DISPLAY REGISTRATION FOR DEVICE INTERACTION
A Proof of Principle Prototype

Nick Pears
Department of Computer Science, University of York, York, YO10 5DD, UK

Patrick Olivier and Dan Jackson
Culture Lab, King's Walk, Newcastle University, Newcastle, NE7 1NP, UK

Keywords: Human-Computer Interaction, Image Registration, Real-Time Vision.

Abstract: A method is proposed to facilitate visually-driven interactions between two devices, which we call the *client*, such as a mobile phone or personal digital assistant (PDA), which must be equipped with a camera, and the *server*, such as a personal computer (PC) or intelligent display. The technique that we describe here requires a camera on the client to view the display on the server, such that either the client or the server (or both) can compute exactly which part of the server display is being viewed. The server display and the clients image of the server display, which can be written onto (part of) the client's display are then registered. This basic principle, which we call "display registration" supports a very broad range of interactions (depending on the context in which the system is operating) and it will make these interactions significantly quicker, easier and more intuitive for the user to initiate and control. In addition, either the client or the server (or both) can compute the six degree-of-freedom (6 DOF) position of the client camera with respect to the server display. We have built a prototype which proves the principle and usefulness of display registration. This system employs markers on the server display for fast registration and it has been used to demonstrate a variety of operations, such as selecting and zooming into images.

1 INTRODUCTION

The last decade has seen an explosion in mobile communications, evidenced by the enormous take up of mobile phones and personal digital assistants (PDAs) or hand-held computers. More recently there has been a drive to integrate the many devices that might exist in our environment through the use of personnel area network (PANs) using technology such as Bluetooth and infrared networking. Easy integration means that a network connection can be established between, for example, a PDA and the desktop computer, and thus information can be exchanged between the two. Whilst Bluetooth connectivity is indeed easy for a user to establish, users are required to use an additional application on the PDA (for example file browser) to access the contents of the desktop computer, and vice versa. Figure 1 illustrates the case of downloading a folder on the desktop to a PDA using specialist software on the PDA. Here the handset and the desktop are used as separate computers, just as we might remotely access one desktop computer from another.

Figure 1: Transfer using separate application.

This physical separation of handset and desktop, and the incumbent complexity for users in trying to connect between the two, is the problem addressed by this paper. Our vision is of a technology whereby the display of the handset could be treated as an al-

Pears N., Olivier P. and Jackson D. (2008).
DISPLAY REGISTRATION FOR DEVICE INTERACTION - A Proof of Principle Prototype.
In *Proceedings of the Third International Conference on Computer Vision Theory and Applications*, pages 446-451
DOI: 10.5220/0001075104460451

Figure 2: Transfer using display registration.

most indistinguishable part of the display of the desktop computer. For example, by holding the handset over the desktop's display, the content of the desktop screen below the handset appears on a region of the handset's display. Figure 2 illustrates the concept; a user holds the handset, equipped with rear-mounted camera, over the desktop. In doing so the user can see, on the handset, the contents of the display below it, and manipulate elements of this display as if they were elements of the handset's display itself. This functionality offers a wide range of applications, for example:

(i) *Data exchange between client and server (data push/pull).* Suppose that the user uses the client display to observe an icon of a file displayed on the server system. Suppose that the user then clicks on the image of this file icon using a button (or stylus) click on the the client device. Since the position of this click on the client display can be converted to the corresponding position on the server display, which is passed to the server over the data communication channel (eg bluetooth), the server system can determine what file is being requested as it knows what has been "virtually clicked" on the server display. Then it can send the file data across the communication channel to be stored on the client system. In this way data can easily be pulled from the server to the client device, or pushed from client to server.

(ii) *Semantic magic-lens interaction.* In an implementation of a *semantic lens*, the action that the user requests is inferred from what the user is pointing at. As an example, there may be a map of the UK on the server display. By pointing the client camera at a particular town (York, for example), the application may infer that all contacts from an address book database that have the keyword "York" in the city field of the address book are copied across to the client address book.

(iii) *Using 6 DOF client pose to mediate interaction.* It is possible to mediate interactions by using the client device in the role of a 2D and/or 3D mouse. Given that the display registration has been computed, the six degree of freedom pose of the client can be computed, if the camera/display screen parameters (such as aspect ratio) are also known through device specification or a standard calibration procedure. The operation of a 2D mouse, for example, is straightforward: given that the two displays are registered, the centre of the client display can be highlighted using cross-hairs, on the server display and thereby act as a mouse pointing device. Selection of a file could consist of pointing at a file icon and then 'peeling it off' using a rotation of the hand. This rotation can be detected on the client device and interpreted as a request to pull a copy of the file off the server system and store it on the client system.

Such a technology has a number of possibilities for intelligent public information displays, with which users might pull and push information simply using their PDAs or mobile phones, thereby opening up a host of new commercial opportunities both for handset vendors, retailers and service providers. Examples include retrieving the details of property for sale in a estate-agent window, or purchase of cinema tickets from an intelligent display.

The immediate realisation of these applications requires one particular innovation: that the position of the handset (PDA or mobile phone) can be tracked relative to the screen of the desktop display (or the display of any computer). We call this problem *display registration*, and the notion of registering one display with another, in this manner, is the core of the technical work required to realise our novel concept.

The rest of the paper is structured as follows. The following section describes fully the concept of display registration. Section 3 describes the the two main categories of display registration, namely marker-based and markerless. Section 4 describes the prototype marker-based system that we have built, while the following section describes our first evaluation of that system. A final section is used for conclusions and suggestions for further work.

2 DISPLAY REGISTRATION

The work described here relates to the interaction of a pair of devices, which can communicate data across a communication channel (typically wireless, such as wifi or bluetooth), where one of these devices is

equipped with or linked to an imaging device, such as a camera, which is able to view the other device's display, such that the camera's image is *registered* with that display. The term *registered* means that for any (pixel) position in the viewed display we know its corresponding position in the captured image of the display. We call the concept of a display with a registered image of that display *display registration*, as this is an instance of image registration. The captured image of the display, which is registered to the display itself, can be passed to the display on the camera-equipped device for the system operator to use in his/her current task. In most applications, the camera equipped device will be smaller and manoeuvrable by hand. We call this the *client* device and movements and button (or stylus) clicks of this device control the way in which the system operates, within a certain context. The other device, will, in general, be a larger static device and we will call this the *server* device.

2.1 Device Interaction via Registered Display Operations

In typical use of this method, the mobile client device is moved around by the user, whilst maintaining at least a small part of the server display in it's field of view. Throughout this motion, the client camera image and hence the client's display of that image to the user are registered with the server display. That is, irrespective of the change in relative position of the client device, we can always compute where any position on the server display appears on the client camera image and the display of that client camera image. Also, since we can easily compute the inverse transformation, we can choose any position on the client camera image, such as the centre or one of the image corners, and determine the corresponding position on the server display. We call the concept of maintaining the correspondence between client and server displays *maintaining display registration*. The fact that the displays are registered enables a large range of possible interactions and data exchanges between client and server devices. It is envisaged that the user may control this interaction through a variety of modes, which are effectively different contexts in which to interpret *registered display operations*.

3 REGISTRATION METHODS

For a planar client image plane and a planar server display systems, we need to find a plane-to-plane mapping that allows us to compute the display registration. This mapping encodes the (idealised) imaging process of the camera (intrinsic parameters) and the six degree-of-freedom pose of the client image plane relative to the server display (extrinsic parameters). It is well-known that this transformation, called a planar homography, can be represented by a 3x3 matrix, $\mathbf{H}$, such that $\lambda \mathbf{x}_i = \mathbf{H} X_i$, where X_i is a point on the server display, x_i is the corresponding point in the client image and λ is a constant. The matrix $\mathbf{H}$, is defined up to a scale factor and hence has eight degrees of freedom. Thus it can be estimated by standard linear methods if four corresponding points are known across the client image and server display, with the constraint that no three are collinear. In this case we have eight independent constraints and $\mathbf{H}$ is fully defined (up to scale). More corresponding points can yield a more accurate estimate of $\mathbf{H}$, using some variant of a least-squares technique. Various estimation techniques for $\mathbf{H}$ are detailed by Hartley and Zisserman (Hartley and Zisserman, 2004).

The question now arises: how to we find four or more corresponding points across the server display and client image of that display? This problem can be divided into two categories: (i) marker-based and (ii) natural (markerless).

In *marker-based display registration*, the server is required to maintain a dynamic display of four distinctively coloured reference targets, no three if which are collinear, which can easily be detected and segmented by the client. Given that the position of these can be detected in the client, these positions can be transmitted to the server, which knows where the targets were displayed on the server display. A planar homography estimation method can then be applied to register the displays without any prior calibration of the camera. Note that, since the homography transformation between the server display and client display is known when the displays are registered, it is possible to change the markers in the server display, such that the shape, size and position of the markers is constant in the client image irrespective of camera viewing pose. This leads to more reliable detection of the markers, since they do not become too small to detect as the client camera moves away from the server display.

In *natural display registration*, no dynamically controlled markers are used to aid registration (homography computation). Registration is achieved by matching the client image to the unmodified server display (although one can choose to use textured backgrounds and windows) and this may be achieved using one of several techniques in the computer vision and pattern recognition literature. Perhaps the simplest approach is to use corner extraction (Harris and Stephens, 1988), (Smith and Brady, 1995) fol-

lowed by matching across the two views. The obvious difficulty is solving the correspondence problem: which corners in the server display match the corners in the client image? The spatial arrangement of corners may be used as a matching constraint, for example, five corners in a general position provide a pair of cross-ratio invariants (Sinclair and Blake, 1996), although cross-ratio computation is noise sensitive. Rather than using the spatial arrangement of corners, one can compute specific features in the image that are distinctive and invariant to the imaging process. Several researchers have formulated invariant features, such as Schmid and Mohr's (Schmid and Mohr, 1997) rotationally symmetric Gaussian derivatives and Baumberg's (Baumberg, 2000) second moment matrix, which gives invariance to affine transforms. Lowe's scale invariant feature transform (SIFT), is perhaps the most successful of these (Lowe, 2004). SIFT features are invariant to similarity transforms (translation, rotation and scale changes). Although this technique also provides some robustness to affine transforms, large non-affine distortions (caused by large pan and tilt rotations of the client) are likely to cause the system to loose track.

4 A PROTOTYPE SYSTEM

We have implemented a working prototype by rendering distinct markers on the desktop display and tracking their position as seen by a smartphone with an integrated camera, as shown in figure 3.

The first stage was to choose a suitable dynamic target pattern. We have elected to use four squares of a distinctive green colour. Note that an image of the four squares has a rotational ambiguity and so, on initialisation of the system, it is necessary to display a target that is not rotationally symmetric. We used one square with a hollowed out centre to break this symmetry and give us an unambiguous orientation. Furthermore, by alternating between "full squares" and "hollowed out squares" in subsequent frames, we are able to deal with the time lag between image capture and processing on the client side and the display of the updated target position on the server screen, which would otherwise cause instability in the tracking process.

The computation of the planar homography between the actual marker positions on the desktop display, and their coordinates in the image seen by the handset, allows the fast and highly accurate calculation of the mapping between pixels on the handset and the desktop, thereby facilitating a range of applications.

The basic sequence of events for the system operation is as follows:

- The client and server establish a communication channel over a Bluetooth link.
- The initialisation process starts with the server moving the special initialisation target systematically around the server screen, starting from the centre and working outwards towards the screen edges. The user starts by aiming the camera approximately towards the centre of the server display.
- When the client is able to acquire the target, the 2D image positions of the four centres of the target squares (eight values) in the target are transmitted to the server. The server then associates the corresponding four display positions with these target positions and computes the plane-to-plane homography mapping between the two displays.
- The server then computes the corresponding server display positions for the corners of the markers in the client image. This indicates how the target pattern should appear in the server display, for the pattern to remain constant in appearance on the client display.
- For further cycles of operation, the four target centres are switched between "filled" and "hollow" so that the time lag between server display of target and client computation of target pose can be determined.

We now explain the target segmentation and acquisition in more detail

4.1 Target Segmentation and Acquisition

The target colour that is detected on the client is modelled using RG-chromaticity colour space. In this space the red and green colour components are normalised by dividing by intensity, which is the sum of the RGB components. This gives some immunity to intensity variations, but there are more sophisticated approaches to colour normalisation, such as those suggested by Alexander (Alexander, 1999). In our approach the RG-plane in colour space is divided into bins and the image of the targets is selected manually. All of the manually selected pixels populate these bins to give a colour model as a histogram in RG-space. We can thus determine whether a pixel falls within the modelled colour space and classify it as either belonging to the target or not. The simple approach that we use is to find the mean pixel position for the segmented pixels and divide the image

into four (not necessarily equal) segments in directions associated with tracked orientation of the target. The mean positions in these segmented regions correspond to the four centres of the square targets, which is the information that we require.

5 EVALUATION

Usability testing has been performed using the talk-aloud protocol, in which participants describe their observations, thoughts and actions as they complete specific tasks. Four participants were asked to each complete two tasks. Both tasks used the display registration system deployed on a PC with a 17" LCD display communicating with a smartphone implementation of the client software over Bluetooth. An image of a user performing these tasks is shown in figure 3, note that the segmented targets on the client smartphone are highlighted in red.

The first task involved a specially-written photo montage demonstration application, in which three digital photographs were laid out in a particular starting position, as in figure 4. The users were asked to rearrange the photos to resemble a second configuration with different positions, orientations and scale (as shown in figure 5) using the display registration system. A target was drawn on the PC display at the centre of the smartphones camera view. By depressing a trigger button on the smartphone, the user was able to manipulate the targeted image. The images could be moved by translating the phone parallel to the display, rotated them by rotating the phone parallel to the display, and scaled by translating the phone towards or away from the display.

The second task used the system to replicate a specific outline drawing of a house, as shown in figure 6, within the Microsoft Paint application. The software set-up provided the correct brush size and colour, and the participants were only required to make their own brush strokes using the smartphone. An example output from one of the participants is shown in figure 7.

In our tests, our client device was a Siemens SX1 smartphone, with a series 60 phone processor (130MHz TI OMAP 310), running Symbian OS V6.1. The system specifications were: frame rate, 8 Hz; camera field of view, 30; maximum phone movement, 0.3m/s at 0.45m from a 17" (0.34m 0.27m) display; target re-acquisition time, 2-3 seconds. AVI videos of the four users performing these two tasks can be found online at http://irgen.ncl.ac.uk/data/temp/displayreg/TaskVideos/.

Here we summarize the observations of usability problems highlighted by the participants in the talk-aloud evaluation described above. In performing the first task, all four participants appeared to comprehend the basic principal of the system with only the briefest explanation of how the task should be performed. In each trial, transient registration errors while the user was performing a manipulation tended to cause temporary changes in the manipulated images position, rotation or scale, which participants generally found distracting. Two participants noted that the direction for scale may not be obvious (it was set up so dragging back from the image would make it larger), and that scaling was more difficult than translating or rotating in general. Three of the four participants satisfactorily completed the task of repositioning the images from figure 4 to 5. One had problems that were due to not keeping the mobile phone aimed at the screen itself and this appeared to be because they were observing the PC screen rather than the smartphone itself. In general, it was clear that whenever a transient registration error temporarily affected the plotted cursor point, the final brush strokes would also be affected, further distracting the user. Some disapproving comments on the aesthetics of the green markers were made, and that the relatively slow end-to-end communication speed affected the maximum allowable velocity of the smartphone. Whenever the marker set could not be found (usually due to the phone not being correctly pointed at a screen), the system would timeout and successfully reacquire the marker positions.

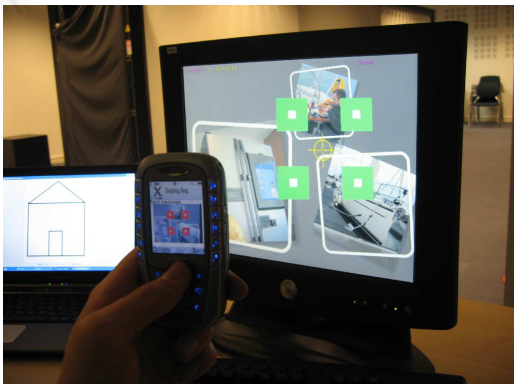

Figure 3: User tests.

6 CONCLUSIONS

We have proposed a new technique for device interaction, which relies on the registration of the display on one device, with an image of that display, captured on another device. This type of interaction opens up a range of new possibilities, in particular those in-

Figure 4: Start position of images on server display.

Figure 5: Target position of images on server display.

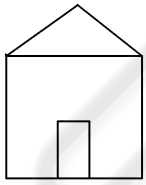

Figure 6: Outline of a template on the server display.

volved with interacting with public displays, using hand-held devices such as smartphones and PDAs. We have build a prototype of such a system which uses coloured markers on the server screen to enable a simple and reliable registration process. We have used this system for translating, rotating and scaling images on a PC screen and for simple drawing applications. Through user evaluations, we have proved that the technique works in principle although further work is required to develop the system for faster frame rates, more robust tracking and to implement a markerless registration process, which would provide a better user experience.

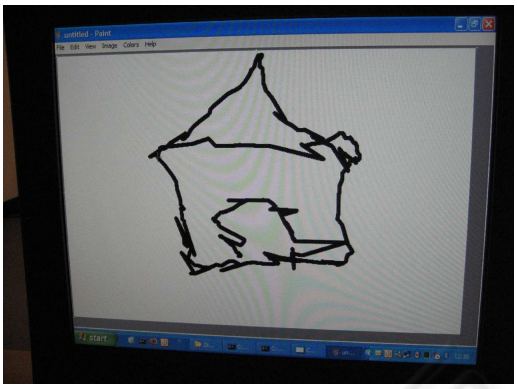

Figure 7: Output of the client motion.

REFERENCES

Alexander, D. (1999). Advances in daylight statistical colour modelling. In *Proc. Conf. Computer Vision and Pattern Recognition*, pages 313–318.

Baumberg, A. (2000). Reliable feature matching across widely separated views. In *Proc. Conf. Computer Vision and Pattern Recognition*, pages 774–781.

Harris and Stephens (1988). A combined corner and edge detector. In *4th Alvey Vision Conference Manchester*, pages 147–151.

Hartley, R. I. and Zisserman, A. (2004). *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition.

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 2(60):91–110.

Schmid, C. and Mohr, R. (1997). Local grayvalue invariants for image retrieval. *IEEE Trans. Pattern Analysis and Machine Intell.*, 19(5):530–535.

Sinclair and Blake (1996). Quantitative planar region detection. *Int. Journal of Computer Vision*, 18(1):77–91.

Smith, S. M. and Brady, J. M. (1995). Susan-a new approach to low-level image processing. *Int. Journal of Computer Vision*, 23(1):45–78.