https://doi.org/10.2352/ISSN.2470-1173.2017.15.DPMI-065

Accurate measurement of point to point distances in 3D camera images

Kalpana Seshadrinathan, Oscar Nestares, Yi Wu; Intel Corporation; Santa Clara, CA

Abstract

3D cameras that can capture range information, in addition to color information, are increasingly prevalent in the consumer marketplace and available in many consumer mobile imaging platforms. An interesting and important application enabled by 3D cameras is photogrammetry, where the physical distance between points can be computed using captured imagery. However, for consumer photogrammetry to succeed in the marketplace, it needs to meet the accuracy and consistency expectations of users in the real world and perform well under challenging lighting conditions, varying distances of the object from the camera etc. These requirements are exceedingly difficult to meet due to the noisy nature of range data, especially when passive stereo or multi-camera systems are used for range estimation. We present a novel and robust algorithm for point-to-point 3D measurement using range camera systems in this paper. Our algorithm utilizes the intuition that users often specify end points of an object of interest for measurement and that the line connecting the two points also belong to the same object. We analyze the 3D structure of the points along this line using robust PCA and improve measurement accuracy by fitting the endpoints to this model prior to measurement computation. We also handle situations where users attempt to measure a gap such as the arms of a sofa, width of a doorway etc. which violates our assumption. Finally, we test the performance of our proposed algorithm on a dataset of over 1800 measurements collected by humans on the Dell Venue 8 tablet with Intel RealSense Snapshot technology. Our results show significant improvements in both accuracy and consistency of measurement, which is critical in making consumer photogrammetry a reality in the marketplace.

Introduction

3D cameras that can capture range information (distance to each point in a scene) in addition to color information are increasingly prevalent today. Existing technology to generate 3D images include stereo or multiple cameras (PointGrey Bumblebee, HTC One M8, Fuji FinePix 3D W3, Dell Venue 8/10 with Intel RealSense Snapshot technology), structured light systems comprising projected light patterns and cameras (Intel RealSense, Kinect for Xbox 360), time-of-flight cameras (Kinect for Xbox One) etc. An important and interesting application enabled by 3D cameras is photogrammetry or measurement, where the distance between different points on an image can be measured in metric units [10]. Close range photogrammetry systems used in industrial or other applications can be fairly complex systems and online systems operating in real-time, in particular, suffer from reduced accuracy and may require manual intervention [7, 12]. Proliferation of 3D cameras in the consumer space make possible photogrammetry of objects in consumer camera imagery with interesting applications in interior design, sports photography etc. In particular, the user can specify two points in the image to measure the distance between them such as the height of a person, the height of a skateboard jump, length of a piece of furniture etc. Essential requirements in such consumer applications are accuracy and repeatability of the measurements and fast computation to enable interactivity, which makes this a challenging problem similar to online industrial photogrammetry.

Distance is typically measured by utilizing the range information, along with geometric calibration information, to compute the 3D coordinates of the points specified by the user using triangulation [13]. The measurement can then be computed as the Euclidean distance between these 3D points. However, such a simple approach to measurement suffers from several drawbacks and cannot deal with noisy range values which are quite common, imprecise user inputs (for instance, when the specified points are slightly away from object of interest and land in background areas) etc. The resulting inaccuracy and lack of repeatability when the measurement is performed at different points on an object leads to poor user experience and affects the success of consumer photogrammetry applications. In many cases, the two points specified represent endpoints on an object of interest and all points lying between these endpoints also belong to the object of interest. Further, real world objects that are being measured have a profile that is linear along the measurement axis. We utilize these observations to develop an algorithm that can accurately estimate distance between two specified points. In this paper, we present this algorithm for accurate consumer photogrammetry that can run at interactive rates on a mobile device and also improves consistency and repeatability. We implemented and tested the proposed algorithm on the Dell Venue 8 tablet with Intel RealSense Snapshot technology and the results show significant improvement in measurement accuracy and consistency.

Motivation

Using simple triangulation to measure the distance between two points can result in reduced accuracy and poor user experience in consumer applications. First, range information is quite noisy and this is especially true in passive stereo or multi-camera arrays which suffer from inaccuracies in texture-less and occluded regions [11]. Another important source of error in these systems is the quantization of disparity, which is usually computed at integer (or up to quarter pixel) precision due to computational considerations. Further, windowed correlation based range measurement systems commonly used in stereo or structured light systems often suffer from inaccuracy along object boundaries that correspond to depth discontinuities [11]. This is particularly problematic in photogrammetry applications as measurements are often performed between boundary points on an object. Further, range information may be incomplete due to a number of reasons and interpolation of range measurements can also result in inaccuracies. Finally, user input is often obtained in the form of clicks or touch in consumer applications, which is prone to user error if a point that is one or a few pixels away from the object of interest being measured is specified. This results in points being specified on foreground or background regions adjoining the object of interest, resulting in grossly inaccurate measurements.

Figure 1 shows an image and the corresponding disparity map from a Dell Venue 8 tablet with Intel RealSense Snapshot technology. A typical measurement application will display the color image to the user to select points for measurement. Disparity errors are visible along the boundaries of the target (especially at the bottom edge) and when the user selects points in these regions, they get inaccurate measurements with simple triangulation. Further, if the user attempts to repeat the measurement by clicking on different points on the target (3 length measurements and 3 width measurements illustrated in Figure 1), inconsistent results may be obtained. Disparity inaccuracies are more pronounced under challenging lighting conditions and with lowly textured objects, which are hard to control with real-world consumer photography. Measurement errors tend to increase as the distance of the measured subject from the camera increases [1]. All of these issues motivate the need for a robust, accurate and consistent method for point to point measurement using consumer range cameras to improve user satisfaction with the application.

In many cases, the two points specified by the user lie along the edges of an object that is being measured and the points that lie along the line joining these endpoints also belong to the same object as illustrated in Fig. 1. In this instance, valuable information can be obtained by analyzing the 3D structure of all points that lie along this line. Further, real world objects that are being measured often have a profile that is linear along the measurement axis. For example, when humans are not too close to the camera, the 3D points on the human appear planar in many range measurement systems. This is also true with many commonly occurring rectangular or polyhedral objects in the world such as buildings, furniture, cars, photographs etc. and with many objects in man-made environments [4]. We utilize these observations to improve the accuracy and consistency of measurement. We find linear structures using all the points that lie along the two endpoints selected for measurement using robust Principal Component Analysis (PCA). Measurement accuracy is enhanced by computing the Euclidean distance between the endpoints after projecting them onto these linear structures. One issue with our proposed approach is when a user is attempting to measure a gap between objects such as the arms of a sofa or the width of a doorway etc. In this case, analyzing the 3D structure of points between the two endpoints is counter productive. We also present a method to overcome this issue. A detailed description of our proposed algorithm is presented next.

Description

We describe the details of our algorithm in this section. We first introduce some notation and describe measurement using triangulation. We then describe our algorithm for 3D measurement, which assumes that all points along the two specified endpoints belong to the same object being measured. Finally, we describe a method to determine if the user is trying to measure a gap between

Figure 1. *Image and co-registered disparity map from a Dell Venue 8 tablet. Disparity inaccuracies are visible along the edges of the target board, especially along the bottom edge. These will result in measurement inaccuracies when the user selects points in these areas to measure the dimensions of the target board.*

objects, which invalidates our measurement assumption, and revert to simple triangulation for measurement when this situation is detected.

Notation

We first introduce some notation. Let $I(x,y)$ denote a single or multi-channel color image at pixel location (x,y) and let $R(x,y)$ denote the corresponding range image registered to the color image. Note that we do not assume that the range information is complete which can occur, for instance, when the range image and the color image are acquired from different positions (for example, in a structured light system) or in occluded/non-overlapping regions in stereo camera systems where disparity estimation is indeterminate. Let $p_1 = (x_1, y_1)$ and $p_2 = (x_2, y_2)$ denote two points in the image specified by the user whose distance is desired to be measured. We assume a pinhole camera model for the color camera and that intrinsic calibration information is available in the form of a projection matrix K. For a pinhole camera, the projection matrix takes the form in Equation 1 where (c_x, c_y) denotes the principal point, (f_x, f_y) denotes the focal lengths and s denotes the skew. Many consumer digital cameras have $f_x = f_y$ and $s = 0$. Without loss of generality and for simplicity, we assume that the skew component is 0 in this paper.

$$K = \begin{bmatrix} f_x & s & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \quad (1)$$

Any point in the image $p = (x,y)$ with associated range information $Z = R(x,y)$ is a projection of the 3D point P given by:

$$P = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} (x - c_x)Z/f_x \\ (y - c_y)Z/f_y \\ Z \end{bmatrix} \quad (2)$$

The simplest way to measure the distance between points p_1 and p_2 specified by the user is to compute the Euclidean distance

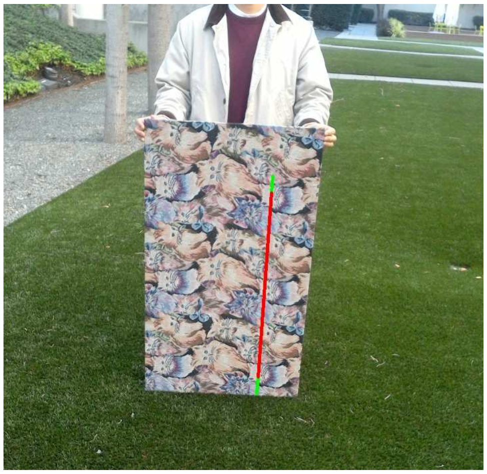

Figure 2. Measurement of an oriented target board along its length.

between the corresponding 3D points P_1 and P_2 computed using Eq. 2. This distance d is defined by:

$$d = \sqrt{(X_1 - X_2)^2 + (Y_1 - Y_2)^2 + (Z_1 - Z_2)^2} \quad (3)$$

As described earlier, this solution suffers from numerous problems and we describe our proposed algorithm for measuring the 3D distance between points p_1 and p_2 next.

Proposed Algorithm

We first sample all pixels along the line defined by p_1 and p_2 and attempt to fit a robust linear structure to these points. To obtain the closest integer coordinates along this line, we sample the image along x if $|p1.x - p2.x| > |p1.y - p2.y|$ and along y otherwise. Depending on the method employed to compute range, 3D coordinates can be obtained for all or a subset of points along this line and we denote these points using $\{L_i = (X_i, Y_i, Z_i), i = 1, 2, \ldots N\}$. Without loss of generality, we assume that $L_1 = P_1$ and $L_N = P_2$. Range information may be unavailable at certain pixel locations (for example, due to lack of texture or in occluded regions in passive stereo/camera arrays) and we ignore these pixels. However, we do assume that range information for p_1 and p_2 is available or interpolated in some manner (for example, median disparity in a window). The locations of the 3D points $\{L_i\}$ are inherently noisy due to errors in disparity estimation, calibration etc.

As motivated earlier, we assume that the 3D points $\{L_i\}$ lie on an object whose 3D profile is linear. PCA can be used to find linear structures in this data. However, traditional PCA is sensitive to outliers in the data and not well-suited to the measurement problem where outliers in the range data are commonly observed. Range estimates from passive stereo systems in particular tend to be noisy at object boundaries, which is often where points are specified for 3D measurement. We propose utilizing robust PCA to address the outliers and solve this problem.

Robust PCA has been studied in the literature and a number of different methods to robustly perform PCA exist [6, 2]. One approach replaces the standard covariance matrix in the PCA formulation with a robustly estimated version. This approach works well on data of small dimensionality (3D in our case) and we adopt this method in our paper for this reason [14]. In particular, PCA can be performed by computing the Eigen decomposition of the covariance matrix of the data, where the first principal component is given by the Eigen vector corresponding to the largest Eigen value and so on. For robust PCA, we compute the minimum covariance determinant (MCD) estimator of the covariance matrix of the data, which is a highly robust estimator of multivariate location and scatter [8]. Its objective is to find h observations out of N whose covariance matrix has the lowest determinant. We utilize the Fast-MCD algorithm to find the MCD solution and this algorithm is also able to detect an exact fit - i.e., when a hyperplane containing h or more observations is present in the data [9]. Our choice of the FAST-MCD algorithm was mainly motivated by its speed of execution allowing for interactive speeds on a mobile device.

Given the data $\{L_i, i = 1, 2, \ldots N\}$ and h which determines the breakdown point of the estimator, the output of the FAST-MCD algorithm is a robust estimate of multivariate location denoted by T and multivariate scatter denoted by S. The algorithm operates by initializing random $h-$ subsets from the data and performing iterations on this data that are guaranteed to reduce the determinant of the covariance matrix [9]. For small datasets, FAST-MCD typically finds the exact MCD and for large datasets, it generates fairly accurate results, although not guaranteed to be the exact result. An important advantage of the Fast MCD algorithm is that it allows for exact fit situations when h or more observations lie on a hyperplane and still yields robust estimates of location and scatter (which is singular in this instance) [9]. The exact fit situation is an important case in our measurement application as this often occurs when the 3D points along the object being measured appear planar, which occurs commonly as described earlier.

We utilize the robust PCA solution to project the 3D points P_1 and P_2 onto the estimated linear structure to generate the predicted 3D points denoted by P_1' and P_2'. Whenever an exact fit situation is detected, the Fast MCD algorithm can also compute the equation of the hyperplane which we denote using $aX + bY + cZ + d = 0$. We first check if $a, b = 0$, which implies that the object has the same range measurement and that the hyperplane is defined by $Z = -\frac{d}{c}$. In this instance, we set the $Z-$ coordinate of P_1 and P_2 to $-\frac{d}{c}$ and re-compute the predicted points P_1' and P_2' using this predicted range measurement in Equation 2. If either a or b is not equal to 0, we project the points P_1 and P_2 onto the hyperplane to compute the predicted points P_1' and P_2'. This can be done using simple geometry by finding the perpendicular projection of a point onto a plane. Equations for computing P_1' are shown below and P_2' is computed similarly.

$$t_1 = \frac{-aX_1 - bY_1 - cZ_1 - d}{a^2 + b^2 + c^2} \quad (4)$$

$$P_1' = \begin{bmatrix} at_1 + X_1 \\ bt_1 + Y_1 \\ ct_1 + Z_1 \end{bmatrix} \quad (5)$$

Whenever a hyperplane is not found, the Eigen decomposition of S can be used to compute the principal components of the data. Let $S = V \Lambda V^T$ denote the Eigen decomposition of S and let v_0 denote the first principal component which is the Eigen vector corresponding to the largest Eigen value. We then project P_1 and P_2 onto this principal component to obtain the projected points P_1'

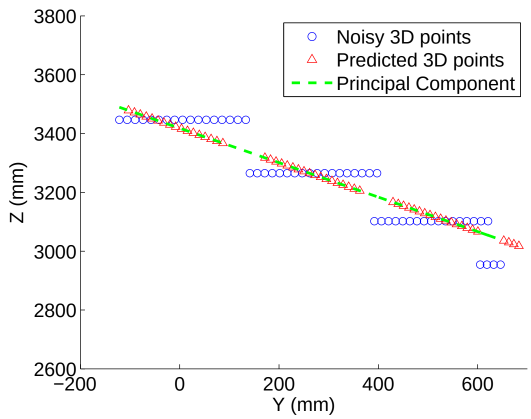

Figure 3. *Plot showing a Y-Z slice of the 3D points resulting from the measurement in Figure 2. Blue circles show the noisy 3D points, green line shows the principal component estimated by our algorithm and the red points show the noisy points projected onto the principal component.*

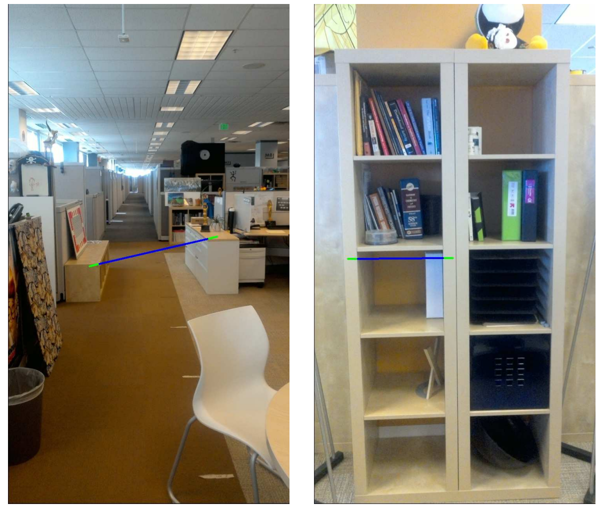

Figure 4. *Blue line shows the line joining two points indicated by user for measurement. Green line segments show the two extensions analyzed to detect gaps.*

and P_2'.

$$P_1' = T + [(P_1 - T) \cdot v_0]v_0 \tag{6}$$
$$P_2' = T + [(P_2 - T) \cdot v_0]v_0 \tag{7}$$

The distance between the two points p_1 and p_2 specified by the user is then given by the Euclidean distance between P_1' and P_2'. Note that our approach is equivalent to performing a robust total least squares (TLS) fitting using a linear model on $\{L_i\}$, which is appropriate since the variables along all 3 directions (X,Y,Z) suffer from error in the measurement application [3, 1].

Figures 2 and 3 helps visualize how our algorithm works. A measurement is performed along the length of the target board shown in figure 2, which is oriented with respect to the camera. Due to the difficulty of visualizing in 3D and since the X- coordinate of this target is almost constant along its length, we show the Y-Z slice of the resulting 3D points along the measurement line as blue circles in Figure 3. Due to integer quantization of disparity, the Z- component of the noisy points lie at four distinct depths. Our algorithm is able to find the principal component of the data points (depicted in green) and the re-projections of each of these points onto the principal component are also shown in red. Our algorithm is able to extract the 3D structure of the target board very well and is hence able to improve measurement accuracy, despite the noisy range data. One interesting point to note is that although the target is planar, our algorithm does not detect a hyperplane in this data and instead fits a 3D line to the data using the principal component. This is due to quantization of the 3D points (particularly along the Z- coordinate due to integer quantization of disparity) resulting in the fact that there is no h-subset of points that lie *exactly* on the oriented plane, which is necessary for the Fast-MCD algorithm to detect a hyperplane.

Detecting gaps

Finally, we need to handle situations where the user measures a gap between different objects as illustrated in Figure 4, which violates our assumption that the points along the line joining the endpoints belong to the same object. We attempt to detect these situations when they occur and revert to basic triangulation to compute the measurement in these instances. We utilize the intuition that when the user is measuring a gap, the two points specified for measurement are *in front* of the points that lie along the line joining them. We construct a fixed-length line segment from P_1 along the line P_1P_2 and away from P_1 as illustrated in Figure 4 in green. We construct a similar line segment for P_2. Whenever the two line segments appear to be in front of P_1' and P_2' respectively, we conclude that a gap has been detected. To do this, we compute a robust estimate of range Z_1^{ext} from the range measurements in line segment near P_1 using the bisquare weighting function and similarly for Z_2^{ext} [5]. If $Z_1^{\text{ext}} < Z_1'$ and $Z_2^{\text{ext}} < Z_2'$, we conclude that the measurement endpoints represent a gap and we abandon refining the measurement between p_1 and p_2 using our proposed method. Note that while we could have used robust PCA to determine the depth of the line segments too, we use the simpler approach of robustly fitting the range due to the small number of data points analyzed.

Results

We tested our algorithm on the Dell Venue 8 tablet with Intel RealSense snapshot technology. A depth map is created for every image captured on the tablet using the three cameras on the system. A measurement application allows the user to select any two points in an image using a touch interface and the computed measurement is displayed to the user. Our proposed algorithm was implemented and run on the Intel Atom Processor (Dual Core) on the tablet. The runtime of the algorithm is variable and depends on the data. Exact fit situations are detected very quickly and the worst case runtime was measured to be 55ms on images of resolution 1280x720, which is well within requirements for interactive applications.

We tested our algorithm on > 1800 measurements of the length and width of a textured poster and the height of a human on the 3D image captured by the Dell Venue8 tablet. A sample set of seven measurements (3 each along the width and height of the checkerboard and 1 of the human) is illustrated in Figure 1. The

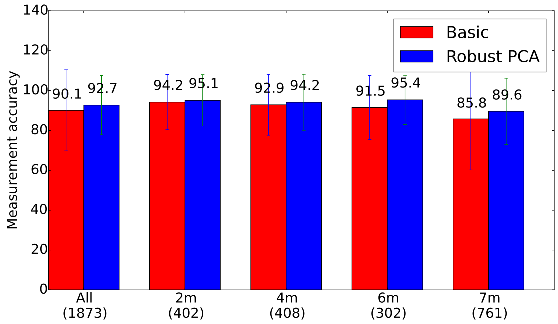

Figure 5. *Measurement accuracy using our method. The "Basic" algorithm uses the median disparity in a window centered around each point for triangulation. The numbers in parentheses under each bar represent the number of data points in each category.*

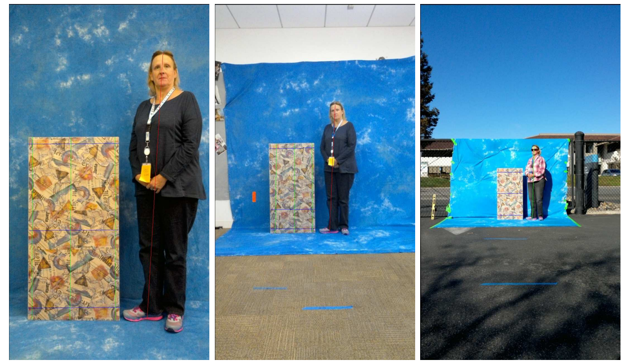

Figure 6. *Test images at different distances of the target and under different lighting conditions.*

points used for measurement were manually clicked by human users on the Dell Venue8 tablet device. The data was acquired at different distances of the target from the cameras (typically at 2m, 4m, 6m and 7m) and under different lighting conditions (indoor, mixed lighting and outdoor conditions) to validate its performance in the real world. Some examples from our dataset are shown in Figure 6.

For a ground truth distance of g and a corresponding measurement m, we compute measurement accuracy using:

$$A = 100 - 100\left(\frac{|g-m|}{g}\right) \quad (8)$$

The results of using our algorithm to compute measurement accuracy is illustrated in Figure 5. For a fair comparison and to account for noisy disparity values, we compare against a method that computes the median disparity in a window surrounding each point and uses this value to determine the Euclidean distance between the points. Our method achieves a 2.6% overall improvement in mean accuracy with a 5% reduction in standard deviation. Accuracy is also shown separately for different distances of the target from the camera (2,4,6 and 7m) and it is seen that the accuracy improvement using our proposed algorithm increases as the distance of the targets from the camera increases. At 6m and 7m, mean accuracy improves by approximately 4%. This is to be expected since the accuracy of measurement decreases with increasing target distance due to the inverse relationship between disparity and depth.

We tested the gap detection algorithm qualitatively on a few images by attempting to measure gaps and observed good performance. For a more quantitative evaluation of false positives produced by the gap detection algorithm, we tested it on this data set which does not contain any gaps. We set the length of the extension line segments to 25 pixels. Only 6.8% of 1873 measurements were detected as gaps which represents reasonable performance of the gap detection algorithm. Most erroneous situations where gaps were detected (81%) occurred at target distances $\geq$ 6m when the disparity was noisy.

We need to explicitly select the parameter h, which determines the breakdown point of the Fast MCD estimator, in our algorithm. Different range measurement systems have different accuracies and these need to be taken into consideration in selecting h, which directly corresponds to the expected number of outliers in the system. In our experiments with the Dell Venue 8 3-camera system, we set $h = 0.5N$ since we observed that contention between the two camera pairs used in disparity estimation often resulted in upto 50% outliers. For multi-camera systems, h can be increased as the number of cameras increases since disparity accuracy is expected to improve. This is also true of structured light or time-of-flight systems that use active illumination to achieve better range measurement accuracy.

Finally, we would like to note that we also tested the simpler approach of fitting a robust line to the depth values (or just the Z- coordinate of the 3D point), rather than modeling the 3D points themselves. While this approach does better than the basic algorithm (mean accuracy improved by 0.6% on all the data), it does not reach the level of performance of our proposed algorithm which is better able to model the 3D structure of different objects in the real world.

Conclusions and Future Work

We presented a robust and accurate method for point-to-point measurement from 3D camera images in this paper. Our algorithm runs at interactive rates (worst case runtime of 55ms) on a mobile device. Our method was tested using images collected under challenging conditions for the algorithm and found to improve both the accuracy and consistency of measurement applications significantly. We believe that these improvements are essential to make consumer photogrammetry perform to the level of expectation of consumers and succeed in the marketplace. One aspect of our work that can be improved upon is the gap detection algorithm we use, whose performance can be improved using more complex 3D scene modeling. We would also like to extend our algorithm to situations where a confidence map associated with the range information is also available. Finally, we would also like to extend some of these ideas to photogrammetry and geometric modeling of 3D videos.

Acknowledgments

We would like to thank Kathryn Feld, Suresh Nampalli, Srini Gosangi and other members of the team for endless hours spent collecting measurement data, which was used in validating our algorithm.

References

[1] Steven D. Blostein and Thomas S. Huang. Error analysis in stereo determination of 3-D point positions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(6):752–765, 1987.

[2] Emmanuel J. Candes, Xiaodong Li, Yi Ma, and John Wright. Robust principal component analysis? *Journal of the ACM*, 58(3):1–37, 2011.

[3] G. Golub and C. van Loan. An analysis of the total least squares problem. *SIAM Journal on Numerical Analysis*, 17(6):883–893, December 1980.

[4] Dirk Holz, Stefan Holzer, Radu Bogdan Rusu, and Sven Behnke. Real-time plane segmentation using RGB-D cameras. In *Robot Soccer World Cup*, pages 306–317. Springer Berlin Heidelberg, 2012.

[5] Peter J. Huber and Elvezio M. Ronchetti. *Robust Statistics*. Wiley, 2009.

[6] F. De la Torre and M. J. Black. Robust principal component analysis for computer vision. In *IEEE International Conference on Computer Vision*, volume 1, pages 362–369 vol.1, 2001.

[7] Thomas Luhmann. Close range photogrammetry for industrial applications. *ISPRS Journal of Photogrammetry and Remote Sensing*, 65(6):558–569, November 2010.

[8] Peter J Rousseeuw. *Mathematical Statistics and Applications, Vol. B*, chapter Multivariate estimation with high breakdown point, pages 283–297. Dordrecht: Reidel, 1985.

[9] Peter J Rousseeuw and Katrien Van Driessen. A fast algorithm for the minimum covariance determinant estimator. *Technometrics*, 41(3):212–223, 1999.

[10] Giovanna Sansoni, Marco Trebeschi, and Franco Docchio. State-of-the-art and applications of 3D imaging sensors in industry, cultural heritage, medicine, and criminal investigation. *Sensors*, 9(1):568–601, January 2009.

[11] D. Scharstein, R. Szeliski, and R. Zabih. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. In *IEEE Workshop on Stereo and Multi-Baseline Vision*, pages 131–140, 2001.

[12] Luhmann Thomas, Robson Stuart, Kyle Stephen, and Boehm Jan. *Close-Range Photogrammetry and 3D Imaging*. e Gruyter, 2013.

[13] Emanuele Trucco and Alessandro Verri. *Introductory Techniques for 3-D Computer Vision*. Prentice Hall PTR, 1998.

[14] H. Xu, C. Caramanis, and S. Sanghavi. Robust PCA via outlier pursuit. *IEEE Transactions on Information Theory*, 58(5):3047–3064, May 2012.

Author Biography

Kalpana Seshadrinathan received the B.Tech. degree from the University of Kerala, India, in 2002, and the M.S. and Ph.D. degrees in electrical engineering from the University of Texas at Austin, USA, in 2004 and 2008, respectively. She is a Senior Research Scientist with Intel Labs in Santa Clara, CA. She is a recipient of the 2013 IEEE Signal Processing Society Young Author Best Paper award and the 2003 Texas Telecommunications Engineering Consortium Graduate Fellowship Award.

Oscar Nestares received his M.S. (1994) and Ph.D. (1997) in Electrical Engineering from Universidad Politecnica de Madrid. After that he was a Fulbright Visiting Scholar at Stanford University and consultant at Xerox PARC (1998-2000) and a tenured Research Scientist at the Institute of Optics, Spanish National Research Council (2000-2003). In 2003 he joined Intel Labs focusing on statistical approaches to video enhancement (de-noising, super-resolution, stabilization), its applications to consumer electronics, workload analysis, and computational photography.

Yi Wu received her Ph.D. in Electrical and Computer Engineering from the University of California, Santa Barbara in 2005. She is a Senior Research Scientist with Intel Labs in Santa Clara, CA. She owns multiple US patents and and has published over 40 technical articles. She has also served as a program and organizing committee member of 7+ international conferences and workshops on multimedia, video and image processing.