012," in *REV 2014: 11th International Conference on Remote Engineering and Virtual Instrumentation*, Porto, Portugal, 2014, pp. 303-308. <http://dx.doi.org/10.1109/rev.2014.6784175>
- [6] H.-K. Wu, S. W.-Y. Lee, H.-Y. Chang, and J.-C. Liang, "Current Status, Opportunities and Challenges of Augmented Reality in Education," *Computers & Education*, vol. 62, pp. 41-49, 2013. <http://dx.doi.org/10.1016/j.compedu.2012.10.024>
- [7] J. M. Andujar, A. Mejías, and M. A. Marquez, "Augmented Reality for the Improvement of Remote Laboratories: An Augmented Remote Laboratory," *IEEE Transactions on Education*, vol. 54, pp. 492-500, 2011. <http://dx.doi.org/10.1109/TE.2010.2085047>
- [8] H. Kaufmann and D. Schmalstieg, "Mathematics and geometry education with collaborative augmented reality," *Computers & Graphics*, vol. 27, pp. 339-345, 6// 2003.
- [9] J. Camba, M. Contero, and G. Salvador-Herranz, "Desktop vs. Mobile: A Comparative Study of Augmented Reality Systems for Engineering Visualizations in Education," presented at the Frontiers in Education, 2014. <http://dx.doi.org/10.1109/fie.2014.7044138>
- [10] D. W. F. van Krevelen and R. Poelman, "A Survey of Augmented Reality Technologies, Application and Limitations," *International Journal of Virtual Reality*, vol. 9, pp. 1-20, 2010.
- [11] S. Abu Shanab, S. Odeh, R. Hodrob, and M. Anabtawi, "Augmented Reality Internet Labs Versus Hands-On and Virtual Labs: A Comparative Study," in *2012 International Conference on Interactive Mobile and Computer Aided Learning (IMCL)*, Amman, Jordan, 2012, pp. 17-21. <http://dx.doi.org/10.1109/IMCL.2012.6396444>
- [12] A. Maiti, A. A. Kist, and M. Smith, "Key Aspects of Integrating Augmented Reality Tools into Peer-to-Peer Remote Laboratory User Interfaces," presented at the REV 2016, Spain, 2016. <http://dx.doi.org/10.1109/rev.2016.744434>
- [13] M. Guenaga, I. Menchaca, A. de Guinea, O. Dziabenko, J. García-Zubia, and M. Salazar, "Serious Games, Remote Laboratories and Augmented Reality to Develop and Assess Programming Skills," in *Frontiers in Gaming Simulation*, 2014, pp. 29-36.
- [14] A. Maiti, "Interactive remote laboratories with gesture based interface through microsoft kinect," in *Remote Engineering and Virtual Instrumentation (REV)*, 2013, pp. 1-4.
- [15] F. Weichert, D. Bachmann, B. Rudak, and D. Fisseler, "Analysis of the Accuracy and Robustness of the Leap Motion Controller," *Sensors*, vol. 13, p. 6380, 2013. <http://dx.doi.org/10.3390/s130506380>
- [16] M. F. Shiratuddin and K. W. Wong, "Game Design Considerations When Using Non-touch Based Natural User Interface," in *Transactions on Edutainment VIII*, Z. Pan, A. D. Cheok, W. Müller, M. Chang, and M. Zhang, Eds., ed Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 35-45. http://dx.doi.org/10.1007/978-3-642-31439-1_4
- [17] A. Bin Tomi and D. R. A. Rambli, "A Conceptual Design for Augmented Reality Games Using Motion Detection as User Interface and Interaction," in *Visual Informatics: Sustaining Research and Innovations: Second International Visual Informatics Conference, IVIC 2011*, pp. 305-315. http://dx.doi.org/10.1007/978-3-642-25200-6_29
- [18] M. B. Ibáñez, Á. Di Serio, D. Villarán, and C. D. Kloos, "Experimenting with Electromagnetism using Augmented Reality: Impact on Flow Student Experience and Educational Effectiveness," *Computers & Education*, vol. 71, pp. 1-13, 2014. <http://dx.doi.org/10.1016/j.compedu.2013.09.004>
- [19] P. Milgram and F. Kishino, "A Taxonomy of Mixed Reality Visual Displays," *IEICE Transactions on Information Systems*, vol. E77-D, 1994.
- [20] S. Ternier, R. Klemke, M. Kalz, P. Van Ulzen, and M. Specht, "ARLearn: Augmented Reality Meets Augmented Virtuality," *J. UCS*, vol. 18, pp. 2143-2164, 2012.
- [21] Z. Zivkovic, "Improved adaptive Gaussian mixture model for background subtraction," in *Pattern Recognition, 2004. ICPR 2004. Proc of the 17th Intl. Conference on*, 2004, pp. 28-31. <http://dx.doi.org/10.1109/icpr.2004.1333992>
- [22] T. Auer, S. Brantner, and A. Pinz, "The integration of optical and magnetic tracking for multi-user augmented reality," *Elsevier: Computers & Graphics*, 1999. [http://dx.doi.org/10.1016/S0097-8493\(99\)00106-5](http://dx.doi.org/10.1016/S0097-8493(99)00106-5)
- [23] P. Moore, A. A. Kist, A. Maiti, and A. D. Maxwell, "Work in progress: Remote experiment control through gesture recognition," in *REV 2016*, pp. 377-379.
- [24] A. Basu, C. Saupé, E. Refour, A. Raij, and K. Johnsen, "Immersive 3DUI on one dollar a day," in *3D User Interfaces (3DUI), 2012 IEEE Symposium on*, 2012, pp. 97-100. <http://dx.doi.org/10.1109/3dui.2012.6184191>
- [25] C. Stauffer and W. E. L. Grimson, "Adaptive Background Mixture Models for Real-Time Tracking," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1999. <http://dx.doi.org/10.1109/cvpr.1999.784637>
- [26] V. Estivill-Castro, "Why So Many Clustering Algorithms: A Position Paper," *ACM SIGKDD explorations newsletter*, vol. 4, pp. 65-75, 2002.
- [27] M. P. Rogers, "Bringing Unity to the Classroom," *Journal of Computing Sciences in Colleges*, vol. 27, May 2012 2012.

AUTHORS

Mark Smith, Ananda Maiti, Alexander Kist and Andrew Maxwell are with the University of Southern Queensland, Toowoomba 4350, Australia. (email: mark.smith@usq.edu.au, anandamaiti@live.com, kist@ieec.org, andrew.maxwell@usq.edu.au)

Submitted 15 May 2016. Published as resubmitted by the authors 17 June 2016.