Distortion Correction of Depth Data from Consumer Depth Cameras

Sean Clarkson[*a], Jon Wheat[a], Ben Heller[a], James Webster[b], Simon Choppin[a]
[a] Centre for Sports Engineering Research, Sheffield Hallam University, Sheffield, United Kingdom
[b] Oxylane Research, Villeneuve-d'Ascq, Lille, France

Abstract

Since the introduction of the Microsoft Kinect in November 2010, low cost consumer depth cameras have rapidly increased in popularity. Their integral technology provides a means of low cost 3D scanning, extending its accessibility to a far wider audience. Previous work has shown the 3D data from consumer depth cameras to exhibit fundamental measurement errors: likely due to their low cost and original intended application. A number of techniques to correct the errors are presented in the literature, but are typically device specific, or rely on specific open source drivers. Presented here is a simple method of calibrating consumer depth cameras, relying only on 3D scans of a plane filling the field of view: thereby compatible with any device capable of providing 3D point cloud data. Validation of the technique using a Microsoft Kinect sensor has shown non planarity errors to reduce to around ± 3mm: nearing the device's resolution. Further validation based on circumference measures of a cylindrical object has shown a variable error of up to 45mm to reduce to a systematic overestimation of 10mm, based on a 113mm diameter cylinder. Further work is required to test the proposed method on objects of greater complexity and over greater distances. However, this initial work suggests great potential for a simple method of reducing the error apparent in the 3D data from consumer depth cameras: possibly increasing their suitability for a number of applications.

Keywords: Kinect, depth camera, distortion, non-planarity, calibration

1. Introduction

Over the past decade there has been an ever increasing interest in 3D vision and its potential applications [1]. The field of 3D scanning has observed considerable growth, in particular the area of 3D body scanning [2]. However, the majority of current 3D scanning systems are very costly and complex [1]. This restricts the potential applications of such technology, and typically limits its use to highly funded research institutes or private companies.

In November 2010 Microsoft launched the Kinect sensor, costing around £80 and selling over 8 million units in the first 60 days of sale [3]. Designed as a motion controller add on for the Xbox 360 games console, the Kinect uses pseudo structured light technology - in conjunction with computer vision techniques - to track the motion of a person. Immediately after its launch, hackers and developers sought to exploit the technology- seeing the potential of low cost 3D scanning [4]. A wide variety of drivers and software development kits (SDKs) are now available for the Kinect, bringing 3D vision and scanning to the masses.

Several low cost consumer depth cameras (CDCs) have been launched since the introduction of the Kinect, designed to support the ever increasing demand for natural user interaction (NUI). Popular devices are the Asus Xtion Pro and the SoftKinetic Depthsense: relying on pseudo structured light and time of flight (TOF) respectively. Like the Kinect, both devices can output 3D scan data in addition to motion tracking data.

The launch of CDCs has generated significant research interest in a wide range of communities including: robotics [5], body scanning [6], healthcare [7], graphics [8] and apparel [9]. Despite their widespread use there is often little cited accuracy. However, a number of studies - focussing on device accuracy - have reported the data from CDCs to lack accuracy [10]. Inaccuracies can be attributed to a number of factors: pseudo random noise, point resolution degrading with distance [11], depth binning errors, and distortion of the depth data [12].

A number of accuracy studies focus on the Microsoft Kinect, and show the 3D data to exhibit significant radially symmetric distortions [12]–[14]. Cui et al [15] showed TOF cameras to exhibit similar distortions, and propose a method of estimating distortion correction parameters. However, their approach is specific to time of flight (TOF) cameras and not applicable to structured light CDCs.

[*] s.clarkson@shu.ac.uk, +44(0)114 225 5590, www.shu.ac.uk/research/cser, www.depthbiomechanics.co.uk

Menna et al [11] investigated distortion of the Kinect's depth data, reporting significant (± 20mm) errors: when scanning a plane filling the field of view (FOV). Smisek et al [13] showed the errors to be consistent with a radial distortion pattern, resulting in a plane appearing convex. The literature suggests the radial distortion pattern to be consistent between devices, but the magnitude of the resulting error to vary considerably [10]. This suggests the problem to be a function of the Kinect's hardware, and the variability due to a factory calibration technique or large manufacturing tolerances. Although currently unreported, this problem is likely apparent with other CDCs (such as the Asus Xtion Pro) which also use the Primesense system on chip (SoC) [16].
As the Primesense reference design is proprietary and closed, there is little available information regarding device calibration. However, it is well reported that devices are calibrated during manufacture and the calibration parameters stored in on board memory [17]. This permits functionality within the scope of the original intended application, but appears inadequate for applications requiring a greater degree of accuracy.

Smisek et al [13] proposed a method of correcting the depth distortion by calculating a distortion correction component from the mean of plane fitting residual errors. Ostensibly, the singular nature of the correction component means its performance will be better in some areas than others. Herrera et al [12] recognised this limitation, and instead proposed a method of calculating a spatially varying offset. Instead of correcting in metric units, Herrera et al corrected correct depth distortion in raw Kinect disparity units, citing this will lead to greater accuracy. The method proposed by Herrera et al indeed performed best, but involved a time consuming process of 2D checkerboard camera calibration and 3D planar calibration. With this information, the radial and tangential distortion of the IR camera was corrected, and a model developed to convert raw disparity to an undistorted metric value.

However, as the technique relies on raw disparity measurements it is device specific and may not be applicable to other CDCs: particularly those based on TOF technology. Furthermore, access to the Kinect's raw disparity measurements is only possible via open source drivers. This poses no problem from a hobbyist or researchers point of view – providing they have no specific requirement to use the Microsoft SDK – but poses problems for commercial developers: as Microsoft's licensing terms dictate their official SDK is used for any commercial applications. The technique proposed by Herrera also has a number of practical limitations. As individual device calibration is required before first use, the process would be very onerous when calibrating the large number of sensors (>6) typically found in 3D body scanning systems. This suggests a need for a method of correcting depth distortion, which is device independent, doesn't rely on proprietary drivers or specific software, and isn't time consuming for the operator.

Presented here is a method of calibrating CDCs in metric units, improving upon the existing work of Smisek et al by calculating a spatially varying correction factor. The proposed technique relies on 3D scans and corresponding depth maps of a plane filling the FOV, providing compatibility with any CDC that outputs 3D point cloud data. Plane scans are captured over the operating range of the device, enabling compatibility with the different operation modes found on many CDCs. The plane scans are used to determine per pixel depth errors across the FOV, and how they vary with distance from the device. These data are used to build a per pixel polynomial distortion model, which is applied to successive depth data.

This study uses the widely available Microsoft Kinect. However, it is important to reiterate the method and results presented here are applicable to a wide variety of CDCs.

2. Methods

2.1 Initial Exploratory Investigation

Primary data showing the nature of the depth distortion across the Kinect's FOV were first collected. The Kinect was setup on a tripod, parallel to a plane spanning the FOV. The plane was moved away from the device in 50mm increments from a distance of 0.6m to 1.5m, figure 1. Custom software based on the Microsoft SDK was used to collect point cloud scans of the plane in each position.

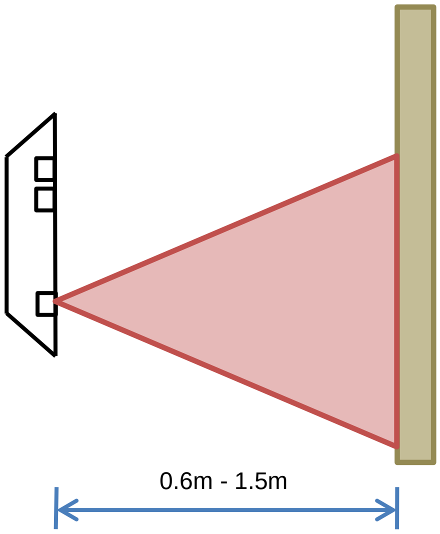

Fig. 1. Experimental setup for the initial exploratory investigation

Principal component analysis (PCA) was used to fit a plane to each scan, enabling analysis of the planar distortion. Figure 2 shows typical error deviation plots, where negative errors represent an underestimation of depth relative to the fitted plane: leading to a plane appearing convex.

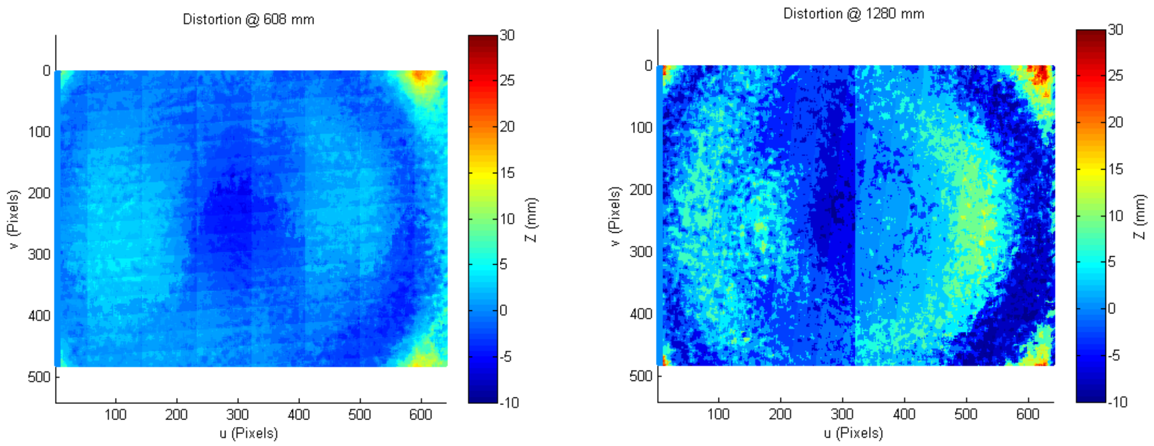

Fig. 2. Typical error deviation plots

The reported radially symmetric nature of the depth distortion is clearly evident in figure 2. The errors are largest at the corners of the FOV: typical of the radial distortion you would expect at the outer extremities of a lens. The radially symmetric error pattern appears consistent across the two depths, but the magnitude of the error appears to increase with depth. This agrees with the literature, and is likely due to the Kinect's depth 'bins' increasing in size with depth.

2.2. Calibration Data Collection

For the purpose of this investigation, calibration plane scans were captured from a distance of 0.60m to 0.90m. This allowed the Kinect to be placed in 'near mode': providing the greatest point resolution. Scans were captured at 10mm intervals - which provides sufficient data to reliably constrain the distortion model - without being time consuming for the operator.

In order to simplify the calibration process a rig was produced, figure 3. This allowed the Kinect to be firmly mounted, ensuring it remains parallel to the moving plane. The rig also indicates the distance between the front face of the Kinect and the plane. This measurement is purely indicative and only assists the operator in ensuring the plane is moved in uniform increments. The Kinect is mounted and calibrated in a vertical orientation. This has no implication on the distortion correction, but best maximises the available space.

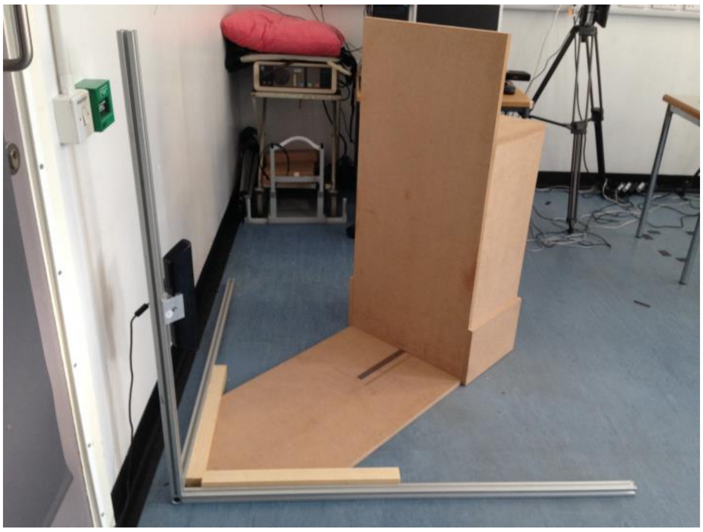

Fig. 3. Kinect calibration rig

For the purpose of this investigation the Kinect's FOV is restricted to 300 x 620 pixels: akin to a FOV for scanning humans with a vertically oriented Kinect. In this case a plane of 0.5m by 1.1m is sufficient to fill the Kinect's FOV across the depth range.

Custom software based on the Microsoft SDK was used to control the Kinect and capture the required data. The software was also used to control the Kinect's IR projector, only switching it on when required to prevent inaccuracies from temperature instability. For each plane position, two sets of data were simultaneously collected: a 3D point cloud scan and a corresponding depth image.

2.3. Calibration Parameter Calculation

PCA was used to fit a plane to each 3D point cloud scan to reduce any errors that may arise from misalignment. Figure 4 shows the typical error residuals from a plane fitting operation, again showing the radial distortion pattern.

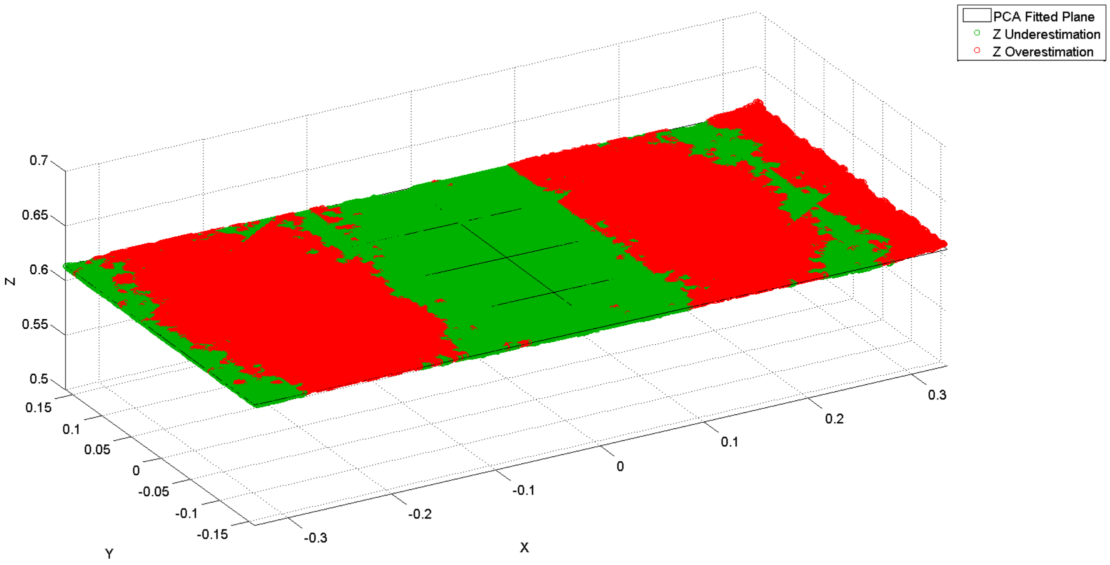

Fig. 4. PCA plane fitting on calibration plane scan

The coordinates of the fitted 3D plane were imported into the Kinect collection software and converted to a depth image (uvZ) via the Kinect's integral coordinate mapping function. This step was important, as it ensured the Kinect's factory calibration parameters were used when performing the conversion. Furthermore, each set of points was converted by the same Kinect used for collection: to allow application of device specific calibration parameters.

Two data sets were produced, one containing depth images of the observed planes and another containing the equivalent fitted planes. To produce full per pixel error deviation data sets, it was important to ensure each pixel had an associated depth. Missing depth data was therefore inserted using a neighbour's interpolation technique.

Figure 5 shows the subsequent image subtraction process, which was used to subtract the fitted plane depth images from the corresponding observed depth image. This produced a set of residual depth values for each plane position, corresponding to per pixel depth errors. Negative values represent an underestimation of depth.

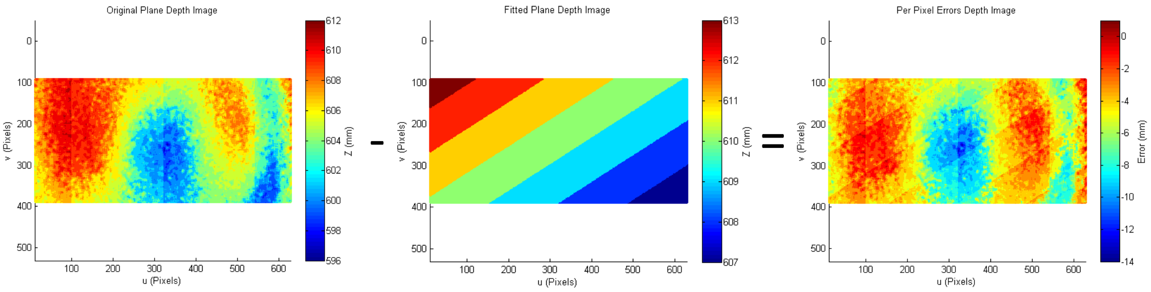

Fig. 5. Image subtraction process and error Deviation plot

Close examination of the original depth image (leftmost plot) shows the top of the plane was slightly further away from the Kinect than the bottom. This meant the plane was slightly twisted relative to the Kinect, and therefore not completely parallel: demonstrating the need to perform the earlier PCA plane fitting process.

The error deviation data was split up into a series of n x 2 arrays, where n is equal to the number of plane positions used, and the number of arrays equal to the number of pixels in the calibrated area. For the region used in this investigation, this equates to 186000 n x 2 arrays. Each array contained the Kinect measured depth and the calculated error deviation at that particular pixel. A second order polynomial was fitted to each n x 2 array, figure 6. This provided a relationship between Kinect depth and depth error for each pixel, valid over the calibrated depth range.

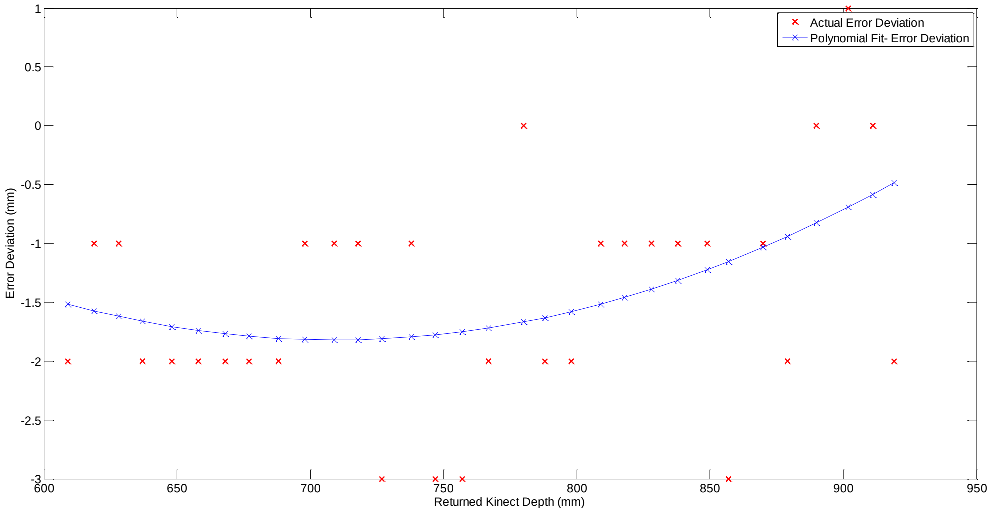

Fig. 6. Typical per pixel error polynomial

Considering these data without merit may suggest a higher order polynomial would provide a better approximation of the error. However, the possibility for variability in the original scan data is high, owing to the pseudo random noise in the Kinect's depth data. By including all of the original data - but fitting a low order polynomial (2nd order) - a high level of smoothing is applied to the data.

2.4. Application of Calibration Parameters

Another custom piece of software - based on the Microsoft SDK - was used to apply the distortion correction to the Kinect data, producing distortion corrected 3D scans on demand.

At the point of calibration, each Kinect was uniquely identified by the serial number on the underside of the device. This ensured a given distortion correction file was unique to a particular Kinect. Unfortunately the Microsoft SDK provides no way of uniquely identifying a Kinect through software. Therefore, a manual intervention was required to ensure the serial number contained within the loaded correction file corresponded to the Kinect in use.

Distortion correction was applied to the scan data via a simple procedure, equation 1. For a given 3D scan the Kinect first returned a uvZ depth image, which directly corresponded to a set of per pixel error polynomials. Therefore, the polynomial corresponding to a given pixel was passed a Kinect depth, and an error value returned. The calculated error value was subtracted from the Kinect depth to produce a corrected depth. This process was repeated for each of the pixels, producing a corrected depth map. The corrected depth map was converted to a 3D point cloud via the Kinect's integral coordinate mapping function.

$$\begin{gathered} Offs = a(Kd)^2 + b(Kd) + c \\ KdCor = Kd - Offs \end{gathered} \tag{1}$$

Where a, b, c are the polynomial coefficients; Kd is the distorted Kinect depth;
Offs is the calculated error; and KdCor is the corrected depth

2.5. Testing Using Random Plane Scans

A single Kinect sensor was first calibrated as detailed above. An additional set of six plane scans were collected at random depths across the calibrated region that were not included in the original training data set. At the point of capture, both distorted and undistorted scans were collected. These two data sets of six plane scans were then used to test the performance of the proposed method.

PCA was used to fit a plane to each of the scans. The projection of each point to the plane, i.e. the residuals in the fit, were analysed to determine the depth error of each pixel. The sum squared error of these residuals (SSE) provides a good measure of the planar deviation, allowing the effectiveness of the distortion correction to be quantified.

2.6. Testing Using Objects of More Complex Geometry

Whilst the testing above evaluates the performance of the proposed method when scanning planes, its performance may be distinctly different when scanning objects of more complex geometry. Further testing using an object with a curved surface – typical of a common body segment - was therefore conducted.

Distortion calibration was repeated on an additional 3 Kinect sensors, producing individual distortion correction files. The Kinects were setup in a similar manner to a body scanning system, focussing on a central point 0.8m away, figure 7. The Kinects were mounted in a vertical orientation to maximise their vertical FOV, with their capture area restricted using a bounding box as before. The Kinects were registered with a global coordinate system, using a modified version of previously presented work [18]. This enabled full 360^{o} scanning within a calibrated capture volume of 0.3m x 0.3m x 0.7m.

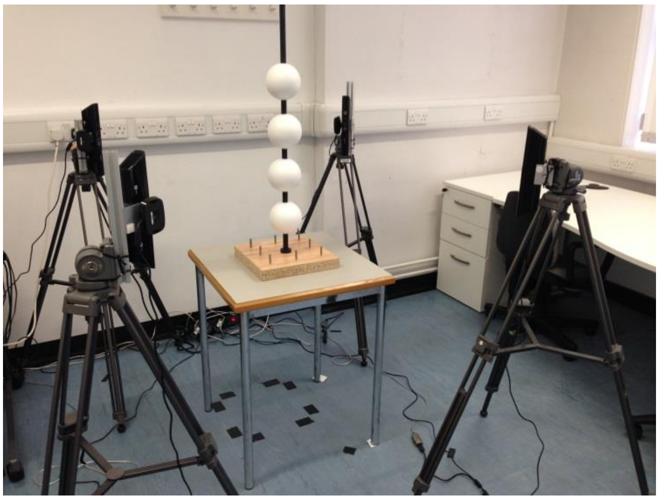

Fig. 7.The four Kinect scanning system

Once setup, the system was used to collect scans of a machined aluminium cylinder, measuring 113mm in diameter and 370mm in length. A total of five scans were collected throughout the capture volume: one scan in the centre, and four in the extreme corners. Care was taken to ensure the centre of the cylinder coincided with the centre of the Kinect's FOV. Distorted and undistorted scan data was collected for each of the locations. Importantly, the global calibration process was repeated using distorted and undistorted scan data. Transformation of individual scans to the global coordinate system was therefore applied via post processing.

Simultaneous data from the Kinect's RGB camera was mapped onto the 3D points, generating coloured point cloud scans. Coloured black bands were applied to the cylinder at the top and bottom, defining a region of interest (ROI). The interface between the black bands and main cylinder body were manually digitised from the coloured point clouds, allowing construction of segmentation planes at the top and bottom of the cylinder. Additional segmentation planes were generated, 10mm apart, and parallel to the bottom plane. This split the cylinder into individual 'slices'.

Each slice was smoothed, and outlying points removed by converting the data to a polar coordinate system and setting exclusion criteria based on theta and rho. The circumference of each slice was calculated: enabling analysis of the circumference change over the length of the cylinder. These data allow the effectiveness of the depth correction algorithm to be analysed over the Kinect's FOV. In contrast, the scans collected from multiple positions allow the effectiveness of the algorithm to be analysed over different depth regions.

4. Results & Discussion

4.1. Random Plane Scan Results

Table 1 shows the SSEs of the fitted plane for the six random plane scans.

Table 1. SSEs of planes fitted to distorted and undistorted plane scans

Plane Distance (m)	0.637	0.678	0.739	0.778	0.818	0.848
SSE Distorted (m)	1.01	1.10	1.27	1.33	1.35	1.90
SSE Undistorted (m)	0.12	0.13	0.16	0.16	0.21	0.24

Ostensibly, the SSEs of the distorted plane scans increase with depth. This doesn't necessarily mean the depth distortion increases with distance, rather that the Kinect's depth 'bins' increase in size with distance. This means there is a greater distance between the depths to which a point can be allocated, hence increasing the possibility - and magnitude - of noise in the original scan data.

The SSEs of the undistorted scans are consistently less than their equivalent distorted scan, indicating smaller residuals in the plane fitting, and hence a better approximation of the original plane. The SSEs are acceptably small over the entire depth range, suggesting the correction algorithm to perform well at varying depths. The undistorted SSEs also increase slightly with distance, and is again likely due to the increasing size of the Kinect's depth 'bins'.

Simply evaluating the performance of the distortion correction by examining the SSEs in plane fitting is not sufficient, as the distortion correction model may perform better in some areas than others. Therefore it is important to visually interpret the distorted and undistorted scan data. Figure 8 shows the scan data set relating to the 0.678m plane distance.

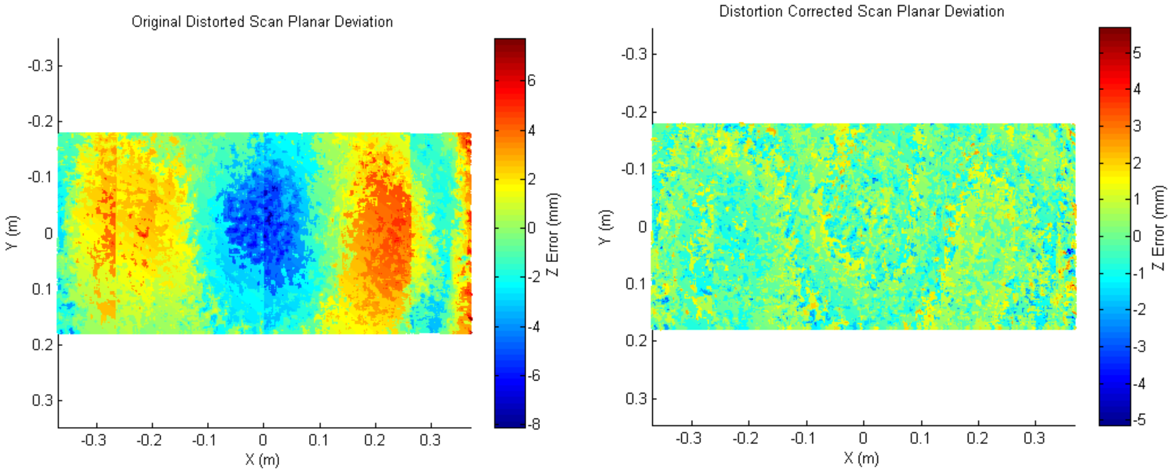

Fig. 8. Planar deviation before and after correction
(Note. Plots are shown on different scales to allow better analysis of trends in the data)

The plot to the left of figure 8 shows the radially symmetric depth distortion in the raw scan data. The plot to the right shows the same scan, but with distortion correction information applied to the original scan data.

The significant reduction in depth error is immediately apparent in the undistorted plane scan. Furthermore, the three areas of high error are not apparent in the undistorted scan data. The majority of planar errors are around zero, and importantly, the errors are randomly distributed throughout the plane: having no clear systematic shape. There are a few larger errors of around -5mm, but are likely related to depth binning errors. Importantly, spurious points such as these can easily be removed with simple smoothing or noise removal techniques.

4.2. Objects of More Complex Geometry Results

Figure 9 and 10 show the deviation in circumference over the length of the cylinder for the distorted and undistorted scans respectively. Tables 2 and 3 show further analysis of the circumference data.

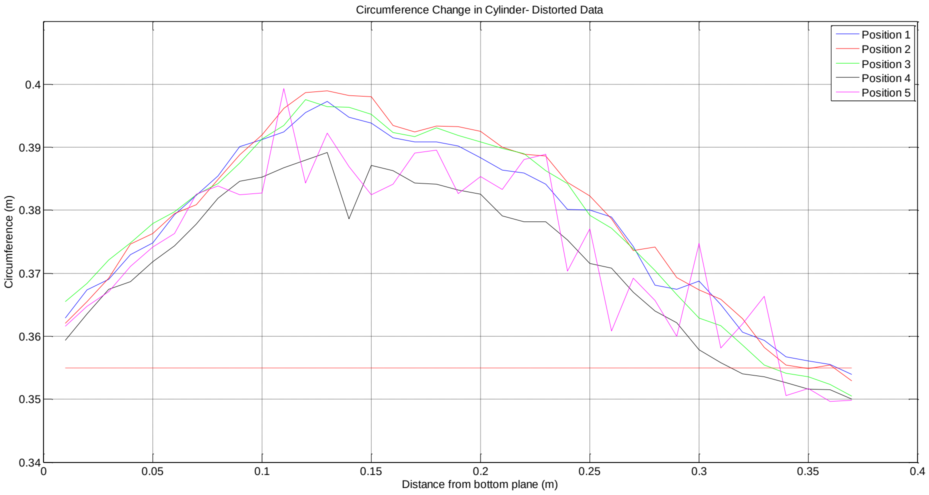

Fig. 9.Comparison of circumference deviation in the distorted scan

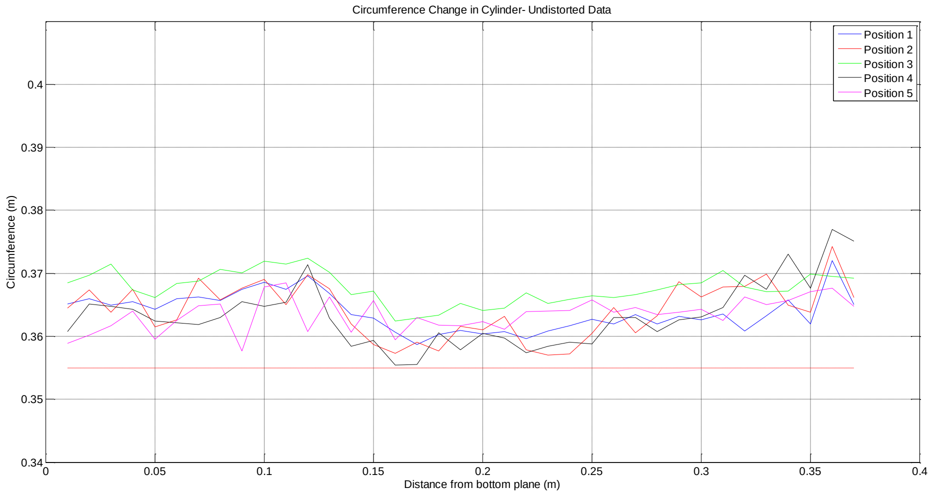

Fig. 10.Comparison of circumference deviation in the undistorted scan

Table 2. Circumference error statistics over the cylinder length (Distorted Scan)

	Position 1	**Position 2**	**Position 3**	**Position 4**	**Position 5**
Mean Error (%)	6.3	6.8	6.5	4.7	5.4
Mean Error (mm)	22.3	24.2	23.1	16.8	19.3
Standard Deviation (Pp)	3.9	4.1	4.1	3.5	3.8
Standard Deviation (mm)	13.7	14.5	14.7	12.5	13.3
Maximum Error (%)	11.9	12.4	12.0	9.6	12.5
Maximum Error (mm)	42.3	43.9	42.5	34.2	44.3
Minimum Error (%)	-0.5	-0.6	-1.3	-1.4	-1.5
Minimum Error (mm)	-1.6	-2.1	-4.5	-5.0	-5.4

Table 3. Circumference error statistics over the cylinder length (Undistorted Scan)

	Position 1	Position 2	Position 3	Position 4	Position 5
Mean Error (%)	2.5	2.6	3.6	2.3	2.4
Mean Error (mm)	8.8	9.1	12.7	8.3	8.5
Standard Deviation (Pp)	0.8	1.2	0.7	1.4	0.7
Standard Deviation (mm)	3.0	4.3	2.5	5.1	2.6
Maximum Error (%)	4.8	5.4	4.9	6.2	3.8
Maximum Error (mm)	17.0	19.3	17.4	21.9	13.4
Minimum Error (%)	1.0	0.6	2.1	0.1	0.8
Minimum Error (mm)	3.7	2.0	7.4	0.4	2.7

Figure 9 shows the raw distorted scan data to exhibit a clear over estimation of circumference towards the centre of the cylinder. This agrees with the previous planar distortion data: indicating a consistent underestimation of depth around the centre of the lens, and hence a larger apparent diameter. This leads to a high intra scan standard deviation in calculated circumference. Importantly, the magnitude and shape of the error is consistent between all five scan positions: indicating the same error characteristics throughout the capture volume.

Figure 10 shows the corresponding undistorted scan data. The effects of radial depth distortion have clearly been significantly reduced, with the large circumference overestimation around the centre of the cylinder being removed. Importantly, this has led to a significant reduction in standard deviation. A systematic overestimation in circumference is apparent, but indicates a consistent error across the FOV. The five scans taken throughout the capture volume show a close agreement of standard deviation and error characteristics, suggesting the algorithm to perform well in different depth regions. The slight difference in error between positions is also evident in the raw scan data, and is likely due to the distance between sensor and object differing for each position and sensor.

The average error in circumference across the 5 undistorted scans is 9.49mm, equivalent to an error of 1.51mm in radius. In contrast, experience has shown the spacing between points in the point cloud to be around 1.5mm at this operating range. If the Kinect's depth quantisation levels consistently floor the calculated depth – and a number of points were therefore incorrect by only a single depth bin - it would be logical to see how a systematic overestimation of the magnitude observed here could arise. This would suggest the errors reported in table 3 were largely related to the Kinect's fundamental measurement resolution, and hence nearing the practical operating limits of the device. Nevertheless, as the overestimation is systematic intra scan - and approximately equal inter scan - a simple offset could be applied to shift the data closer to reality.

5. Conclusions

The method presented here provides a simple and timely method of correcting the symmetrically radial depth distortion found in many CDCs. Unlike the majority of existing work, the method relies only on raw 3D scan data: permitting compatibility with any CDC capable of outputting 3D scan data. Results show the method to perform well when scanning planes, reducing the depth distortion to random errors. Tests using a cylindrical object showed the method to reduce a variable error trend to a systematic overestimation of circumference: regardless of position within the capture volume.

The results presented here relate to the Microsoft Kinect sensor, and are likely directly applicable to other CDCs (such as the Asus Xtion Pro and Fotonic P70) which also use the Primesense SoC. The depth calibration method presented here should also work well with other CDCs which exhibit similar depth distortion characteristics; however, this requires further investigation.

A large proportion of the error discussed here can be attributed to depth binning errors. This may differ between CDCs - but is likely a consistent problem - owing to the conversion from discrete quantisation levels to a metric depth measurement. Little work can be done to avoid this problem, and is a characteristic of the low cost sensors. However, a simple solution is to take multiple scans and average the depth measurements between the scans. This technique could be used at the point of planar calibration, as it would likely reduce the level of noise in the original distortion data and lead to a better calibration. If successful, this would mean calibration scans could be collected with a greater spacing.

The systematic overestimation in corrected cylinder circumference data appears due to the separation of the Kinect's depth points in the operating region used for the testing. Further tests with the Kinect at closer operating ranges would provide smaller point separation, and may lead to a smaller systematic offset. However, the systematic nature of the error means it could easily be negated with a simple offset factor.

The studies conducted here indicate the method to perform well over a range of depths, on objects of different shape and size, and on multiple sensors. However, the tests conducted are very limited. Further investigation is required to verify the performance of the technique over a larger depth range, larger capture volume, and a range of objects with complex geometry. Further investigations are also required with regards stability of the calibration, ensuring the per device calibration parameters are still applicable over a period of time.

From a practical perspective, the calibration rig developed here provides a simple method of collecting the required planar scans. However, collecting the scans over a large distance and repeating for multiple sensors would likely be a time consuming process. Possible improvements would be to calibrate multiple sensors at once. This would require a larger calibration plane, but would lead to a much quicker calibration process. Further improvements lay around automating the calibration process: leaving the plane stationary, and moving the sensor(s) away from the plane via a linear stepper motor.

A significant limitation of this approach is the lack of any constraint between neighbouring pixels, meaning the error in each pixel is treated independently. This can lead to spurious points in the corrected data – which can be easily removed – or two neighbouring pixels having distinctly different distortion models due to noise in the original scan data. Whilst current tests have found no problem with this approach, it is an area requiring further work. Investigations are required around taking blocks of pixels and applying the same error model, based on the average error across all the pixels. As the error pattern is radially symmetric, another possibility would be to apply a radial model to the error, with an associated spatially varying offset.

References

[1] K. Voosen, "3D Vision and the Kinect," in *NIDays Graphical System Design Conference*, 2011, pp. 1–41, London.

[2] N. D. Apuzzo, "3D Human Body Scanning Technologies Overview , Trends , Applications," in *Imagina*, 2011, pp. 1–49, Monaco.

[3] T. Ricker, "8 million Kinect sensors sold in first 60 days," *Engadget*, 2011. [Online]. Available: http://www.engadget.com/2011/01/05/8-million-kinect-sensors-sold-in-first-60-days/. [Accessed: 22-Sep-2013].

[4] Adafruit, "WE HAVE A WINNER – Open Kinect driver(s) released," *Adafruit*, 2010. [Online]. Available: http://www.adafruit.com/blog/2010/11/10/we-have-a-winner-open-kinect-drivers-released-winner-will-use-3k-for-more-hacking-plus-an-additional-2k-goes-to-the-eff/. [Accessed: 20-Sep-2013].

[5] P. Henry, M. Krainin, E. Herbst, X. Ren, and D. Fox, "RGB-D mapping: Using Kinect-style depth cameras for dense 3D modeling of indoor environments," *Int. J. Rob. Res.*, vol. 31, no. 5, pp. 647–663, Feb. 2012.

[6] J. Boehm, “Natural User Interface Sensors for Human Body Measurement,” *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.*, vol. XXXIX, no. B3, pp. 531–536, 2012.

[7] K. Labelle, “Evaluation of Kinect joint tracking for clinical and in home stroke rehabilitation tools,” Notre Dame, Indiana, 2011.

[8] S. Izadi, D. Kim, O. Hilliges, D. Molyneaux, R. Newcombe, P. Kohli, J. Shotton, S. Hodges, D. Freeman, A. Davison, and A. Fitzgibbon, “KinectFusion : Real-time 3D Reconstruction and Interaction Using a Moving Depth Camera,” in *UIST Symposium*, 2011, pp. 559–568, Santa, Barbara, CA.

[9] P. Stampfli, A. Rissiek, R. Trieb, and A. Seidi, “SizeITALY - The Actual Italian Measurement Survey,” in *3D Body Scanning Technologies*, 2012, pp. 261–268, Lugano, Switzerland.

[10] J. Boehm, “Accuracy Investigation for Natural User Interface Sensors,” in *Low Cost 3D Sensors, Algorithms and Applications*, 2011, Berlin, Germany.

[11] F. Menna, F. Remondino, R. Battisti, and E. Nocerino, “Geometric investigation of a gaming active device,” *Proc. SPIE*, vol. 8085, no. XI, p. 80850G(1)–80850G(15), 2011.

[12] D. C. Herrera, J. Kannala, and J. Heikkila, “Joint Depth and Color Camera Calibration with Distortion Correction,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 10, pp. 2058–2064, 2012.

[13] J. Smisek, M. Jancosek, and T. Pajdla, “3D with Kinect,” in *IEEE Workshop on Consumer Depth Cameras for Computer Vision*, 2011, Barcelona, Spain.

[14] J. C. K. Chow and D. D. Lichti, “Photogrammetric Bundle Adjustment With Self-Calibration of the PrimeSense 3D Camera Technology : Microsoft Kinect,” *IEEE Access*, vol. 1, 2013.

[15] Y. Cui, S. Schuon, D. Chan, S. Thrun, and C. Theobalt, “3D shape scanning with a time-of-flight camera,” in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2010, pp. 1173–1180, San Francisco, USA.

[16] Primesense Ltd, “Three Dimensional Scanning using Speckle patterns,” U.S. Patent US/2009/00967832009.

[17] Computer Vision Group- Technische Universitat Mumchen, “Intrinsic Camera Calibration of the Kinect,” 2013. [Online]. Available: http://vision.in.tum.de/data/datasets/rgbd-dataset/intrinsic_calibration. [Accessed: 25-Sep-2013].

[18] S. Clarkson, S. Choppin, J. Hart, B. Heller, and J. Wheat, “Calculating Body Segment Inertia Parameters from a Single Rapid Scan Using the Microsoft Kinect,” in *3D Body Scanning Technologies*, 2012, pp. 153–163, Lugano, Switzerland.