The Potential of Fusing Computer Vision and Depth Sensing for Accurate Distance Estimation

Jakub Dostal
School of Computer Science
University of St Andrews, UK
jd67@st-andrews.ac.uk

Per Ola Kristensson
School of Computer Science
University of St Andrews, UK
pok@st-andrews.ac.uk

Aaron Quigley
School of Computer Science
University of St Andrews, UK
aquigley@st-andrews.ac.uk

CHI 2013 Extended Abstracts, April 27–May 2, 2013, Paris, France.
ACM 978-1-4503-1952-2/13/04.

Abstract

Accurately inferring the distance between the user and the interface enables the design of a variety of proximity-aware user interfaces. This paper reports our work-in-progress on designing a toolkit called SpiderEyes that will enable systems to accurately estimate the distance between the user and the interface by fusing computer vision with depth sensing. Potential advantages of this approach include increased accuracy and the ability to reliably estimate the user's distance to the interface even when the user is far from the sensor (up to five metres). We evaluated the feasibility of this approach in a controlled experiment and found that it is likely to yield distance estimations with less than a 10 cm estimation error when users are between 50 cm and 5 metres away from the system.

Author Keywords

distance estimation, proxemics, sensor fusion

ACM Classification Keywords

H.5.m [Information interfaces and presentation (e.g., HCI)]: Miscellaneous.

Introduction

Proxemics is defined as the interpretation of spatial relationships [2]. Proximity-aware applications and devices use the distance between the user and the interface as an interaction modality. This may take the form of continuous interface changes (such as the system Lean-And-Zoom [5]) and discrete interface changes (e.g. using distance to change between explicit and implicit interaction [8]).

There are a number of sensors capable of detecting the distance between a person and a display. For instance, in the past, systems have used WiFi and RFID for coarse-grained distance detection [6] as well as computer vision techniques utilising web cameras and markers [5] or optical motion capture systems such as Vicon [8, 2]. The major disadvantage of these approaches is that they require user augmentation or careful configuration in order to function accurately. Recently, depth sensors such as Microsoft Kinect and readily available computer vision (CV) algorithms have opened up the possibility of designing markerless distance-estimators.

While proximity-aware user interfaces are slowly starting to emerge, they are still difficult for non-specialists to build. Further, while depth sensors such as the Microsoft Kinect are becoming ubiquitous, depth sensing alone has problems in robustly estimating the user's distance to the sensor if the user's body is partially occluded or if the user is outside a particular range (see the subsection on Kinect below for details).

The SpiderEyes Toolkit

We have started building a toolkit called SpiderEyes[1] that will enable non-specialists to easily design and construct accurate markerless proximity-aware user interfaces based on inexpensive off-the-shelf hardware. As part of this toolkit, we are investigating how to best leverage and combine depth sensing (via the Kinect sensor) and computer vision to provide reliable distance estimations.

We believe that sensors requiring user augmentation have many limitations for practical use beyond the controlled environment of a research laboratory. It is technically possible to use other sensors, such as sonar and IR sensors, but these sensors may not be practical because they are not capable of distinguishing between an object and a person, unless deployed in dense grids (see Ward *et al.* for an example [9]). For the purposes of this work-in-progress paper, we focus on approaches that do not rely on user augmentation or an *a-priori* configuration, specifically computer vision and depth sensing.

Depth Sensing - Kinect

The Kinect sensor is an example of a depth-sensing camera. Since it was introduced in 2010, the Kinect sensor (based on PrimeSense sensors) has gathered significant attention in the HCI research community. This is due to the wide range of functionality the sensor offers - including distance sensing, skeleton tracking, gesture recognition and so on. As an example, the sensor has been used by Clark et al. [3] to create a proximity-based interface that allowed different types of interaction with a display at various distances.

The Kinect sensor has a number of characteristics that make it very attractive for research. It is low-cost and offers relatively accurate distance-sensing as well as user and skeleton tracking. The sampling rate is relatively fast (20–30 fps) and the range of recognised distances is practical (80-400 cm[2]). The sensor latency is circa 45 ms.

[1] After release, it will be accessible here: http://sachi.cs.st-andrews.ac.uk/research/software/spidereyes/

[2] source: http://msdn.microsoft.com/en-us/library/hh973078.aspx#Depth_Ranges

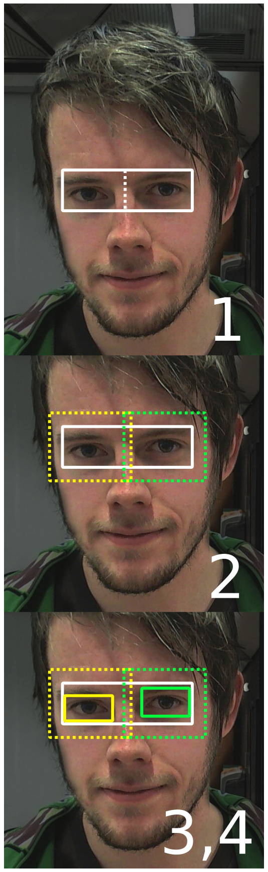


Figure 1: The CV algorithm. 1- finds eye-pairs. 2 - defines search areas for single eyes. 3,4 - finds left and right eye. The pupil distance is the distance between the centres of the bounding rectangles for left and right eyes. Pupil distance is calibrated on a per person basis.

Computer Vision

The computer vision (CV) component of SpiderEyes employs a distance-detector that uses consumer-level cameras that we have developed. As a more advanced version of a previously published algorithm [4], our solution uses the OpenCV implementation of the Viola-Jones feature tracking algorithm [7]. It employs eye-pair and single eye classifiers and a custom tracking algorithm to provide coarse-grained gaze and user tracking. The basis for distance estimation is pupil distance. Due to the minimum pixel size for an eye-pair to be recognised, the maximum detected distance is limited by the resolution of the camera images. The maximum distance that can be detected depends on the pupil distance of the tracked person. For a person with 60 mm pupil distance, using a 5 megapixel (2592×1944 pixels) image taken with a camera with a 62° horizontal field of view, the maximum distance that can be detected is approximately 684 cm.

The distance detector algorithm is scalable. When processing a 5 megapixel image, the sampling rate is up to 25 fps for every single CPU core used once the user is tracked. The sensor latency depends purely on the amount of image processing work, with a latency of circa 800 ms when no user is tracked and a latency of approximately 40 ms once a user acquisition is made (using a single CPU core). When using four CPU cores, the latency decreases to approximately 200 ms with no tracking and approximately 12 ms with tracking. It is important to understand that these are the characteristics of the software part of the system. Actual performance will depend on the camera that is to utilised for tracking. For example, the camera used in our evaluation, Logitech C910, is not able to provide images at more than 10 fps at the 5 megapixel resolution.

Benefits of Computer Vision

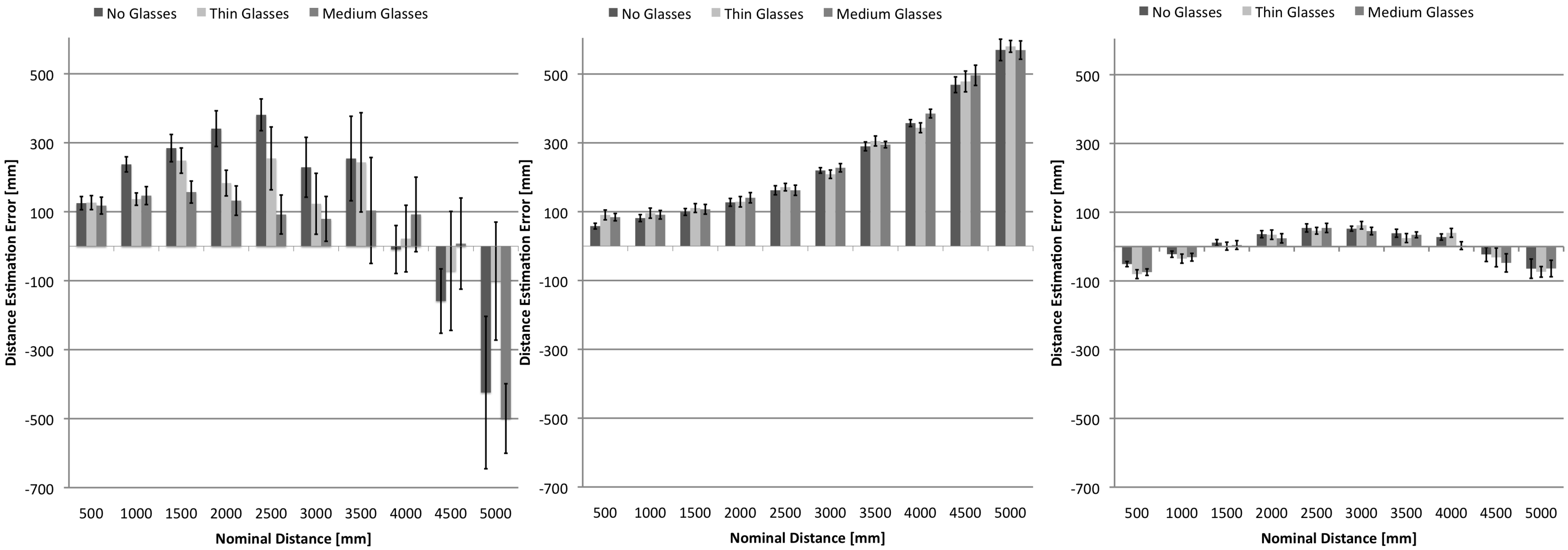
The Kinect sensor has many attractive properties as a distance sensor. The distance of the head can be easily extracted from the skeleton data. However, the sensor also has a number of limitations. First of all, it is severely range limited. The skeleton data can only be provided within the 80-400cm range. In addition, the skeleton data is provided only when the sensor has an unrestricted view of most of the body of a person. Therefore, body occlusion is a serious problem.

In contrast, the SpiderEyes toolkit does not suffer from these limitations to the same extent. Occlusion is generally not a problem because only the eyes need to be visible for the system to work and when a person is looking at a display (and thus at the camera attached to it), the line of sight tends to be maintained. Moreover, SpiderEyes has one characteristic that cannot be matched by the Kinect. Since it detects the eyes when they are looking at the camera, it provides us with information on the direction of gaze as well. In contrast, while the Kinect detects the distance of the head, it is reported as a single point without any indication of directionality.

Fusing Computer Vision and Depth Sensing

The two sensors use different approaches to distance estimation. The Kinect employs an IR camera to capture reflections of a projected IR pattern to estimate distance. SpiderEyes applies computer vision algorithms to ordinary RGB camera images to track the eyes and uses pupil distance to estimate distance.

Even though the two sensors are based on different technologies some of the limitations of the sensors are identical. For both sensors, the spatial resolution decreases exponentially with increasing distance. This is due to the



(a) Computer Vision Only **(b)** Kinect-CV Fusion **(c)** Kinect-CV Fusion–Corrected

Figure 2: Mean distance estimation error for computer vision only, computer vision-guided Kinect, and computer vision-guided Kinect with a pre-computed correction made using a linear regression model. The error bars show standard error.

increasing spread of the IR pattern for the Kinect and the decreasing pupil distance (from the viewpoint of the camera) for SpiderEyes' computer vision algorithms.

The SpiderEyes's Kinect-CV fusion algorithm combines Kinect and Computer Vision data by turning them into a single complex sensor, which overcomes some of the limitations of both of the underlying sensing technologies. We use the CV component to identify the position of the eye-pair. We identify a point in the Kinect depth map and read the depth information from that point to estimate the distance between the user's eyes and the interface. We perform a translation between the coordinate spaces of the two sensors to determine which point in the depth map to read. Using the position of the eye-pair to process raw Kinect depth data, we can make a more specific and more accurate distance estimation, while increasing the possible range of detections compared to the skeleton tracking's maximum range of 4 m (the spatial resolution of the Kinect depth data at 8 metres is still $<$ 20 cm [1]).

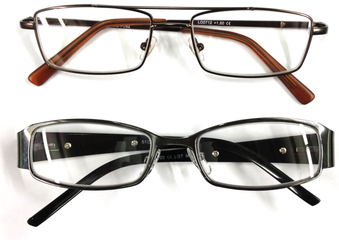


Figure 3: An image of the glasses with highly reflective lenses used. The thin rimmed glasses are shown above the medium rimmed glasses.

As we will see in the next section, the above process results in acceptable accuracies. However, we noticed an exponential over-estimation of this Kinect-CV Fusion technique (Figure 2b). We therefore complement the above fusion strategy with a pre-computed offset correction model that adjusts the overestimation error using a linear regression model. The linear regression correction model is: $y = 0.9005x + 48.411$. Using our experimental data we found that this correction model explains 99% of the variance of the overestimation error ($R^2 = 0.99$). The result of using this correction model on our Kinect-CV Fusion data can be seen in Figure 2c.

Experiment

To evaluate the potential of fusing computer vision and depth sensing we conducted an experiment. We recruited eight participants (three female, ages ranged between 21 and 39) from our university campus. The experiment was a within-subjects experiment with two factors: glasses (participants wearing no glasses, participants wearing glasses with a thin frame, and participants wearing glasses with a thick frame) and sensor (Computer Vision Only, Kinect-CV Fusion, and Kinect-CV Fusion Corrected). Figure 3 shows the two pairs of glasses used in the study.

We positioned the Logitech C910 camera on top of the Kinect sensor. We also marked the floor with distance markers at 50 cm intervals at a range from 50 cm to 5 metres. Each participant was asked to stand with their feet aligned with each of the distance markers, while the study administrator manually read the distance value from each of the sensors. We repeated the process for each participant three times. Each time the participant either wore glasses with thin or thick frames or no glasses.

Figures 2a, 2b and 2c show the distance estimation error for Computer Vision Only, Kinect-CV Fusion, and Kinect-CV Fusion Corrected respectively. In each case, the perfect performance would be represented by a constant error of approximately 5 cm (due to the difference in the position between the tips of the feet of the participants and their eyes).

The estimation profile of Computer Vision Only follows a distinctly different curve to the Kinect. At distances closer than four metres, Computer Vision overestimates the distance, while beyond four metres, SpiderEyes starts to severely underestimate the distance. This behaviour is consistent with the underlying algorithm, where up to four metres, the algorithm achieves high confidence detection using a combination of three different classifiers (the eye-pair, and the left and right eye). Beyond the four metre point, the algorithm can only rely on lower confidence single classifier detection (the eye-pair) because too few pixels capture the individual eyes for the single eye classifiers to work. The accuracy decreases because the precise location of the eyes can no longer be established and the location is instead estimated from the bounding rectangle of the eye-pair.

The Kinect-CV Fusion data was collected by manually reading a distance value from the Kinect depth map. The chosen point was always the pixel in the middle of the nose ridge of each participant. The resulting distance estimation is very stable whether participants are wearing glasses or not. However, there is a clear increase in inaccuracy as the distance increases. While this is partially due to the decrease in spatial resolution, there seems to be a bias towards overestimation that increases with distance. As can be seen in Figure 2c, correcting the Kinect-CV Fusion model using a pre-computed linear regression model substantially reduces estimation errors and results in highly accurate distance estimation. The linear regression model also corrects for the approximately 5 cm constant offset between the tips of the feet when participants stood aligned with the distance marker and the actual distance of the eyes.

Conclusions and Future Work

This paper has reported our work-in-progress on designing a toolkit called SpiderEyes that that fuses computer vision and Kinect depth sensing to accurately estimate user-interface distance. The system uses computer cision to identify the centre between the user's eyes and then reads depth information via a depth map provided by the Kinect. The system then uses a pre-computed linear regression correction model to correct for an exponential increase in over-estimation by the depth sensor as the user is farther away from the sensor. We evaluated the feasibility of this approach in a controlled experiment and found that it is likely to yield distance estimations with less than a 10 cm estimation error when users are between 50 cm and 5 metres away from the system. Further, SpiderEyes appears to be robust for users who do not wear glasses as well as users who wear glasses with thin or thick frames.

The fusion approach described in this paper has not yet been fully implemented. We are currently implementing several versions of the fusion algorithm and working on providing useful programming abstractions that enable non-specialists to easily create proximity-aware user interfaces that can a) reliably sense when users are engaging with an interface and b) determine the user's position from 50 cm to 5 metres away from the sensor with an estimation error that is always less than 10 cm.

References

[1] M. Andersen, T. Jensen, P. Lisouski, A. Mortensen, M. Hansen, T. Gregersen, and P. Ahrendt. Kinect Depth Sensor Evaluation for Computer Vision Applications. Technical report, Department of Engineering, Aarhus University, Denmark, 2012.

[2] T. Ballendat, N. Marquardt, and S. Greenberg. Proxemic Interaction: Designing for a Proximity and Orientation-Aware Environment. In *Proc. ITS*, pages 121–130. ACM, 2010.

[3] A. Clark, A. Dünser, M. Billinghurst, T. Piumsomboon, and D. Altimira. Seamless Interaction in Space. In *Proc. OzCHI*, pages 88–97. ACM, 2011.

[4] J. Dostal, P. O. Kristensson, and A. Quigley. Estimating and using absolute and relative viewing distance in interactive systems. *Pervasive and Mobile Computing*, 2012, doi:10.1016/j.pmcj.2012.06.009.

[5] C. Harrison and A. K. Dey. Lean and Zoom: Proximity-Aware User Interface and Content Magnification. In *Proc. CHI*, pages 8–11. ACM, 2008.

[6] T. Prante, C. Röcker, N. Streitz, R. Stenzel, C. Magerkurth, D. van Alphen, and D. Plewe. Hello.Wall Beyond Ambient Displays. In *Proc. Ubicomp*, 2003.

[7] P. Viola and M. J. Jones. Robust Real-Time Face Detection. *International Journal of Computer Vision*, 57(2):137–154, 2004.

[8] D. Vogel and R. Balakrishnan. Interactive public ambient displays: transitioning from implicit to explicit, public to personal, interaction with multiple users. In *Proc. UIST*, pages 137–146. ACM, 2004.

[9] A. Ward, A. Jones, and A. Hopper. A new location technique for the active office. *IEEE Personal Communications*, 4(5):42–47, 1997.