Freehand Laser Scanning Using Mobile Phone

Ron Slossberg
ronslos@tx.technion.ac.il
Aaron Wetzler
twerd@cs.technion.ac.il
Ron Kimmel
ron@cs.technion.ac.il

Technion Israel Institute of Technology
Haifa, Israel

Abstract

3D scanners are growing in their popularity as many new applications and products are becoming a commodity. These applications are often tethered to a computer and/or require expensive and specialized hardware. In this note we demonstrate that it is possible to achieve good 3D reconstruction on a mobile device. We describe a novel approach for mobile phone scanning which utilizes a smart-phone and cheap laser pointer with a cylindrical lens which produces a line pattern attached to the phone using a 3D printed adapter. Non-linear multi-cale line filtering is used to detect the center of the projected laser beam in each frame with sub-pixel accuracy. The line location coupled with the estimated phone position and orientation in 3D space, obtained from publicly available SLAM libraries and marker tracking, permits us to perform a 3D reconstruction of a point cloud of the observed objects. Color and texture are extracted for every point along the scanned line point by projecting the reconstructed points back onto previous keyframed images. We validate the proposed method by comparing the reconstruction error to the ground truth obtained from an industrial laser scanner.

1 Introduction

3D sensing has become an integral part of everyday life as related consumer products are becoming increasingly available each year. These devices utilize many different technologies such as structured light, time of flight and stereo to accurately reconstruct observed surfaces and objects. Common traits among these devices are that they are relatively expensive and often require specialized hardware as well as demanding tethering to a wired computing platform (such as Rubinstein *et al.* [12], Acosta *et al.* [2] and Surmann *et al.* [14]).
In addition to 3D sensors based on specialized hardware, several recent papers such as Kolev *et al.* [8] ,Tanskanen *et al.* [15] propose various ways to reconstruct 3D scenes using hardware available onboard a standard "smart-phone". Such research has led to commercial applications such as [1, 13, 16] which are widely available. These methods, however, cannot compete with the specialized hardware scanners in quality and often require large servers to carry out computations which render them useless in the absence of a network connection. Additionally, multi-view stereo methods rely on finding correspondences between different

Pages 88.1-88.10
DOI: https://dx.doi.org/10.5244/C.29.88

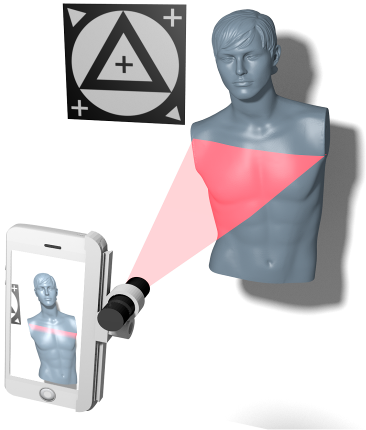

Figure 1: Illustration of the scanning process with a mobile device.

views, a task which is sensitive to texture, lighting and other environmental factors. In this paper we propose a method which is both cheap and simple, requiring only minimal hardware and is able to achieve results comparable to some hardware scanners on a completely autonomous system. Previous efforts such as Bouguet and Perona [5] and Winkelbach *et al.* [17] have popularized the use of amateur hobbyist scanner setups, however having a static camera is often a disadvantage. Our proposed device utilizes a laser beam as described by Winkelbach *et al.* [17], but differs from previous setups by having the laser device and camera attached to each other thus forming a complete hand-held scanner for freehand 3D reconstruction.

As a solution to current limitations the proposed contribution is a straightforward approach matched to mobile constraints that can perform 3D reconstruction at a sufficient level of usability. Figures 1 and 6 illustrate the design the proposed system.

The proposed scanner is based on the ability to detect the location and orientation of the mobile device during the scanning process. This stage is crucial because otherwise there would not be a way to integrate the images viewed by the camera into a common global coordinate system. We take advantage of a combination of OpenCV Library [6] and PointCloud SDK [1] which is capable of tracking marker images and feature points (SLAM) from a live camera feed. The advantage in using a real world marker to initialize the coordinate system is that the resulting point cloud is scaled to real world coordinates. The scanner uses an off the shelf laser pointer which generates a focused laser line. This line is extracted every frame using a non-linear multi-scale filtering process in real time. We speed up the computationally demanding extraction process by using the available NEON vector processor available on the iPhone. Once the laser line is determined with sub-pixel accuracy we calculate the spatial position of the line based on our knowledge of the laser beam's plane in the camera coordinate system. As we will show later, accurate knowledge of the laser plane relative to the camera is critical to the reconstruction process. We extract it after using a calibration process that is designed to be applicable for use on a mobile device in regular settings such as in home or office environments. Once calibrated the system can be used in a fairly general setting.

The scanning process provides real time feedback thereby requiring only a few seconds to acquire a scene. The density of the scan depends on the camera frame rate and the rate at which the user moves the device. It is capable of scanning static objects and scenes and enables continuous update of the scanned object.

2 Laser line extraction

The seemingly straightforward process of extracting the laser line as observed by the mobile device's camera is in fact non trivial. Various approaches were attempted such as simple thresholds on the *RGB* channels of the image and decision tree stumps for pixel classification. These methods failed to extract the laser line with the required accuracy and a more general technique was deemed necessary. Our proposed laser line extraction method is based on the work of Matiukas and Miniotas [10] and on Koller *et al.* [9] who use multi-scale non-linear image filters to search for curvilinear structures in an image. Using these methods combined with polynomial fitting, we are able to extract the laser line with sub-pixel accuracy.

2.1 Multi-Scale Non Linear Filtering

The first stage in our laser detection scheme is the use of the multi-scale line filtering method described by Koller *et al.* [9]. The filter is designed to have a strong response to curvilinear structures. Each filter is comprised of two parts which are first order derivatives of shifted Gaussian Kernels,

$$\begin{aligned} E_L(x,y) &= -\tfrac{d}{dx} G_\sigma(x+s,y), \\ E_R(x,y) &= \tfrac{d}{dx} G_\sigma(x-s,y). \end{aligned} \tag{1}$$

Here G_σ denotes the Gaussian kernel with standard deviation σ. E_L and E_R denote the left and right filters respectively. For simplicity we use $s=\sigma$. We combine the outcome of both filters in the following manner:

$$\begin{aligned} &F(x,y) = \min\left[pos\left\{(E_L \otimes I)(x,y)\right\}, pos\left\{(E_R \otimes I)(x,y)\right\}\right], \\ &s.t. pos\{u\} = \begin{cases} u, u>0 \\ 0, else \end{cases} \end{aligned} \tag{2}$$

The intuition behind this approach is that the filter was engineered to detect bar like structures in an image I. In order to detect these patterns with rotational invariance, we use 3 filters which are rotations of the original filter as given in (**??**) and evaluate their response by taking the maximal response per pixel. For detecting laser lines of varying width, we use 4 different scales for the filters with varying σ. In total we apply the filter 12 times for the 3 rotations and 4 scales to achieve our rotation and scale invariant line detection.

2.2 Sub-Pixel accuracy

In order for the scanner to be as accurate as possible we need to find the center of the laser line. For this purpose we find a 1D quadratic function using LS for each suspected ridge pixel. The laser plane is always positioned so that the projection of the line onto the mobile device's screen is approximately horizontal when the device is held in portrait mode. We therefore search for local maxima in the filter response along image columns. and do this by fitting a $3rd$ order polynomial with form $ax^3+bx^2+cx+d=0$ to the neighborhood of each pixel along it's column.

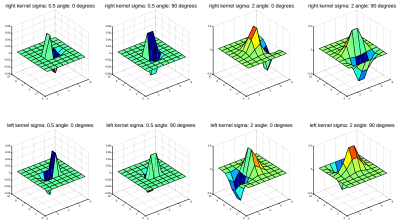

Figure 2: The non-linear multi-scale gaussian kernels used to detect the laser line. We use 4 different scales and 3 rotations.

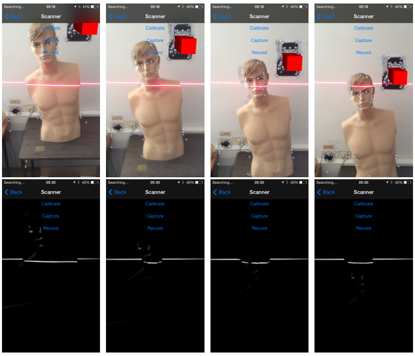

Figure 3: Above: Scenes captured during scans where the laser line is visible. Below: Response of filters to the images with the laser line running on a mobile device.

We first apply a threshold to the filter response that was empirically found to eliminate most pixels which are not part of laser line. For all our experiments this threshold was set at 85 on the red channel. The remaining pixels are all candidates for laser center lines. The candidate with the maximum response is selected together with it's four vertical neighbors and we use them to find a, b, c and d such that:

$$\begin{pmatrix} -8 & 4 & -2 & 1 \\ -1 & 1 & -1 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 8 & 4 & 2 & 1 \end{pmatrix} \begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix} = \begin{pmatrix} F(u, v-2) \\ F(u, v-1) \\ F(u, v) \\ F(u, v+1) \\ F(u, v+2) \end{pmatrix}, \tag{3}$$

where u and v are discrete image locations. Using (3) we can calculate the zeroes of the first order derivative of the polynomial and check the second order derivatives for the maximum condition. We check that the location of the maximum is sufficiently close to the central pixel in order to be verify it's validity. Each point must reside within the pixel boundaries. We allow for a maximum detection distance of 0.5 pixels from the pixel center. The resulting optimal location is denoted by $(u^*, v^*) = (u, v + \Delta v)$ where Δv denotes the offset of maximum from the position of the pixel which fulfills the aforementioned conditions.

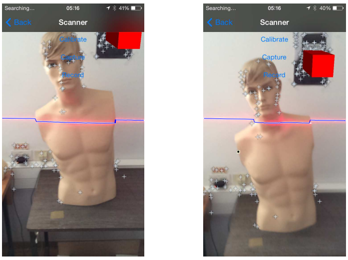

Figure 4: The center of the laser line is traced in blue as detected with sub-pixel accuracy.

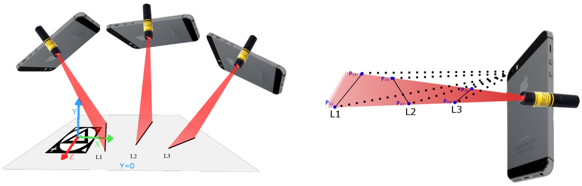

Figure 5: Top: We place the marker on a flat surface and shine the laser on that surface from several angles and distances. Using the gathered points we find the approximation of the laser beam plane. Bottom: The captured points along straight lines from various camera positions as viewed in the camera coordinate space.

3 Calibration

An image taken by a camera is effectively a 2D projection of the world onto the image plane which is represented by a perspective transformation in the pinhole camera model. Without depth information about a point, this transformation is irreversible. In order to find the 3D location of a laser point we solve this problem by finding the intersections between each pixel's reprojected ray and the plane formed by a laser pointer projecting a line pattern. In order to carry out this calculation we must first find the plane that the laser beam creates in camera space coordinates. We next define the calibration procedure which is essential for our reconstruction process and is focused around determining the plane of the laser relative to the mobile device's camera. In addition, intrinsic camera calibration is required. This process provides the intrinsic camera matrix and lens distortion parameters which are required for the reconstruction process. Here we use the calibration routines supplied by OpenCV [6] which use a checkerboard pattern for this purpose.

3.1 Calibration process

The calibration process is carried out by placing an image marker on a flat surface as shown in Figure 5 and using PointCloud SDK [1] to track the marker in order to obtain the relative device position. We then shine our laser beam onto the surface while simultaneously capturing the image of the laser line as illustrated. The laser points are extracted using the method of the previous section. The calibration surface is flat and therefore the extracted lines are straight. We apply the Hough transform to obtain representations of these lines in each image. The best candidate is selected for a straight line and we save the in-lying points along this line as illustrated in Figure 5. We know these points lie on $y = 0$ in the global coordinate frame defined by the SLAM process. We can therefore construct rays from the camera center through pixel points represented in image frame coordinates in order to find their intersection with the global marker plane. This enables us to reconstruct the 3D location of the points on this plane. By using the rotation and translation information, we can obtain the points in the camera coordinate frame. We formulate this as

$$p_{cam} = \alpha K^{-1} \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} \tag{4}$$

$$\begin{pmatrix} R^t & -R^t t \\ 0 & 1 \end{pmatrix} \begin{pmatrix} p_{cam} \\ 1 \end{pmatrix} = \begin{pmatrix} x \\ 0 \\ z \\ 1 \end{pmatrix}. \tag{5}$$

Here α is an unknown variable which signifies the size of a vector from the camera to the obtained point p in the direction of the projected ray. We solve (5) $s.t, y = 0$ to find α. We then substitute into (4) to find the point p_{cam} in the camera coordinate system. We repeat this process while moving and tilting the camera in various directions until a large number of points is recorded.

3.2 Laser plane acquisition

The recorded points all reside on the plane of the laser beam as illustrated in Figure 5. The closest plane in the least squares sense to all the points is therefore a good approximation to the laser beam plane. We solve this by finding the null space of the following matrix

$$USV^T = \begin{pmatrix} x_1 & y_1 & z_1 & 1 \\ x_2 & y_2 & z_2 & 1 \\ \vdots & \vdots & \vdots & 1 \\ x_n & y_n & z_n & 1 \end{pmatrix}, \tag{6}$$

where USV^T is the singular value decomposition. $\{x_n, y_n, z_n\}$ represent the coordinates of some point p^n_{cam} in $\mathbb{R}^3$ which was obtained in the previous stage. We extract the null vector given by $V_{i,4} = (a, b, c, d)$. (a, b, c, d) now represent the surface coefficients such that $ax + by + cz + d = 0$.

4 Reconstruction

4.1 Point cloud Reconstruction

We wish to calculate a 3D position for each point found on the center line of the laser beam. The recognition of a laser beam in the image means that the 3D point in space lies on the intersection of the laser beam's plane and the straight line representing a ray of light entering

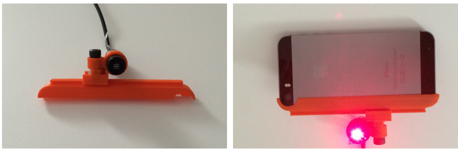

Figure 6: The images depict an iPhone smart-phone device attached to a laser pointer using a 3D printed plastic connector. The three parts comprise all the hardware required for our scanner. See the supplementary material for a video of the scanning process.

the camera and falling on a specific pixel. The light ray is described by

$$P_{cam} = \alpha \cdot (K)^{-1} \cdot \begin{pmatrix} u \\ v + \Delta v \\ 1 \end{pmatrix}, \tag{7}$$

where α is an unknown scaling coefficient, K is the intrinsic camera matrix and Δv is the maximum point of the fitted polynomial. We solve for the α coefficient using the laser plane coefficients determined during calibration. The solution is given by

$$\alpha = \frac{-d}{(a,b,c) \cdot K^{-1} \cdot \begin{pmatrix} u \\ v + \Delta v \\ 1 \end{pmatrix}}. \tag{8}$$

By substituting back into (7) we find the point in the camera coordinate system. We obtain the absolute $3D$ coordinate using the camera position matrix obtained from the tracking library as described by equation (9).

$$P_{world} = \begin{pmatrix} R & t \\ 0 & 1 \end{pmatrix} \begin{pmatrix} P_{cam} \\ 1 \end{pmatrix}. \tag{9}$$

4.2 Color Reconstruction

Simply taking the pixel color for each point we reconstruct will always return the color of the laser beam. We mostly overcome this problem by collecting a temporal series of images during the scanning process. We do this by finding the color of a certain 3D point by looking for it at a time when it was not under the laser beam. We chose this time according to the recording frame-rate and the scanning speed, allowing the laser to move out of the point's region. To accomplish this we save a buffer of previous frames and camera matrices. We then reproject each 3D point onto the oldest image in the buffer using the correct camera position matrix. We use the color from the pixel that the point was projected onto as the correct color value for the 3D point. The projection of each 3D point is given by

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = K \begin{pmatrix} R_{old} & t_{old} \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}. \tag{10}$$

We use the image coordinates (u,v) to extract the color from the old image, $f_{old}(u,v) = R,G,B$.

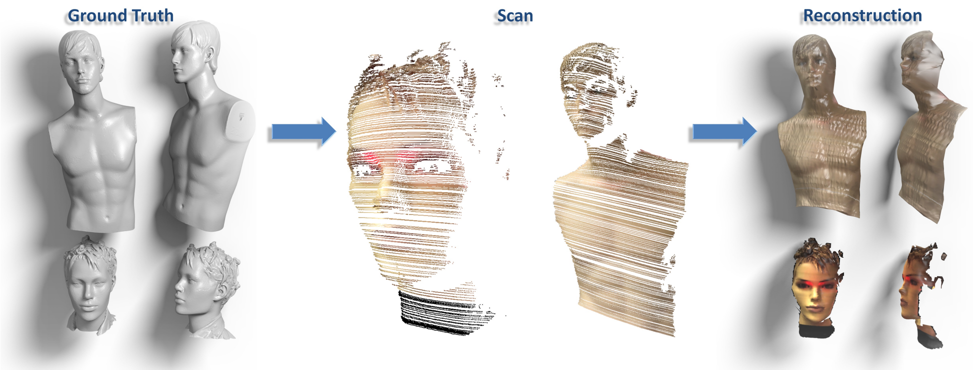

Figure 7: Left: Ground truth models obtained by a professional Artec scanner. Center: Point clouds obtained by our scanning process. Right: Triangle mesh reconstructed from point cloud. The scan was done only on the frontal side of the object and therefore other sides are not covered in these examples. Our method can be seen to produce qualitatively good reconstructions despite the low cost of the attached laser combined with the rolling shutter of the mobile device. See the supplementary material for a video of the process.

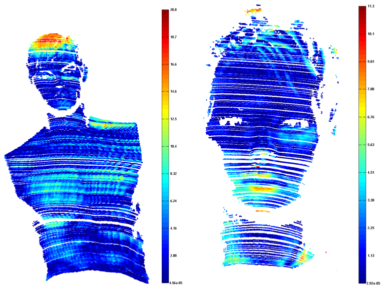

Figure 8: Distortion maps of reconstructed point clouds. Left: Total RMSE = 4.48 mm Right: Total RMSE = 2.25 mm

5 Results

We tested our scanning method using an iPhone 5s from a distance of approximately one meter with the camera resolution set to 720p. We used OpenCV Library [6] for the camera intrinsic calibration and pose estimation for better accuracy. Using an industrial quality 3D scanner with a reported point accuracy of 100 micron we obtained ground truth models of our subjects which were used to assess our method's accuracy. We obtained calibration coefficients as described and generated point clouds of our scans. Using Poisson surface reconstruction [11] we created a triangle mesh which we smoothed using several iterations of Laplacian smoothing to achieve the final results. The point clouds were transformed rigidly onto the ground truth models using Iterative closest point (ICP) [4, 7]. We obtained the per vertex deviation from the ground truth model, and calculated the RMSE for the entire point cloud.

In order to reduce processing time, we used the iPhone's on board vector processor. This helps us achieve real time data processing at 10 FPS for low resolution image capturing at $240X360$ pixels resolution. This is not true for higher resolution scans such as we preformed in (7) where the processing time is more than one second per frame.

6 Conclusions

We demonstrated a way to transform a smart phone device into a 3D scanner using common and cheap hardware. The proposed setup also demonstrates the concept of free hand laser scanning using marker tracking as a cue for device position which improves on standard scanning approaches for hand-held devices where either the laser or the camera are static [5, 17]. Compared to previous mobile 3D reconstruction techniques, the proposed framework produces an accurate reconstruction regardless of lighting or scene texture, and does not require additional computational power besides the smart-phone's onboard processor. To the best of our knowledge it is the first application to combine a cheap line laser and a smart-phone into a fully portable laser scanning device. The method has a number of shortcomings which we are currently addressing. For example, the laser line is occasionally included in the keyframe used to extract the color of the laser points. This distorts the recorded color. Also, the rolling shutter nature of modern mobile phone cameras means that the scanning motion needs to be performed steadily and relatively slowly to avoid unnecessary rolling distortion. We believe that the ubiquity of powerful computing platforms and available commodity hardware will continue to enable the development of applications which require 3D reconstruction. The proposed method is aimed at contributing to these efforts.

Acknowledgments

This research was supported by European Community's FP7- ERC program, grant agreement no. 267414.
Thank you to Roman Rabinovich for his help with designing and printing the adapter.

References

[1] 13th lab. Pointcloud SDK. `http://developer.pointcloud.io/sdk`, 2014.

[2] D. Acosta, García O, and J. Aponte. Laser triangulation for shape acquisition in a 3D scanner plus scan. In *Electronics, Robotics and Automotive Mechanics Conference*. IEEE, 2006.

[3] Autodesk. 123d catch. `http://www.123dapp.com/catch`, 2012.

[4] P.J. Besl and N.D McKay. Method for registration of 3D shapes. In *Robotics-DL*, pages 586–606. International Society for Optics and Photonics, 1992.

[5] J.Y.Y. Bouguet and P. Perona. 3D photography on your desk. In *Sixth International Conference on Computer Vision. (ICCV)*. IEEE, 1998.

[6] G. Bradski. Opencv library. *Dr. Dobb's Journal of Software Tools*, 2000.

[7] Y. Chen and G. Medioni. Object modeling by registration of multiple range images. In *Proceedings of International Conference on Robotics and Automation*. IEEE, 1991.

[8] K. Kolev, P. Tanskanen, P. Speciale, and M. Pollefeys. Turning mobile phones into 3D scanners. In *Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2014.

[9] T.M. Koller, G. Gerig, G. Szekely, and D. Dettwiler. Multiscale detection of curvilinear structures in 2D and 3D image data. In *Fifth International Conference on Computer Vision. (ICCV)*. IEEE, 1995.

[10] V. Matiukas and D. Miniotas. Detection of laser beam's center-line in 2D images. *Electronics & Electrical Engineering*, 2009.

[11] K. Michael, B. Matthew, and H. Hugues. Poisson surface reconstruction. In *Proceedings of the fourth Eurographics symposium on Geometry processing*, 2006.

[12] O. Rubinstein, Y. Honen, A.M. Bronstein, M.M., Bronstein, and R. Kimmel. 3D-color video camera. In *Twelfth International Conference on Computer Vision Workshops (ICCV Workshops)*. IEEE, 2009.

[13] G. Schindler. trimensional. http://www.trimensional.com/, 2014.

[14] H. Surmann, K. Lingemann, A. Nüchter, and J. Hertzberg. A 3D laser range finder for autonomous mobile robots. In *Proceedings of the 32nd International Symposium on Robotics (ISR)*, 2001.

[15] P. Tanskanen, K. Kolev, L. Meier, F. Camposeco, O. Saurer, and M. Pollefeys. Live metric 3D reconstruction on mobile phones. In *2013 IEEE International Conference on Computer Vision (ICCV)*. IEEE, 2013.

[16] trnio. trnio. http://www.trnio.com/, 2013.

[17] S. Winkelbach, S. Molkenstruck, and F.M. Wahl. Low-cost laser range scanner and fast surface registration approach. In *Pattern Recognition*, pages 718–728. Springer, 2006.