Abstract

This document details the work involved in the 2004 4th year Un-

teractive Architecture

Artificial Intelligence and Software Engineering

4th Year Project Report

Interactive Architecture

School of Informatics

University of Edinburgh

March 2, 2005

Michael Waller
Tommy French, Ross Denton, Denneen Ademiyone and Nick Johnson.

Camera and projector. Further guidance has also been given by Barbara Webb.

to IAPR (Institute of Pattern Recognition, Action and Behaviour) for the use of their

department of Edinburgh University. For use of their equipment and in particular

beneficial in the development of this project. Thanks also go to the Information

Guidance was given from him. His experience in vision systems has proved greatly

Interactive Architecture was proposed by Prof. Robert Fisher and much of the
3. Implementation Description

2. Walkthrough of Implemented System

1.7 Previous Work

1.6.2 Software Issues

1.6.1 Hardware Issues

1.6 Hardware and Software Restrictions

1.5 Potential Issues

1.4 Proposed System Proposal

1.3 Review of Proposal

1.2 Initial System Proposal

1.1 Introduction: What is Interference Architecture?

0.1 Acknowledgements
1.1 Initial System Proposal

Interactive Architecture?
Reviewing the initial project proposal in conjunction with an investigation of the hardware available highlighted changes that would be necessary to the project.

The main result of this review was that the emphasis of the project changed. Initially the focus was on a fast, easy to use interactive system. Examining the hardware showed that this would not be possible due to large delays on the image capture (as described later). Further it was agreed that user detection and tracking should only be implemented if there was sufficient time, as adding these components would require an additional camera, which was not available. Also their inclusion at a later date would be easy (due to their modular nature) and the increase in terms of functionality was not enough to justify their implementation.

Once it was agreed that the user detection and tracking would be superfluous to the system, the need for the projection of directional arrows was also considered. As the inclusion of this feature would require a second projector (or a mirror-tilt projector both of which were unavailable) it was decided that this component should also be omitted.

Much discussion focused on the initialisation and set-up of the projected image. Aligning the image to the wall so that it appeared correctly whilst being projected from an arbitrary angle could be done either by hand for each installation or set up automatically.

Due to the hardware available the projected menu system should only project 6 options in a 2x3 grid. This could be easily updated for a real-world installation by increasing the number of options or making the menu system hierarchical.

1.3 Revised System proposal

After the above review the focus of the system shifted from fast interaction to automatic initialisation. There are two main driving forces behind this decision. The first is that whilst there has been a fair deal of work done on interactive gesture based systems, there has been relatively little done on automatically aligned camera-projection systems. The second is that, as mentioned above, the hardware in use would not allow for a fast enough response time for novel users.

The new system centers around the visual menu interface. The translations required to establish a correct projection of the menu are calculated automatically with prompts given where necessary. The user is then presented with a menu from which they can select one of 6 options and their choice is displayed. It is easy to see how such a system could be used in other systems and that it would offer
Although the camera and projector are reasonably positioned, there has been a design to accommodate arbitrary positions; this has to be done with heavy varying light conditions.

Although the system has been designed to be both versatile and adaptive, there will be restricted space.

There would be minimal effect as long as the likelihood of any interference is minimal, however with the correct hardware and software the system can deal with a single user environment. In a full installation, this may further limit the interference will be picked up on the camera and disrupt the system. By using a projected mean the system relies heavily on the user's light. Any

Restrictions

In order to further implement the functionality demonstrated the functionality involved whilst not overcrowding the scope of the project, several restrictions have been implemented.

I.4 Potential Issues

set up each time
1.6 Hardware and Software

For the purposes of image capture the system uses a Creative webcam. These cameras can take around 2 frames per second, although access is restricted by the streamer command. The cameras are connected to the computer through a Usb cable, and images were taken at 480x480, although larger sizes are available. There are some temporary controls in regards to brightness and colour levels, however no way of setting them out with xawtv.

The projector that the system uses is the Sony CX-PSD5. This mini projector plugs into the graphics output, and displays images at 900x700 pixels. This is irregardless of the actual size of the image as it would appear onscreen.

All applications and hardware are run from an Informatics’ Dell Workstation running a KDE Linux distribution.

The system runs via Matlab version 6.5.1 installed on the Linux machines provided by Edinburgh University. Matlab is a commonly used mathematical analysis tool, and the majority of the image analysis was done via Matlab’s Image Processing Toolbox. This provides several common functions that were used within this project. See appendix 1 for more on the commands used.

The streamer command line program is part of the xawtv package created by Gerd Knorr. Streamer allows command line control over webcams and is used here to take a series of images to monitor the position of the user's hand, and is also used within the calibration. Within this system the ‘-s’ and ‘-o’ options are used to specify the size and output respectively.

1.6.1 Hardware Issues

1.6.1.1 Camera

The Creative webcam has been a constant hindrance to the system due to its temperamental nature. Often the brightness and saturation of the camera will change without any apparent reason. This threw out many of the processes and accounts for the majority of the anomalies within the system. Consequently the system has been built to withstand this and is both reliable and adaptable.

Also, the camera suffers from radial distortion causing the edges to curve. As this was not noted until fairly late in the development of the system, it has not
1.6.2.1 Mathlab

1.6.2 Software Issues

The previous sections showed that the Mathlab will display only show a small area of the screen, which

The two things that the monitor displays do not change the aspect ratio of the screen. It is not possible to display more than one monitor on the same system. The

The Mathlab will display only one monitor at a time. This means that when using the Mathlab system, the user will only be able to display one monitor at a time.

Here is the difference between this prototype system and a full implementation of

1.6.2.2 Projector

Through the University Cooperative, a Creative WebCam was found, and re-implemented within

The system was altered to allow for this, however middleware that handles the Mathlab images and is not yet available. This would have allowed the

Unfortunately the Mathlab Image Acquisition Toolbox, which allows Mathlab to


draw on the Mathlab's screen, was not available. This would have allowed the

Unfortunately the Mathlab Image Acquisition Toolbox, which allows Mathlab to


draw on the Mathlab's screen, was not available. This would have allowed the
Within the system the 'streamer' command line is used to capture images. Streamer offers two options, the first is to specify a timeline over which multiple images are to be taken. Under Matlab these are all taken before proceeding to the next Matlab instruction. This means that all of the images except the final one would be redundant; the user would have moved on, before the system could process their actions. Alternatively, streamer allows one image to be taken, but then the connection to the webcam is closed. Opening this connection incurs a minimum delay of 1 second each time, and thus the user would have to point for a longer time at their target before receiving feedback.

The solution is sadly to 'hope for the best'. Tests show that users take approximately 2 seconds to point to a menu item naturally, and a survey showed that users would be willing to pause for a further second. The runtime of the pointing-analysis is around 2 seconds. This means that it is likely the user will be on the target as the photo is taken. By using feedback the user will wait longer, as they expect the result of their actions to be displayed.

1.7 Previous Work

The majority of the work done has been on two separate components, 'Gesture based Interactive Menu Systems' and 'Image and Projection Alignment', with a strong bias towards the former.

Initially the idea of using a more complicated interface system was explored and thus considerable reading was done in regards to gesture based interfaces. The attractions inherent in such systems are easily understood: a natural, intuitive control method and the only extra hardware needed is a camera. Initial attempts at implementing gesture recognition revolved around glove-based systems, but as these systems already require a camera it makes more sense to explore the vision based systems. Thus work was done based on the static pose of the hand, however with advances in the field, and greater accessibility due to falling hardware costs, work progressed onto interpreting dynamic hand gestures. Now there are numerous successful gesture recognition systems. [2] [3].

Whilst this success is admirable and could well have been integrated within this project, the scope of the project did not require it, and thus reading was refocused towards image analysis and tracking. This reading would help when developing methods to analyse and track of the user's hand. Paul Rosin's papers on 'Thresholding for Change Detection' [4] were an invaluable resource here, and gave an insight into techniques used to identify adequate thresholds. Further work on
The image alignment for this system, by Rahul Sukthankar et al., is based on the method of "deformed presentations." [9] Form as the basis of the Illusion of the image, this and work done by Ralph et al. [8] reveals.

In "Auto-Calibration of Multi-Profoctor Display Wall," the necessary transformations required to project from one 2D plane to another are necessary. A modern approach [2] to provide information on projective geometry and the problem of scene reconstruction requires this method would, however, allow for a technique for each pixel of the intensity of the function of the probability density function presented. Although the work done by Ahmed Elgammal et al. [5] showed that even in highly complex scenes it was possible to do very successful background subtraction, this isn't...
The first step is to boot up the PC and load the required files. Obviously, in a real environment, this would be loaded on chip and would launch automatically.

Also the projected image should not be greater than 120cm to maintain correct alignment.

When installing the hardware, it is essential that the camera has a full view of the scene, including the entire area which the human is to be projected.

The system is installed.

The calibration procedure will be carried out by the technician that installs the hardware.

The software calibration needs only be carried out once, when

2.1 Calibration

1) Initialize System

2) Take Photo of Wall

3) Take Photo of Camera Alignment

4) Camera Calibration

5) Take Photo of Projection

6) Projector Calibration

7) User Points

8) Direct User to Target

9) Translate to Virtual Point

10) Tran...
The technician then follows the prompts. The first task is to take a photograph of the background (called ‘bg’). Then, following the prompt, the orientation markers should be placed at the corner of the desired menu location and another photograph taken (called ‘camera’). These should then be removed and the projector turned on and used as the display. With the projector calibration image shown, the next photograph should be taken (called ‘projector’). After taking the camera and projector photographs, the system will prompt the technician to check that it has correctly isolated the calibration points. If the system can not identify them, the user should manually select them by clicking on-screen.

With the points selected the system will display a test image via the projector which should be aligned to the wall. A final confirmation is required, before the system proceeds to the Pointing stage.

2.2 Pointing

This is the main stage of the system and will be used as soon as calibration has been completed. Here the user will point at the menu to indicate their desired target. The projection will provide feedback and will display the calculated target, once the user has hovered over it for long enough (about 2 seconds). If the system is incorrect, feedback should be provided to allow the user to compensate.
Isolation and Transformation are explained in greater depth. Describe a mapping from one set to the other. Below the two main components, points are then passed to the transformation method which uses them to find the card bound. This gives you points the midpoints of each piece of card. These four cards are isolated in the scene and the median of the points of each card are then mapped to four known points. Here, four cards are arranged in a square.

The calibration works by identifying four points in the captured image, which are:

- For visualization after each image and after major calibrations.

To calibrate the transformation matrix of the webcam, users are asked to follow the steps. To obtain the transformation matrix, which leads the user through the set-up steps, motivated a right-handed system, which leads the user through the set-up steps. This indicates that when using information is required adequate instructions are given. This part is important that all stages be as automated as possible, and the camera and projector angles are accurately compensated for. In the design of the camera as described above, covers the task of setting up the system so that

3.1. Description

3.1 Calibration
3.1.2.1 Approach

To accurately calculate the transformations, the transformgenerator method requires the position of four points on the captured image which correspond to four previously identified points. These are gained through the same method for both the projector and the camera transformations.

The initialisation procedure requires the user to take three photographs. The first is of the wall or other surface onto which the menu is to be projected. The second is of the same wall with an orientation card or four markers, which will define the area of the menu. The final image is of the wall (without orientation card) with the projector orientation image projected into its correct place.

In both camera and projector isolation stages the same technique is used. First the background is removed via the Matlab command ‘imsubtract’. This subtracts every element in the background image array from the corresponding element in the camera or projector image array and returns an array of the difference. Whilst in theory this would completely isolate the squares, in reality fluctuating light levels and other noise mean that thresholding is required to properly isolate the shapes.

Once these shapes have been isolated each square is classified and subsequently averaged to locate the centre of mass which is then passed to transformgenerator to create the relevant transform.

3.1.2.2 Problems Encountered

Thresholding here was done with the Matlab method 'im2bw' which takes a value between 0 and 1 as the threshold. The precise level of thresholding was difficult to decide as adaptability was strongly desirable. Because of this, the threshold value is calculated within the system, rather than hardcoded. If the squares are not correctly identified on the first attempt, the system prompts the user
The final method proposed is below.

All of these changes created a somewhat complex, although very effective, isol dissection.

Another, but it can be overridden by the user.

By checking that the calibration points are at least 50 pixels away from one check was put in place to ensure that the results were more realistic. This works

Unfortunately the points were still sometimes incorrectly chosen, and a sanity

minor alteration greatly improved results.

thresholded image, and those below a set size were removed from the image. Thus

thresholded image was removed from the image. Thus

the assumed calibration shape. A list of connected shapes was found from the

introduction, command returned background elements as targets. To get rid

of these (and any noise), a method was introduced to remove shapes smaller than

still introduced some errors. If the background or viewpoint has changed slightly,

While the above changes made drastic improvements to the location ability, tests

threshold and calibration resulted in very accurate results.

This combination of

(threshold) with a (digital) pixel spatter. This combination of

computation gap in the calibration shapes. To compute this the image was divided

of shapes. This was often far greater than 4 in tests, as the thresholding had

After the image is thresholded, the polynomial method is used to display the number

is not been adequately identified, then the user is prompted for further guidance.

reach 20% error. If the result is within 10%, the new threshold is increased, 20% away, and this process is continued. The amount

in accordance with the current number of detected objects. Thus if there are 20

if incorrect calculations. By selecting the points via their mouse. Otherwise the
Within this system there are three transformations that are required to properly correct the projected image and interpret the captured image:

\[
\begin{pmatrix}
1 & \theta & \gamma & \eta \\
\lambda & 1 & \mu & \tau \\
x & \eta & 1 & \tau \\
\end{pmatrix}
= \begin{pmatrix}
m \\
m \\
m \\
\end{pmatrix}
\]

Such a transformation \( T \) can be expressed:

By calculating the appropriate transformations the system should be able to

---

3.1.3 Transformation

---

Object

Imaging size of detected object can make it difficult to determine if the control and system automation, a possible improvement would be to set the image size of detected object, the feedback system which allows the user to specify the points detected at all. The feedback system which allows the user to specify the points detected at all. However, when the points are not accurately detected (or not detected at all) the feedback system which allows the user to specify the points detected at all. However, when the points are not accurately detected (or not detected at all) the feedback system which allows the user to specify the points detected at all. However, when the points are not accurately detected (or not detected at all)
\[
\begin{pmatrix}
X_2 & Y_2 & 1 & 0 & 0 & 0 & -X_2x_2 & -Y_2x_2 & -x_2 \\
0 & 0 & 0 & X_2 & Y_2 & 1 & -X_2y_2 & -Y_2y_2 & -y_2 \\
X_3 & Y_3 & 1 & 0 & 0 & 0 & -X_3x_3 & -Y_3x_3 & -x_3 \\
0 & 0 & 0 & X_3 & Y_3 & 1 & -X_3y_3 & -Y_3y_3 & -y_3 \\
X_4 & Y_4 & 1 & 0 & 0 & 0 & -X_4x_4 & -Y_4x_4 & -x_4 \\
0 & 0 & 0 & X_4 & Y_4 & 1 & -X_4y_4 & -Y_4y_4 & -y_4 
\end{pmatrix}
\]

- Wall to Camera.
- Projector to Wall to Camera.
- Projector to Wall.

3.1.3.1 Approach

The first two transformations can be calculated via the points gained in the isolation step described previously. Given four identified points, hereafter referred to as \((x, y)\), and four corresponding known points on the virtual (ideal) image, \((X, Y)\) let \(A\) be the following matrix:

\[
T = \text{null}(A)
\]

Then \(A\) is simply the null value of \(A\) which can be found easily in Matlab via the 'null' command. With more than 4 identifiable points this matrix can be extended to provide greater accuracy, however it was thought that four points would be sufficient.

Using the four identified wall calibration points the above gives the Wall to Camera transform ('cameratransform'). Then using the four projection calibration points the Projector to Wall to Camera transform ('projectortransform') can be found. From these two transformations the more practical transformations can be derived. The mapping from Projector to Wall is given by

\[
cameratransform^{-1} \times projectortransform
\]

This transformation allows us to find the necessary 'prewarp' required to make the image appear correctly after projection. This prewarp is simply the inverse of Projector to Wall mapping.
Although the transformations established here are not new, they have

3.3.3 Review of implementation

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.

been due to its performance within Matlab.
3.2 Pointing

3.2.1 Description

This section describes how the user's interactions are handled by the system. The 'Pointing' section illustrates the success (or failure) of the calibration, as it is only here that the results will be evident. Within the Pointing stage there are two main sections; Isolation and Projection.

3.2.2 Isolation

3.2.2.1 Approach

Although similar to the above method of isolation, here things are somewhat different. Whereas before, the system was dealing with a static image, here frames from a dynamic scene are used. This means that the isolation must be fast and subsequently adaptive thresholding is less suited. Fortunately, here accuracy is not as large an issue, and thus the threshold can be quite high. Further, the matte properties of skin allow for higher contrast.

The translation of the target in the captured image to the virtual target should be simple to achieve via the previously calculated matrices.
3.2.3.3 Review of Implementation

The ceiling and floor when processed.

The processing to each pixel a new image is calculated. This should appear orthogonal to

stereo images. The processing has been calculated, and by applying the transformation

structure the projection via the previously calculated transformations should be

obtained. It was highlighted that the restriction of enforcing the user's position

would allow for a greater variety of shaders and point lighting techniques. This is

the best light that the current system is inheritable, just that there is scope for greater

The thresholds and depth to the captured image is of concern as it is hardcoded.

3.2.3.2 Review of Implementation

performed very well. Identifying the user's interactions from a variety of angles

weather a photo. However, once the thresholds were set appropriately, the system

The problem was somewhat unpalatable, however there was due to a small

solution could be achieved.

As mentioned above, the streamer command line causes a large delay in capturing

images. Considerable time was spent attempting to fix this, however no better
the space, and the angular distortion was completely accounted for. Strangely the base of the prewarped image was occasionally over-warped. This is most likely due to the radial distortion of the webcam.
Projected mean least 200 cm from the center of the projection at all times to enable a usable
longitudinal spread. The camera and projector were kept at
an adequately short distance on the wall. The camera and projector were kept at

For wall calibration points pink square pieces of paper were used and placed in

Appendix 2 shows the results of a set of experiments in which the camera and

4.1 System Results
4.2.1 Isolation

The isolation of the wall calibration squares has been done well, failing only once when the camera was mounted at 40 degrees. This may well be because of a slight slip in the camera’s position during the trials, as the webcam used had no stand. Any significant changes in position or orientation of the camera would alter the image sufficiently that the image subtraction would recognize the parts of the background as new objects. The simplest way to solve this problem, besides an adequate camera mount, would be to threshold and then dilate the original images (background and projector/camera) before performing the image subtraction.

Also, a more precise method of calibration in regards to the projector alignment would be a large improvement. The current system rarely recognizes the projected light patches. This is due to the poor cameras, however the images are still of a high enough quality that a more developed system would be able to isolate the patches.

4.2.2 Calibration

Projector calibration is definitely the system’s strong point. From these results you can see that the system performs well, aligning the display grid almost perfectly in all trials. Although not shown formally, trials were done where the camera and projector were positioned at mirrored angles (the camera at 60, the projector at 150) and the alignment worked well here also.

Although the top angle was well calibrated, the warp applied to the shape also caused deviations along the bottom edge. This gave mixed results, with some projections being wildly distorted along the bottom edge. After investigation and further testing it was shown that this was due to errors with the camera lens which suffers from heavy radial distortion. The best method for bypassing this is to ensure that the camera is vertically aligned with the center of the projection.

4.2.3 Pointing Analysis

The system performed very well identifying the user’s input in most cases. Although a few cases gave unexpected results, these may be down to the sensitivity of the transform matrices. However, judging from the thresholded images it is more likely that they are due to the value of the threshold in the locator method.
4.3.1 Calibration

A component will be revealed before concluding on the system as a whole. In order to decide whether not the system has been a success the individual components will need to be reviewed.

4.3 Conclusion

For any real-world installation, the process of locating the user’s target, and would be absolutely necessary. A better method for calibrating images would have already been tested, allowing for more complete software here has been a major drawback. By using a better quality camera, although noted earlier it is worth stating once again that the hardware and software held their own, a high degree of distance to the wall. Depending on where the system was installed, this close proximity may or may not a factor and the system would likely not be considered to be viable. However, the location method would have to be done at a high level of precision. A better option than having the user’s aim that area with this basic version is to use a projector. A significant issue being a projected menu causes difficulties. Given

4.2.4 Miscellaneous

Another reason for the user to use the menu is to project. As mentioned at the beginning, the menu is correct. The menu can be revised. Adding a calibration step to this would not be overly difficult, and the benefits would be worthwhile.
The importance of a well calibrated menu can not be over-stressed. The entire system is dependant on proper calibration, in order to facilitate the pointing interface. The combination of accurate (and partially automated) isolation and strong corrective prewarp ensure more accurate results.

4.3.2 Pointing

The pointing analysis works incredibly well from a large angle of incidence. Even when the angle is more than 45 degrees from a perpendicular position to the projected image, it can accurately detect the user's intentions. This accuracy is somewhat limited by the distance from the wall, in that the camera's poor resolution causes pixel confusion at long distances. This could be easily rectified through the use of a more modern camera.

As stated earlier the main problem with the pointing method is that the projected menu interferes with the user as they point at their target. Whilst the use of a laser pointer would help to eliminate the problem, such tools detract from the purpose of this project. Ideally the system should be open to all users and require nothing from the user (except themselves). Introducing laser pointers and other such interface devices introduces the possibility of theft and loss, and thus would not be suitable here.

Instead a more precise method of detecting the changes such as those used by Elgammal et al. [5] would perhaps pick up the distortion. If not, the shadow cast by the users arm would provide another potential source for the locator method.

4.3.3 System

Although both Calibration and Pointing are an integral part of the project, they can be used separately. By mounting a projector with a simple webcam and using the techniques described above, the projector could provide automatic keystone correction. The stationary position of the webcam and projector, as well as the increased distance, would provide far better results, eliminating most errors.

The Pointing component could be used in a variety of environments, and would actually work better without the projected menu. One possible use would allow

---

2The walls used in the test scenario were glossy and reflected the projected menu.
The system is not without flaws, most notably the delay caused by streamer in the manner:

"from a user. Then the user can interact with the menu in an easy and intuitive:

a projector and camera (in arbitrary positions) to a screen, with minimal help
a great success. The calibration section will accurately and automatically align
Pointing components but to ensure they worked together. Here this system is
However, the purpose of this project was not only to create Calibration and
4.4.1 Matlab Image Processing Toolbox Commands

'\texttt{im2bw}': This function takes a colour image and threshold as variables, and converts the image to binary based on the threshold value.

'\texttt{bwall}': This function takes a binary image and returns a matrix of the same size as the input image matrix, listing all the connected components as different objects. A second output variable gives the number of objects.

'\texttt{im2dilate}': This function takes an image matrix and dilates the image with a specified shape.

4.4.2 Implemented Methods

'\texttt{doall}': This is the main function, from which all of the others are called. This is what the technician would need to start. Doall takes no variables, and returns no output, simply looping indefinitely.

'\texttt{init}': This function takes two images and identifies the changes, before attempting to locate the four points required to calculate the transformation matrix. This is one of the more advanced functions.

'\texttt{locator}': This function takes two images, and returns the coordinates of the leftmost part of any new objects. Locator is used to identify where the user is pointing, however these coordinates relate to the captured image and must be translated to find the region.

'\texttt{myinput}': This function takes a variety of inputs and translates them from a captured image to the ideal one.

'\texttt{myoutput}': This function translates menu.jpeg into the prewarped image required to properly align with the walls.

'\texttt{region}': This function simply returns the relevant region of the coordinates passed to it from the output of myinput.

'\texttt{takephoto}': This function interacts with the user to confirm that photos taken are correct. This was done to minimise problems caused by the webcams.

'\texttt{transformngenerator}': This function creates the 3x3 transformation matrices required from the points found via init.
<table>
<thead>
<tr>
<th>Projector Position</th>
<th>Camera Position</th>
<th>Distortion</th>
<th>Wall Isolation</th>
<th>Camera isolation</th>
<th>Pointing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Angle X Y</td>
<td>Angle X Y</td>
<td>Angle X Y</td>
<td>Success?</td>
<td>Success?</td>
<td></td>
</tr>
<tr>
<td>90 0 270</td>
<td>90 0 270</td>
<td>0 3.11</td>
<td>Yes</td>
<td>No</td>
<td>1</td>
</tr>
<tr>
<td>65 111 238</td>
<td>65 111 238</td>
<td>0 0</td>
<td>Yes</td>
<td>No</td>
<td>2</td>
</tr>
<tr>
<td>40 167 140</td>
<td>40 167 140</td>
<td>0 1</td>
<td>Yes</td>
<td>No</td>
<td>3</td>
</tr>
<tr>
<td>65 111 238</td>
<td>65 111 238</td>
<td>6 2</td>
<td>Yes</td>
<td>No</td>
<td>4</td>
</tr>
<tr>
<td>40 167 140</td>
<td>40 167 140</td>
<td>6 0</td>
<td>Yes</td>
<td>No</td>
<td>5</td>
</tr>
<tr>
<td>40 167 140</td>
<td>40 167 140</td>
<td>9 0</td>
<td>Yes</td>
<td>No</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Page 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>