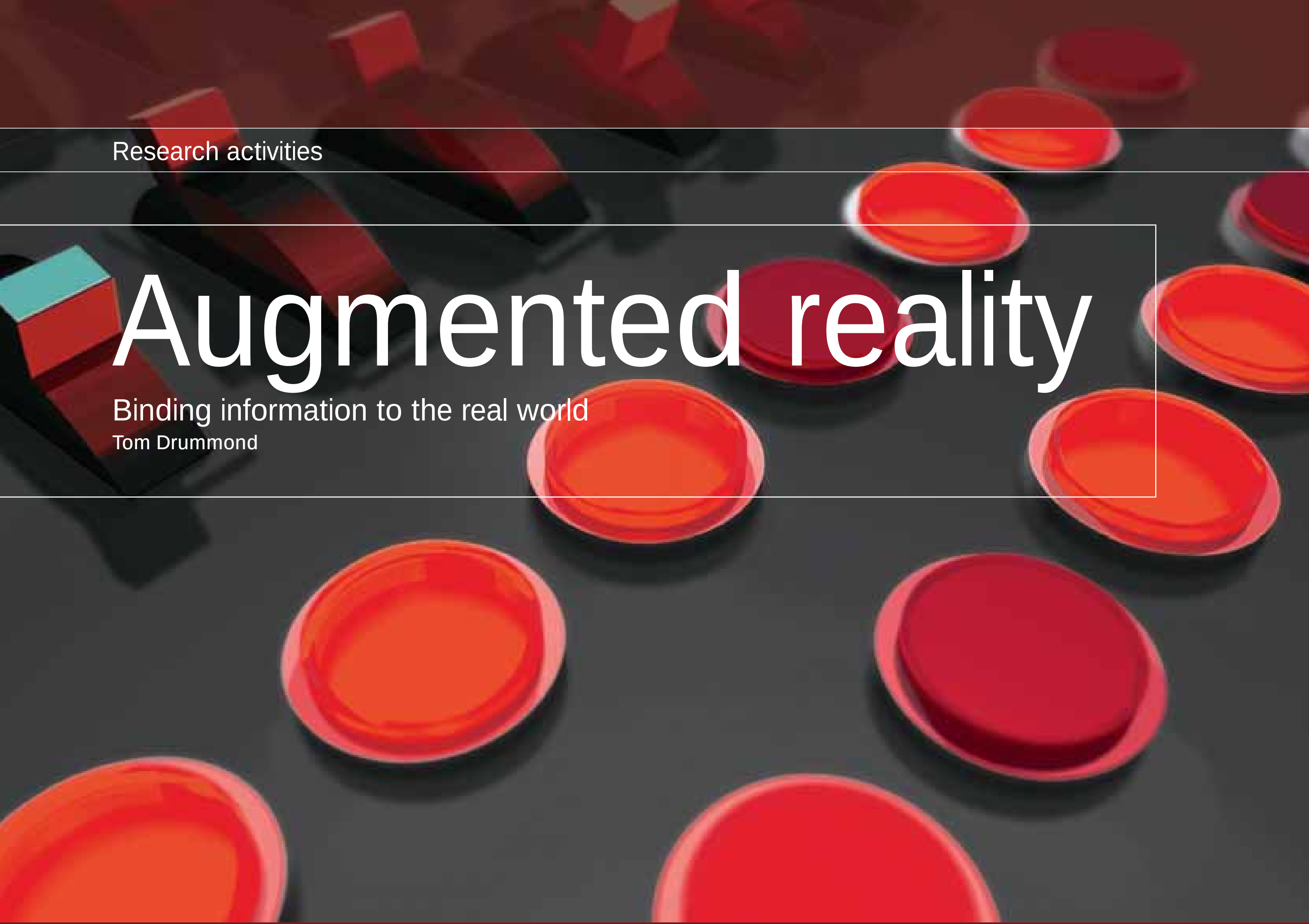
Augmented reality

Binding information to the real world

Tom Drummond



Adjusting the volume of a television set might seem a fairly intuitive task. But when there are several remote controls to choose from, finding the right one may be a trial and error process.

In industry, similar situations occur. Even well-trained and experienced operators may sometimes have difficulty knowing which switch corresponds to which function. In a rarely occurring but critical situation, valuable time is lost as the operator struggles to find the right switch.

In contrast to using the wrong remote control in the living room, which causes no more damage than wasted time and embarrassment, errors in an industrial environment can have severe and costly consequences. Is there a better way of providing relevant information?

Most modern engineering solutions contain a software component. There are computers in everything from televisions, cars and kitchen appliances, to large-scale industrial plant systems. A modern credit card has an embedded microprocessor and enough software to run a file system, authenticate the owner's presence and create digital signatures for financial transactions. This ubiquity of computing in engineering solutions has resulted in an increase in software complexity.

Just as engineering involves the construction of mathematical models of a problem and its environment, so software engineering requires the creation of software models; a credit card has a software representation of financial transactions (who is paying whom how much?), the owner (name and billing address) and authentication (has the user authenticated and what is the procedure for doing so?).

This increase in complexity brings with it a number of problems that arise because the relationship between the software's model of the world and the real world (or the user's model of it) may not be obvious. One way in which this manifests itself occurs when software has an internal state that may not be visible or obvious to the user. One example is a software controlled indicator in a car. Instead of the state being represented by the position of the indicator stalk (up = right, down = left, centre = off), such software-controlled systems see the stalk immediately return to its central position on release by the human. The record of the direction in which the stalk was pushed is stored only by the software. The same is true for how hard the stalk was pushed (soft = blink indicator lights only a few times while changing lanes and then cancel automatically, hard = blink lights until cancelled by pushing softly in the other direction). The problem is that the driver does not necessarily know which of these modes has been engaged as both result in the same immediate behaviour (the indicator lights activate). If the driver incorrectly believes that the soft mode has been activated, the vehicle will continue to indicate beyond the time required. On the other hand, if the driver incorrectly believes the hard mode is engaged, the vehicle will indicate in the wrong direction when he or she attempts to cancel.

Another way that increased complexity causes problems can loosely be termed the *binding problem*.

The binding problem

One classic example of this problem occurs in large rooms with many light switches. A user may wish to turn off a particular bank of lights, such as those near a projection screen, but is faced with a panel of many switches. How can the user know which switch operates which light? (the usual approach is to try them all in turn until the desired outcome is achieved). This problem arises because the *binding* between the switches and the lights is not obvious, and because such rooms are typically only used intermittently by a given individual, making the binding difficult to memorise.

The author's living room provides another typical example. On the table next to the sofa there is a small mountain formed of six remote controls. These control the TV, video, CD, networked music, cable and digital TV box. Visitors find it nearly impossible to figure out which remote controls which unit (some members of the household have difficulty too). Again, this is because the binding is not obvious. The situation could be improved somewhat with a universal remote control, but even then an operation such as muting the sound to take a telephone call requires identifying the correct device 1.

Anyone who has hiked in mountainous country knows that map reading is a real skill. It is possible to have a very clear view of both the local scenery and a map, but still to be lost as the binding between mountains as seen in the real world and those seen on the map is not always obvious.

> Even well-trained and experienced operators may sometimes have difficulty knowing which switch corresponds to which function. In a rarely occurring but critical situation, valuable time is lost as the operator struggles to find the right switch.

The design and operation of large plant control systems also introduces a binding problem. Such systems present the operators with control views of the plant, which are topological in nature. It is not uncommon for such views to collectively make reference to tens of thousands of tags, each of which corresponds to a component or location within the plant. Many of the alarms generated by the control system require that the control room operator interact with a plant operator by radio in order to make a visual verification concerning some component of the plant. This means that these two users must manually perform the binding between the very large database of tags and the real plant components to which they refer.

This human complexity has consequences in both training times and in speed of response to rare circumstances.

1 Which remote control goes with which device? This is a simple example of the binding problem

Augmented Reality interfaces

One approach to solving the binding problem is to exploit new advances in Augmented Reality. These allow the software engineer to take the graphical user interface (GUI) capabilities of a conventional computer setup which are normally restricted to the computer screen, and make them available across the whole world.

This would be easy if the whole world had computerised input and graphical display output, however this is not the case and intermediary hardware is required. In the early days of AR, the most common way to achieve this was to exploit Head Mounted Displays (HMDs). Optical see-through head mounted displays contain a half silvered mirror which allow computer generated graphics to be merged with the user's view of the real world. This allows arbitrary graphical displays to be superimposed on real objects. Provided that the user's head can be tracked to sufficient precision, and the display can be rapidly updated in response, these graphics can be made to look as though they are stable in the real world and hence belong there rather than to the screen of the HMD.

One of the earliest examples of AR was implemented by the Boeing Company to assist in the production of wiring harnesses for aircraft. Such harnesses are normally constructed on peg-boards with each wire of the harness printed onto the peg-board, thus necessitating one peg-board per aircraft. In the AR setup, the worker wore a HMD which displayed the location of each wire onto a generic peg-board. Such a solution allows a significant saving in storage of peg-boards, but it admits many other benefits as well. It is now much easier to upgrade a design since the per-aircraft information is now stored only in the computer, rather than printed onto a board. Further, the dynamic capability of the HMD can be exploited so that the wires can be presented to the operator one or a few at a time.

Similarly, one can imagine that a hiker wearing a HMD could see the name of each mountain (or a grid reference) superimposed above the real view of their summit. Or the author could use a HMD in his living room, together with a pointing device displayed in the HMD to identify the item of entertainment hardware that he wishes to interact with by clicking on the real view of it . . .

. . . or are these examples rather a stretch of the imagination? In fact it is rather hard to picture a hiker choosing to obstruct his view of nature and the scenery by choosing to wear a bulky HMD. Similarly if somebody were to wear a HMD in the living room, family members would be likely to object to the impediment to social interactions.

There are also many other problems with HMDs as a means of delivery of Augmented Reality user interfaces. They can typically only augment a rather narrow field of view of the user and they still are rather expensive. For about $35,000 it is now possible to purchase a see-through HMD which augments 60 degrees of the user's field of view.

2 A handheld AR can simplify navigation

Factbox About the author

Dr Tom Drummond is a university senior lecturer in the machine intelligence laboratory of the department of engineering at Cambridge University and a Fellow of St Catharine's College. He received his PhD from Curtin University in Western Australia. His research interests include real-time computer vision, sensor fusion with an emphasis on applications to visually guided robotics and augmented reality user interface technologies.

Such devices need to be calibrated specifically for each user so that real and virtual objects are seen to align. Finally, they inevitably have latency which means that as the user moves their head, the computer graphics lag behind the real motion, typically by at least 100ms. This is enough to cause serious motion sickness, restricting use to a few minutes at a time.

> Research sponsored by ABB, is looking at ways of representing and matching models of the world against the live view coming from the video camera.

Handheld AR

Such problems with HMDs have led researchers to consider other media for delivery of AR interfaces. The most popular alternative is handheld AR **2**. This works by using a video camera attached to a handheld screen which acts as the viewfinder for the camera. AR is then implemented by trapping the video in-flight between the camera and the screen and editing it to insert virtual graphical objects and user interface components.

Handheld devices solve most of the problems with HMDs: they are cheap, nobody is concerned about small latencies, wide fields of view can be obtained by using a wide angle lens, they can be put away in a pocket and taken out as required. They also bring other benefits, for example by allowing the user interface designer to integrate conventional two-dimensional UI components into an application.

In the author's living room, anybody could pick up such a handheld device, point it at the entertainment hardware and see a live view of that hardware on the screen. Using a stylus, they could click on the image of the CD player and be presented with a conventional looking GUI for controlling it.

Similarly an operator equipped with this device in a plant could receive graphical navigation information superimposed over a live view of their environment to indicate which physical object corresponds to a particular tag. They could also click on the live view of an object to be presented with a topological view, or maintenance information, or sensor trends, or whatever else the application designer deems to be valuable to that operator. Further, such a device can be used to widen the communications channel between two operators beyond the single audio channel used at present to include geographically registered spatial information.

What needs to be done?

Currently such devices are in their infancy and significant research effort is going into developing the tracking technologies that are needed to make them work. The most promising prototypes combine multiple technologies, for example using inertial sensors to supplement computer vision, or adding GPS or ultra wide band (UWB) localisation.

Such approaches have significantly improved the robustness and precision with which these systems operate. The author's research, sponsored by ABB, is looking at ways of representing and matching models of the world against the live view coming from the video camera so that AR can be provided robustly and reliably on demand in large scale environments, such as complex plant.

The benefits that can come from this are potentially substantial. Augmented Reality provides a way of solving the binding problem by making the real world act as an index into the database of computer system components. This in turn exploits the spatial capabilities of the human users which remains one of their great untapped strengths in conventional software systems.

Tom Drummond
Cambridge University
Cambridge, UK
twd20@hermes.cam.ac.uk