Forensic use of Mobile Phone Cameras: Measuring the Height of Person

José Mendes, Pedro Miraldo, and José Gaspar
Institute for Systems and Robotics,
Instituto Superior Técnico, University of Lisbon, Lisbon, Portugal
joseamendes@tecnico.ulisboa.pt, pedro.miraldo@tecnico.ulisboa.pt, jag@isr.utl.pt

***Abstract*—This work addresses the height estimation of a person in a picture, that was captured by a RGB camera on a mobile phone. In this work, we use the assumption that an auxiliary mobile color-depth camera is used to aid in the calibration of the phone camera that originally took the picture. The proposed calibration method use the Direct Linear Transformation (DLT) algorithm with points and/or lines. The use of lines will allow the calibration data to benefit from image processing fitting and detection tools, improving the results. The height estimation will be based on 3D pose prediction model, or considering that the person is in vertical pose. Uncertainty analysis in height estimation shows how camera parameters (such as tilting angle and zoom) can make the estimate more accurate and precise.**

Keywords: Camera calibration, DLT-points, DLT-lines, height estimation, 3D body pose, uncertainty.

I. Introduction

The increasing need of surveillance in public spaces, and the recent technological advances on embedded video compression & communications, made camera networks ubiquitous. Nowadays technological advances already allowed such a wide installation of camera networks. However the calibration of these cameras, considering an unique reference frame, is still an active research topic. These calibration parameters are essential for further higher level processing, such as: people/car tracking, event detection, and metrology, i.e. some of the most active research subjects in Computer Vision / Video Surveillance.

Many times, surveillance and mobile phone cameras capture suspects that need to be arrested. A key step for re-identifying those suspects is the image based measurement of biometric data, such as the body height. Methods developed so far use simple ratios to make approximate estimate of heights. Even when the suspect try to minimize his biometric signature, the perpetrator's vertical height is calculated in the image to make an estimation of his actual body height (no consideration is taken to the subject's posture). Due to different postures, individual variations in standing & gait, and loss of information when the 3D reality is captured in a 2D picture, the body height estimation from surveillance camera images is difficult.

A. Related Work

Conventional calibration methodologies (such as the one proposed by Tsai [19], Heikkila [13], Zhang [22], or Bouguet [4]) require a known pattern in image. Precise calibration demands that the know pattern is covering most of the imaging area, meaning that it will be impractically large if camera is mounted at high position. Also the methodologies used for camera calibration are mostly focused in the intrinsic parameters, and thus do not provide distance (rigid pose transformations representing the extrinsic parameters) among various cameras. They are not designed to provide a global coordinate system for all cameras.

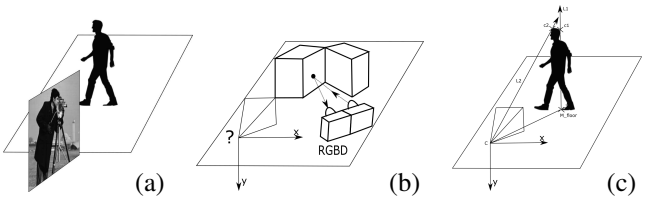


Figure 1: Representation of the problem addressed in this thesis. Person image captured from uncalibrated camera (a). Camera calibration using an auxiliary Kinect camera (b). Height estimation using the calibrated camera (c).

Laser Range Finders (LRF), combined with SLAM, proved to provide reliable scene information (3D clouds of points) of large areas [14]. These 3D maps can, therefore, be used to provide required data to calibrate a camera, by selecting a region of interest on the map.

Recently, color-depth cameras (also known as RGBD cameras) have become an interesting low cost alternative to LRF sensors, [7]. A set of 3D points is simply acquired by back-projecting 2D points from the RGBD image plane. Features can be detected in a network camera image and, then, matched with the RGBD image points. This defines a set of 2D-to-3D points correspondences, which can be used to estimate the camera projection matrix using the Direct Linear Transformation (DLT) [2], [10].

The use of 2D lines images allows some methods for line fitting to be added to the DLT. The linear constraints for calibration from line correspondences, used to estimate the projection matrix, are as simple as with points [18]. Also, despite the low computational complexity of the proposed calibration, the results based on lines provided some similar accuracy. The ability to assess the uncertainty in the estimates of a calibration method is important, not only to infer errors on 3D reconstruction but also as a means to validate and improve the calibration process. This level of accuracy is analyzed in

[9], when calibration based on DLT-Lines is used, showing how the uncertainty propagates from the measurement process to the uncertainty on the calibration parameters (either intrinsic, extrinsic, or 3D reconstruction).

In [20], it is presented an effective calibration method based on planar template methodology. The described method is based on the two step method of Tsai's,[19], improving it computationally efficient by using co-planar points.

About height estimation in body measurement, in [15], it is shown how it is possible to perform height estimation and, in order to have more sophisticated surveillance videos, the importance to track people. Height has been long used in forensic procedures to narrow the list of possible suspects (it is not distinctive enough to be used in biometric identification). However, by estimating the heights of tracked subjects on different cameras, it could provide an important additional feature, allowing a more robust estimation of the tracking object over different scenes. This work introduces a method to estimate the height of human subjects tracked on calibrated surveillance camera images.

In the topic of estimating the person's pose, using static images, there are some works in the literature. For example, in [1], the detection and articulated pose estimation uses a trained appearance model, and a flexible kinematic tree prior of body configurations parts, to estimate 2D pose. Some extension of tree-based models were proposed, using part mixtures for capturing contextual (co-occurrence relation between parts) and spacial relations with histogram of gradients ([17]) to estimate 2D pose, [21]. The computational advances of Deep Neural Networks have revolutionized 2D pose estimation, producing accurate 2D predictions, even for poses with self-occlusions [5]. With the "Big-data" sets of 3D information, it is possible to predict the 3D pose from the 2D pose, using simple memorization [5] (nearest neighbor model).

B. Problem Formulation

It is possible to perform height estimation from a calibrated camera, with the assumption that some parameters are known. Since a calibration can be performed using methods [18], with data from LRF or from color-depth camera, then it can be used to estimate height, see Fig. 1(c). SIFT algorithm matches will perform a camera calibration estimate, as in [16], that will use DLT-Lines.

In [15], height estimate is made considering that the person is standing. In this work, it is used a tree-based model of part mixtures for capturing contextual co-occurrence relation between parts and spacial relations, with histogram of gradients to estimate 2D pose, [21]. Having a 2D pose approximation, it is used a "nearest neighbor" algorithm to perform matching with library from [5], in order to estimate 3D pose. This will allow the estimation of the person height, in cases where non-vertical poses are considered.

II. Background

A. Camera Projection Model and Back-projection

The used projection model is the pin-hole model. According to this model, a camera is represented by a single point, pin-hole, and all light beams pass through that point, being then projected onto the image plane. Camera image point is formed by the intersection of the optical ray r, with the image plane, describing a transformation between a 3D Euclidean space and a 2D Euclidean space. A scene point $M = [X\ Y\ Z]^T$ can be mapped to an image point $m = [u\ v]^T$, applying the following equation:

$$m \sim PM = K\,[R\ t]\,M. \tag{1}$$

where $\sim$ denotes equal up to a scale factor, P is a 3x4 projection matrix, K is a 3x3 upper triangular matrix containing the intrinsic parameters of the camera, R is a 3x3 rotation matrix representing the orientation of the camera and t is a 3x1 vector representing the position of the camera. The rotation, R and translation, t are defined with respect to a fixed absolute (world) coordinate frame.Knowing projection matrix P, it is possible to characterize camera by decomposing it into extrinsic and intrinsic parameters, [11].

Having the projection model, it is possible to define a projection ray that, from a given image point, allows us obtain the line that represents all 3D points that are images of such image point, as $M = C + \alpha D, \alpha \in \mathbb{R}$ where $C = -P_{1:3}^{-1}P_4$ and $D = P_{1:3}^{-1}m$ can be obtained from (1). Replacing the above equalities, back-projection ray can be rewritten as $R_{proj} = -P_{1:3}^{-1}(-P_4 + \alpha m), \alpha \in \mathbb{R}$. Defining a 3D straight line that corresponds to all 3D points that project to the same point m on image.

B. DLT Based Camera Calibration

The camera model in (1), in its essence based on the projection matrix P, can be estimated using correspondences between world and image points in the formalism of a DLT. More precisely, the DLT (developed by Aziz in [2]) obtains the projection matrix P, by solving a linear system on the matrix entries, based on a set of known 3D points in the world frame $\{M_i : M_i = [X_i\ Y_i\ Z_i\ 1]^T\}$ and their images (2D image points) $\{m_i : m_i = [u_i\ v_i\ 1]^T\}$.

1) DLT Points: Applying the cross product of m_i in both sides of (1), $m_i \times m_i = m_i \times (PM_i)$, results in zero in the left hand side of the equation and, thus, $[m_i]_\times PM_i = 0$ where $[m_i]_\times$ denotes the linear cross product operation[1]. Now, considering the properties of the Kronecker product (here denoted as $\otimes$), one can obtain:

$$(M_i^T \otimes [m_i]_\times)vec(P) = 0, \tag{2}$$

where $vec(P)$ represents the column vectorization of the matrix P. Each of the pairs (M_i, m_i) provides a set of three equations in the entries of $vec(P)$. However, only two of them linearly independent, requiring at least six pairs of points to estimate P. Pre-normalization of the input data is crucial on the implementation of this algorithm. In [12], Hartley suggested an appropriate transformation to translate all data points (3D and 2D points) so that: 1) their centroids are at the origin; and 2) the average distance of data points to the origin is equal to $\sqrt{2}$ for image points and $\sqrt{3}$ for 3D points.

[1]Using this operation, one can write $a \times b = [a]_\times b$ where $[a]_\times$ is a 3×3 skew-symmetric matrix, containing the entrances of the vector a.

a) DLT-Points with Radial Distortion: In this thesis we use the division model for radial distortion [8], that can be defined as $m_u = m_d/(1+\lambda||m_d||^2)$, where λ represents the distortion parameter. It can be rewritten in homogeneous coordinates as $[u_u\ v_u\ 1]^T \sim \left[u_d\ v_d\ 1+\lambda||m_d||^2\right]^T$, which implies that an undistorted point is a simple function of a distorted point $m_u = m_d + \lambda||e_d||$ where $e_d = [0\ 0\ ||m_d||]^T$. Now, adding radial distortion to (2) allows us to rewrite the system of equations, for *i* pairs of (M_i, m_i) as:

$$\begin{bmatrix} M_1 \otimes [m_{1d} + \lambda e_{1d}]_\times \\ \vdots \\ M_i \otimes [m_{id} + \lambda e_{id}]_\times \end{bmatrix} \begin{bmatrix} P_{11} \\ P_{12} \\ \vdots \\ P_{34} \end{bmatrix} = 0, \tag{3}$$

or $(A_{i1} + \lambda A_{i2})vec(P) = 0$ where $A_{i1} = M_i^T \otimes [m_{id}]_\times$ and $A_{i2} = M_i^T \otimes [e_{id}]_\times$ which can be solved as polynomial eigenvalue problem, [3], i.e.:

$$(A_1^T A_1 + \lambda A_1^T A_2)vec(P) = 0 \tag{4}$$

where $A_1 = M_i^T \otimes [m_{id}]_\times$ and $A_2 = M_i^T \otimes [e_{id}]_\times$ for the i pairs (M_i, m_i), obtaining projection matrix P and distortion parameter λ.

2) DLT Lines: Given a 3D line L_i, its image l_i can be represented by the cross product of two image points, $l_i = m_{1i} \times m_{2i}$. Any point m_{ki}, lying in the line l_i, implies that $l^T\ m_{ki} = 0$. Applying the multiplication by l^T on both sides of (1), i.e., $l^T\ m_{ki} = l^T\ PM_{ki}$, leads to:

$$l_i^T P M_{ki} = 0, \tag{5}$$

where M_{ki} is a 3D point in projective coordinates, lying in L_i. Similar to DLT-Points, using the Kronecker product, one can obtains a factorizing form (vectorized projection matrix) as:

$$(M_{ki} \otimes l_i^T)vec(P) = 0. \tag{6}$$

Each pair (M_i, L_i) allows us to write a constraint on the form of (6). So, in order to determine the twelve entries of matrix P, 12 pairs of matches (M_i, L_i) will be needed.

a) DLT Lines with Radial distortion: Also, applying radial distortion model (as in II-B1) to a line results in:

$$l_{12} = \begin{bmatrix} u_{1d} \\ v_{1d} \\ 1+\lambda s_1^2 \end{bmatrix} \times \begin{bmatrix} u_{2d} \\ v_{2d} \\ 1+\lambda s_2^2 \end{bmatrix} = l_{12} + \lambda e_{12}, \tag{7}$$

where s_i is the norm of distorted point $s_i^2 = u_i^2 + v_i^2$. $l_{12} = [u_{1d}\ v_{1d}\ 1]^T \times [u_{2d}\ v_{2d}\ 1]^T$ and there is a distortion correction term $e_{12} = [v_{1d}s_{22} - v_{2d}s_{21}\ \ u_{2d}s_{21} - u_{1d}s_{22}\ \ 0]^T$.

Now, applying the point-to-line constraint allows us to write:

$$\begin{bmatrix} M_1 \otimes [l_{1d} + \lambda e_{1d}]_\times \\ \vdots \\ M_i \otimes [l_{id} + \lambda e_{id}]_\times \end{bmatrix} \begin{bmatrix} P_{11} \\ P_{12} \\ \vdots \\ P_{34} \end{bmatrix} = 0, \tag{8}$$

which can also be solved using polynomial eigenvalue solver, $(B_1^T B_1 + \lambda B_1^T B_2)vec(P) = 0$, as in II-B1, where $B_{i1} = M_i^T \otimes [l_{id}]_\times$ and $B_{i2} = M_i^T \otimes [e_{id}]_\times$, obtaining projection matrix P and a distortion parameter λ.

The main advantage of DLT lines over DLT points is the possibility of applying line fitting and finding techniques, that will add more robustness to user error inputs.

C. Human Pose Estimation

For 3D Human pose estimation, it is first performed a 2D pose detection estimate (based on Histogram of Gradients [21]) and, then, the 2D model parts from image, followed by a matching with a library of 3D model poses [5].

1) Image based 2D pose estimate: 2D pose estimation, presented in [21], uses a novel representation of model parts where, instead of using articulated oriented limb parts, it approximate the model to non-oriented parts. The goal is to represent near-vertical and near-horizontal limbs, see Fig. 2.

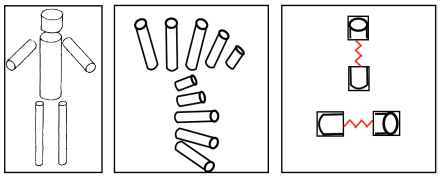


Figure 2: The classic articulated limb model of Marr and Nishihara, [17] . In the middle, the different orientation and foreshortening states of a limb, each of which is evaluated separately in classic articulated body models. On the right, these transformations with a mixture of non-oriented pictorial structures, in this case tuned to represent near-vertical and near-horizontal limbs, from [21].

So, for an image, the pixel location of part i, $p_i = (x, y)$, the mixture component of part i can be written $i \in \{1, ..., K\}, p_i \in \{1, ..., L\}, t_i \in \{1, ..., T\}$, being t_i the type of part i (e.g. a human hand). First, to score a configuration of parts, it is defined a compatibility for type of parts that factors into a sum of local and pairwise scores:

$$S(t) = \sum_{i \in V} b_i^{t_i} + \sum_{ij \in E} b_{ij}^{t_i, t_j}. \tag{9}$$

The parameter $b_i^{t_i}$ favors particular type assignments for part i, while the term $b_{ij}^{t_i,t_j}$ favors particular co-occurrences of part types. For example, if part types correspond to orientations and part i and j are on the same rigid limb, then $b_{ij}^{t_i,t_j}$ would favor consistent orientation assignments. A K-node relational graph can be written, $G = (V, E)$, where edges specify which pair of parts are constrained.

A full score associated with a configuration of part types and position can be written as:

$$S(I, p, t) = S(t) + \sum_{i \in V} w_i^{t_i} \cdot \phi(I, p_i) + \sum_{ij \in E} w_{ij}^{t_i, t_j} \cdot \psi(p_i - p_j), \tag{10}$$

where $\phi(I, p_i)$ is a feature vector extracted from pixel location, p_i in image I, and $\psi(p_i - p_j) = [dx\ dx^2\ dy\ dy^2]^T$, where $dx = x_i - x_j$ and $dy = y_i - y_j$, the relative location of part i with respect to j. This relative location is defined with respect to the pixel grid (and not the orientation part).

The first sum in (10) is an appearance model, that computes the local score of placing a template. The second term can be

interpreted as a "switching" spring model, that controls the relative placement of part i and j by switching between a collection of springs. Each spring is tailored for a particular pair of types $(t_i; t_j)$, and it is parametrized by its rest location and rigidity, which are encoded by $w_{ij}^{t_i,t_j}$. Maximizing S from (10) over p and t, on a given graph G, allows us to compute the message part i by the following:

$$score(t_i, p_i) = b_i^{t_i} + w_{t_i}^i \cdot \phi(I, p_i) + \sum_{k \in kids(i)} m_k(t_i, p_i) \quad (11)$$

where $m_i(t_i, p_i) = \max_{t_i} \; b_{ij}^{t_i,t_j} + \max_{p_i} \; score(t_i, p_i) + w_{ij}^{t_i,t_j} \cdot \phi(p_i - p_j)$. While: 1) (11) computes the local score of part i, at all pixel locations p_i and for all possible types t_i, by collecting messages from the children of i; 2) $m_i(t_i, p_i)$ computes, for every location and possible type of part j, the best scoring location and type of its child part i. Once the messages are passed to the root part, $(i = 1), score1(c1; p1)$ represents the best scoring configuration for each root position and type. One can use these root scores to generate multiple detections in image I by thresholding them and applying a non-maximum suppression (NMS). By keeping track of the argmax indices, one can backtrack to find the location and type of each part, in each maximal configuration.

2) 3D model matching: For 3D pose estimation, since big sets of 3D poses are available (that makes it possible to predict 3D poses from 2D), using 2D members and joints positions, 3D pose can be estimated based on matching from [5].

A probability $p(M|m)$ is modeled with a non-parametric nearest neighbor model. Assuming that we have a library of 3D poses $[M_i]$, paired with a particular camera projection $[P_i]$ such that the associated 2D poses are given by $P_i(M_i)$, the probability distribution over 3D poses based on re-projection error becomes

$$p(M = M_i|m) \propto e^{-\frac{1}{\sigma^2}\|P_i(M_i) - m\|^2}, \quad (12)$$

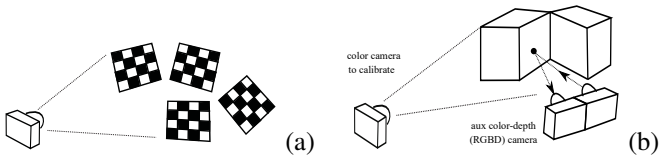
where the MAP estimate is given by the 1-nearest neighbor method. Then squared re-projection error can, then, be reduced to $P_i^* = argmin \, \|P(M_i) - m\|^2$.

A short list of k candidates is built according to (12). These k candidates can be re-sorted, according to the camera matrix. Since we know its intrinsic parameters, and the corresponding 2D and 3D pose for best score candidates, it is possible to estimate camera rotation that will align 3D points with their 2D respective projections. Having the best candidate, simply replace the 3D coordinates (X_i, Y_i) by their scaled 2D counterparts (u, v), obtaining $M^* = [su \; sv \; Z_i], where \; s = \frac{average(Z_i)}{f}$, being f the focal length, given by the camera's intrinsic in P_i, and (Z_i) is the average depth of the 3D joints.

III. Camera Calibration

In order to extract measurements from image, we need calibrated camera to obtain the relation between world points and their projection onto the image, so that later one can obtain a metric information of the world through the image.

In the Bouguet calibration technique, Fig. 3(a), a chess board in multiple poses is used, while in our setup a color depth camera, Fig. 3(b), provides the required 3D information to solve camera calibration problem.



(a) (b)

Figure 3: Calibration Setups: (a) Bouguet calibration and (b) Calibration helped by setup a color-depth (RGBD) camera

In this section the developed calibration methodologies will be described.

A. Assisted Matching of Lines

Since the clicked points will have about 1 pixel uncertainty, this method tries to avoids that, by allowing to pick matching lines found by local gradients maximums. It will remove the click error, while making it easier and faster for the user.

This method can be described as follows. First run Canny Edge detector and Hough Transform to detect lines in RGB; and, then, in depth-color image, run line fitting techniques. After that, it is requested that the user perform the match between the fitted lines in RGB and depth-color image. In the 3D points obtained from depth map, a RANSAC technique is used. Solving (6) allows to estimate projection matrix P.

B. Automated Calibration

The number of lines provided for the user to click on might not be enough to estimate camera projection matrix. In order to solve this, the method concatenates some equations from SIFT matching points between images, (2), with matching lines (6). SIFT results will provide (M_i, m_i) one projection matrix estimate. This estimate will try to automatically make line correspondence (M_i, l_i), making it easier for the user (since it won't require any user input).

This method uses the following steps. It uses SIFT algorithm, in order to detect matching points between RGB (m_i) and depth-color (M_i) images. Using (2) from DLT-Points calibration, and matching of points from previous step is computed, in order to perform a first estimate of projection matrix P. In depth-color image, through Canny Edge detector and Hough Transform methods, lines are detected. Fitting and filtering processes are applied to those lines, as well as a RANSAC algorithm to the 3D points. Using the first estimate of projection matrix P, it is then possible to obtain the matching between the lines found in depth-color image that lie in RGB image. Solving the system (6) concatenated with i 2 from SIFT results, allows us to perform a new estimate of projection matrix P.

IV. Height Measurement Methodologies

In this section, developed measurement methodologies from a calibrated camera are described. Techniques for obtaining vertical or non-vertical measures of the height of a person.

A. Measuring the Height of a Person

In order to estimate a person height relying on the fact that the person is standing, since the depth information is lost, we back-project image points and intersect them with any known 3D information, in this case the person's feet are on the floor (see Fig. 1(c)).

1) Ground point between feet and optic ray tangent to the head: For a calibrated camera, an image point can be mapped into a back-projection ray in the world. So, knowing the floor point back projection ray, and since it is known that point lies on the floor (this is $Z = 0$ in the world frame), it is possible to find the $(X,\ Y)$ coordinates of image clicked point on the world. More precisely:

$$M_{floor} = [X\ Y\ Z = 0]^T = f(m, \alpha), \tag{13}$$

where f is the linear function (back-projection ray) , m is the user clicked point and α is the chosen scalar that will make M having $Z = 0$. Using another user clicked point, on the top of head, it is possible to obtain optic ray tangent to the head, L_2, as shown in Fig. 1(c), where the Back-projection line of user clicked point on floor with respective projection on vertical is L1. And L2 is the back-projection line of user clicked point in head. Closest point in each one of them is respectively c1 an c2.

2) Height estimation: Since it is assumed that the person is standing up, it is possible to define a vertical line from M_{floor} using vector normal to the floor, in this case $[0\ 0\ 1]^T$:

$$L_1 = M_{floor} + t \cdot [0\ 0\ 1]^T, \quad t \in \mathbb{R}, \tag{14}$$

and intersect it with the projection ray from user clicked point on head, i.e.:

$$L_2 = C + s \cdot B_{ray}, \quad s \in \mathbb{R} \tag{15}$$

where $B_{ray} = P_{1:3}^{-1} m_{head}$ defines the back-projection ray, being m_{head} the clicked point on head. Since these two lines do not intersect, we get the points where their distance is minimum, c_1 and c_2 as shown in Fig. 1(c). In order to find the point where L_1 is closest to L_2 it is performed the vector projection of each ray direction with vector normal to both of them. Obtaining $c_1 = M_{floor} + \frac{(M_{floor} - C)^T \cdot n_2}{d_1^T \cdot n_2} d_1$. Similarly, the point on Line 2 that is nearest to Line 1 is given by $c_2 = C + \frac{(C - M_{floor})^T \cdot n_1}{d_2^T \cdot n_1} d_2$, where $n_1 = d_1 \times N$, being d_1 directional vector of L_1 and N normal vector to both lines. Knowing where the point lies on the L_2, it is possible to estimate the person height by that third coordinate, "Height" of the mid point between c_1 and c_2, obtained as $c_{midpoint} = \frac{c_1 + c_2}{2}$.

B. Non-Vertical Body Pose

This section considers the case where a person is in a non-vertical pose, e.g. in a sitting position. Despite the fact that the developed methodologies for finding a person in a image and estimating its skeleton pose , [21], [5], Sec. II-C1,II-C2, are already very effective, it will only provide results in non-metric units, up to scale factor.

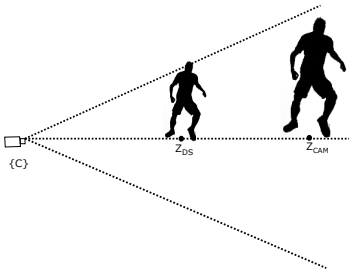


Figure 4: 3D pose model estimation using aligned camera frame, C. Z_{DS} stands for 3rd coordinate in camera frame in normalized body, while Z_{cam} stands for 3rd coordinate in real camera, which means distance from camera center to person foot.

1) Up-to Scale Pose: Toolbox used provides an up-to scale 3D pose. It starts by determining 2D person position in picture, using the techniques described in II-C1.

With the 2D joint locations, it is estimated a list of k, best possible poses candidates, from a 3D skeleton database, Sec. II-C2. Best poses candidates can be aligned with their image projections points,estimating the best 3D pose the one with minimum reprojection error.

2) Metric 3D Pose: To scale 3D skeleton pose to valid metric values, we get the distance to one skeleton point, and we scale it according to such distance. Having a calibrated camera P_w the distance of a point ${}^wM$ to the camera center,${}^wC$ is $\|{}^wM - {}^wC\|$. Knowing ${}^c\widetilde{M}$, the 3D coordinates of the same point provided by the toolbox, and no rotation exists the scale factor between them can be obtained as $\alpha\|{}^c\widetilde{M}\| = \|{}^wM - {}^wC\|$ and so $\alpha = \frac{\|{}^wM - {}^wC\|}{\|{}^c\widetilde{M}\|}$.

Considering just 3rd coordinate one can rewrite scale factor as $\alpha = \frac{[0\ 0\ 1]\widetilde{M}}{[0\ 0\ 1]M} = \frac{Z_{cam}}{Z_{DS}}$ where Z_{DS} is 3rd coordinate of dataset used, `Human3.6M`, and the distance of our person foot to the camera (with (13)), is Z_{cam}, see Fig. 4

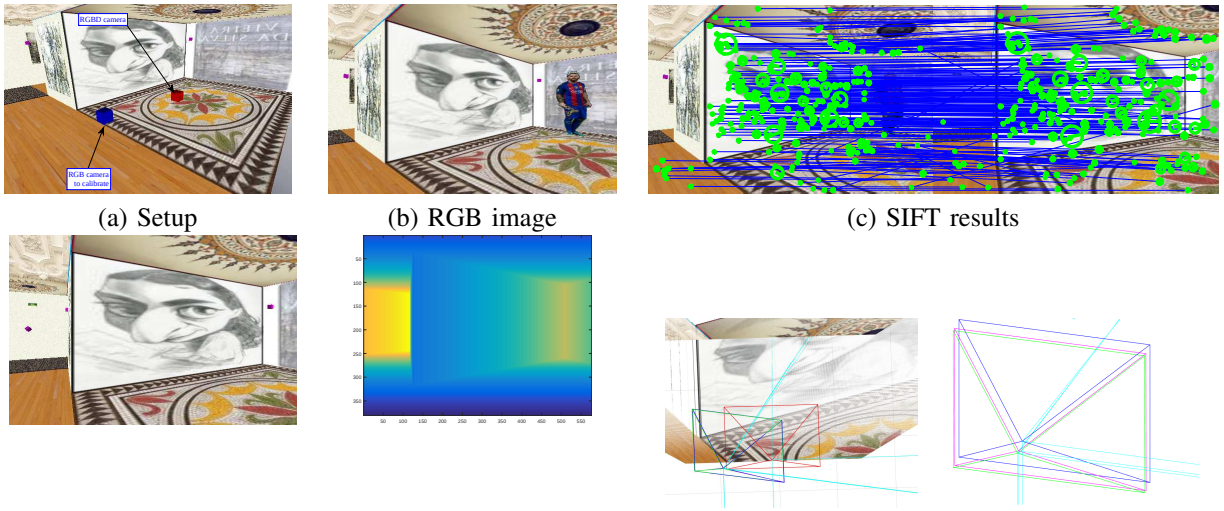
All points can now be scaled $M_i = \alpha\widetilde{M} = \frac{Z_{cam}}{Z_{DS}}\widetilde{M}$ where α is a constant scale. Converting $\widetilde{M}$ into m_i and Z_{DS} and applying the scale factor α one can write $M_i = \alpha H(H^{-1}(K_{intrinsic}^{-1} \cdot m_i)) \cdot Z_{DS}(i)$. Finally expression to obtain scaled valid metric skeleton pose is

$$M_i = \frac{Z_{cam}}{Z_{DS}} H(H^{-1}(K_{intrinsic}^{-1} \cdot m_i)) \cdot Z_{DS}(i) \tag{16}$$

where H stands for operation converting Cartesian coordinates to homogeneous.

V. Experiments and Results

In this section the calibration and height measurements methodologies are tested in real and simulated setup. The simulated setup it is built using the Matlab Virtual Reality Modeling Language toolbox and in Unity. For the real environment it is used a Asus X-tion(RGBD) that will provide 3D information to calibrate an Axis P1347 IP, and a mobile phone camera on the 7th floor of North tower.



(a) Setup (b) RGB image (c) SIFT results

(d) depth-color image (e) depth map (f) Calibration Results

Figure 5: Camera calibration setup in (a), RGB image in (b), SIFT results obtained between RGB and depth-color image (c), depth-color image in (d), and respective depth map in (e), Calibration results obtained(f), RGBD camera in red, ground truth in blue, 1st estimate in green, 2nd improved estimate with lines in magenta

A. Calibration Results

Using now Automated Calibration method in order to test developed calibration method and how it performs, in VRML world, [6].

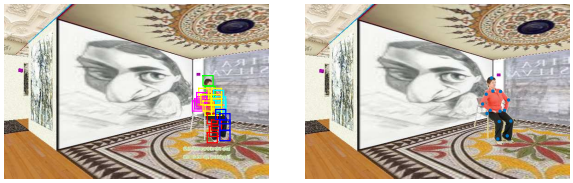
So using the following setup, defining the position and orientation of RGBD and RGB camera to test calibration method, see Fig. 5(a), and their respective views,Fig. 5(b) for RGB and Fig. 5(c) for RGBD, having this last one the depth data, see Fig. 5(d), required for calibration.

We start by obtain SIFT matches as it can be seen in Fig. 5(c). These results will provide the data (M_i, m_i), to solve DLT-Points based approach that will make a 1st projection matrix estimate. Since SIFT algorithm gives some bad matches, outliers, in order to reject them a RANSAC based approach is used. Selecting 6 points from SIFT results, it is performed a camera projection matrix estimate, and the number of inliers from all SIFT matches is counted, an inlier is defined in this case as being one pair (M, m) where the reprojection error is bellow a certain threshold, e.g. 1 pix. Reprojection error is $Err = \sqrt{\sum (m - \hat{m})^2}$ where $\hat{m}$ is the estimate projection of M. The process is repeated until a minimum number of inliers is achieved. Having those inliers projection matrix is re estimated, green camera in Fig.5(f) where the outliers from SIFT were excluded. This estimate is used to match detected lines from Fig. 5(d), converted to 3D through Fig. 5(e), with lines from Fig. 5(b).

Knowing the position of the expected RGB camera, ground truth, it is possible to estimate the position error from performed calibration. With first estimate, green camera, it was obtained a position error of 0.0105m with SIFT results, after constraining with the matching lines to this estimate, the results improved, obtaining a position error of 0.0088m. The distance between rotation matrices was of 0.02 [rad], Showing like this that fitted lines improve camera calibration estimation.

B. Non-Vertical Height Measurement

In order to perform height measurement of non-vertical pose the following test was used, first for 2D pose computation. Allowing to obtain the position of 2D joints in the RGB image, Fig.6



(a) Members position estimate (b) Joints estimate

Figure 6: Person detection (a). Estimated joints location (b).

Using then this 2D RGB points, and knowing camera intrinsic parameters, it is possible using methods described in IV-B to match with the best 3D global pose.

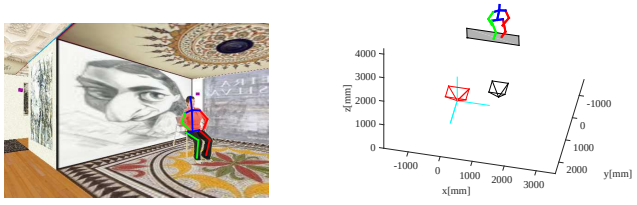


Figure 7: 3D Pose estimate with aligned camera in black and camera who took the picture in red.

Having now 3D model it is possible to estimate the depth of each joint using (16) and then perform one height estimate by adding the important segments.

Having picture of person in same depth conditions and with same resolution, so that the facet in VRML will have same height for both cases, sitting and standing.

Since the facet where this person was declared in VRML has 1.6m of height, the height was estimated with an error of 6.3cm.

In order to test height measuring for Non-Vertical Poses at different depths in the scenario it was used the game development platform Unity. In this scenario with pre-calibrated camera obtained from DLT-Points, it were taken four pictures of a character in non-vertical pose defined with height of 1.84m, see Fig. 8(a), at different depths.

The estimated heights for the increasing depth in scenario were the following: 1.9192m, 1.8546m, 1.8754m and 1.9865m. As it can be seen in the predicted 2D joint position 8(b) the head is detected a bit higher that it should what causes the estimated heights, to have a slightly superior error. The predicted 3D model obtained is pretty close to the character pose, Fig. 8(c).

C. Height Measurement Uncertainty Study

To analyze height estimation uncertainty a Monte Carlo methodology was used. In this method repeated tests are performed in order to obtain numerical results, statistic data. In the different tests it is added random noise to the user clicked point, and in some cases to the calibration data, in order to see how the described methods perform when in presence of noise.

a) Noisy data SIFT and noisy SIFT based calibration: First using Automated Calibration, without any noise, allow us to perform a first height estimate. Applying uniformly distributed noise to the points used to obtain that height, in 100 tests, Fig. 9, estimate several heights with some uncertainty, as it can be seen in Fig. 9(b) and Fig. 9(c).

As it can be seen with the increase of the user uncertainty clicked points also the range of estimated heights will also increase. When noise is also added to calibration data Fig. 9(c) the uncertainty increases but not significantly when compared to Fig. 9(b).

b) Height estimation vs depth: Analyzing how the height estimation varies due to depth and camera tilting, 200 tests were performed where the noise in user click is kept with constant value of 2 pix of std. deviation. We have started by changing the depth of the person in the scenario, as in Fig.10(a) andFig.10(b). After several heights estimates, the results in Fig. 10(c) and in Fig. 10(d) make us confirm that with the increase of the depth also the uncertainty of the estimated height increases, Fig. 10(c). When the person comes closer to the camera there is less uncertainty in height estimation, Fig. 10(d)

c) Height estimation vs tilting: Changing the tilting in camera, as it can be seen in Fig. 11(a) and performing 200 tests in the height estimate with same 2 pix std. dev. noise in user click, Fig. 11(b).

It can be seen that when estimation is near one of the corners of the picture the uncertainty is smaller,Fig. 11(c).

D. Real-World Datasets

In this section the experiments using real data are described. It is used a Asus X-tion(RGBD) that will provide 3D information to calibrate an Axis P1347 IP in the first experience using the Automated Calibration method described in III-B, to obtain a camera calibration that will then be used to perform some measures. For the next experience it is used the same 3D data provided by Asus X-tion(RGBD) but now to calibrate a mobile phone camera, using Assisted Matching of Lines method from III-A and previously described method to make some height estimates.

a) Measurements with Automated Calibration: In this experiment the objective is to estimate the height of a "air conditioning" in the 7th floor of north tower of IST, in order to evaluate described method, using an RGB image of an uncalibrated camera, Fig. 12a) , and depth-color data from kinetic, Fig. 12b)c).

Applying same method described on image Fig. 12(b) and on cropped image Fig. 12(a), it was performed a 1st estimation of projection matrix of camera, as it can be seen in Fig. 12(d), then using this estimate to match lines between RGB and depth-color image, it was possible to improve the projection matrix estimate. The obtained distance between RGB and RGBD cameras from performed calibration was 3.26m, since we know that the distance between them is 3.55m, it was possible to estimate RGB position with an error of 29cm.

b) Person height measure with Assisted Matching of Lines: Now using same depth-color image and depth image as in the previous example, Fig. 12b) and c) but using an RGB image of a mobile phone, Fig. 13a), in order to estimate the height of a subject.

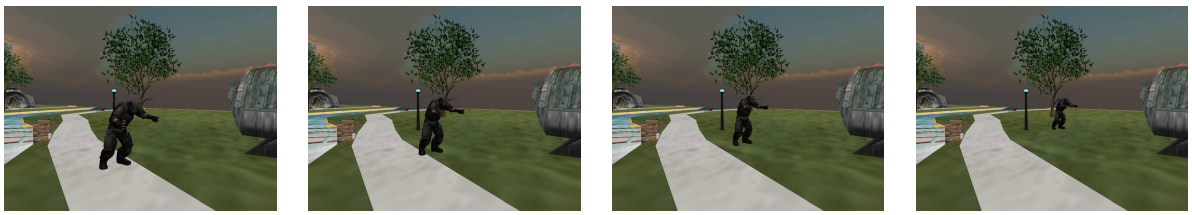
Since SIFT algorithm did not provide results good enough, it was used Assisted Matching of Lines method, where it is asked at user to perform the matching of lines, and pick some more while line fitting techniques are being used. The data used for camera calibration can be seen in Fig. 13a) and b). Where the estimated camera can be seen in point cloud in Fig. 13c).

The methods described previously, IV-B are used for 3D pose model estimation, that will provide an height estimate, I, to be compared with true value of the individual height of $1.84m$. Using methods developed for vertical height estimate, the one where it is used the click on head and on feet and the one who uses predicted model. Results are shown in Tab. I, for both, user clicked points can be seen in Fig. 14(a).

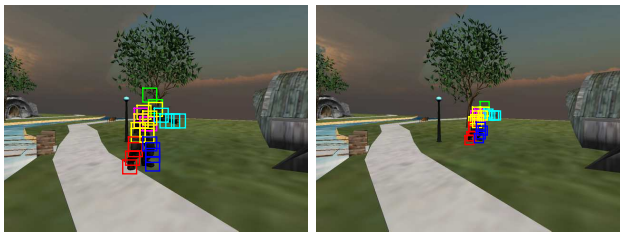
Table I: Height Estimate Results Real Data

Pose	Method	Height Estimate [m]	Err[%]
Vertical	Click method	1.8008	2.13%
	Aided Model	1.7582	4.45%
Non-Vertical	Predicted Model	1.7135	6.87%
	Aided Predicted Model	1.8164	1.28%

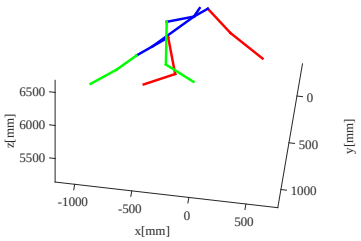
As it can be seen Fig. 13(d),(e) the predicted 2D joint have slight error, left arm of subject is lifted, and hips points considered up, which make the height estimate with the predicted model to have a slightly greater error. When those



(a) Character at different depths

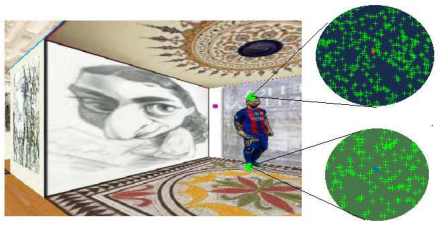


(b) Predicted 2D joints position

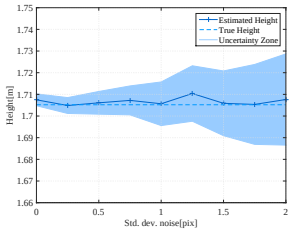


(c) 3D model found

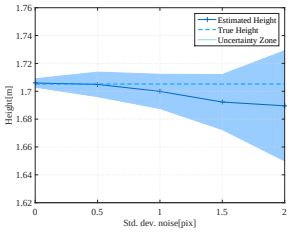
Figure 8: Different depth of character in scenario in (a), person detection results in (b) for different depths, 3D model predicted (c) from 2D detection in one of the depths.



(a) Uniformly distributed noise applied to user clicked (user point in orange, and blue, noisy points in green).



(b) Noisy click SIFT based calibration



(c) Noisy click and noisy data SIFT based calibration

Figure 9: Uniformly distributed noise applied to user clicked (user point in orange, noisy points in green) in (b) and noise applied to user click and calibration data in (c).

points are adjusted by the user the *"Aided Predicted Model"*, Fig. 13(f), pose improves which therefore will make height estimate more accurate.

As it can be seen both methods present some error that might be from camera calibration itself, it will propagate error trought distance measures to height estimate.

VI. Conclusion and Future Work

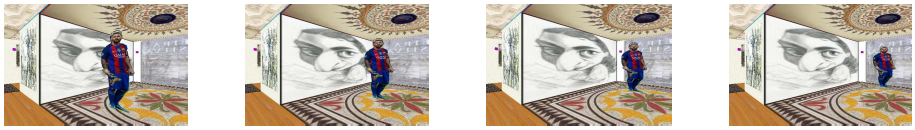
The main objective of this project is the estimation of the height of a person captured in image of a mobile phone camera. We proposed estimating the heights by first calibrating the camera imaging the person. We take the advantage that in most forensic investigations it is possible to visit again the scenario, to obtain 3D data of the scenario required for calibration.

Methods for camera calibration were studied and developed by considering the use of an auxiliary color-depth camera which provides 3D calibration data. Given the forensic nature of applications, correspondences can be obtained by user input, pointing directly to matching lines, which are fine tuned automatically, and finally still obtaining precise calibrations.

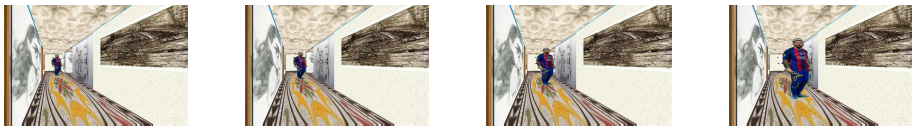
We propose automating the process of calibration by automating the process of data registration. SIFT features for registering the color camera with respect to the color-depth camera and therefore obtain a first calibration of the color camera. The SIFT based calibration allows then registering line features, found in the images and in 3D, and improving the calibration.

Estimating the height of a person methodology, standing up is based on the assumption that the ground is planar and the camera is calibrated with respect to the ground-plane, while in a general position, estimation is based in a toolboxes that detect a person and match the detected person position in the image with a dataset of different human 3D poses. If the person detection has lesser precise results, then the detection can be aided by user input allowing to improve the matching with the 3D dataset of poses. After having determined the human pose, which the toolboxes provide up to a scale factor, we propose assigning metric values to the pose by using the camera calibration and retrieving one 3D point in metric coordinates, similar to the one used for the person standing up case.

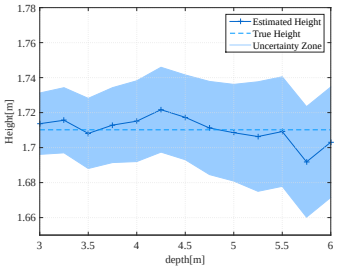
As future work we consider determining the relationship between random perturbations in the input data and the calibration errors for the proposed methodology in non-vertical case. These rules would then serve the purpose of building recom-



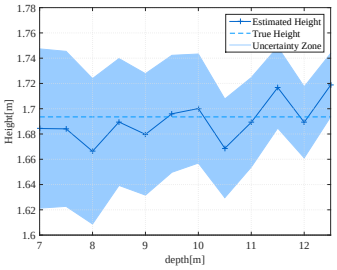
(a) Different depth in hall



(b) Different depth in corridor



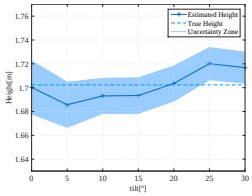
(c) Height mean and std dev. vs depth in hall



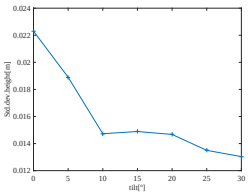
(d) Height mean and std dev. vs depth in corridor

Figure 10: Different depth in hall starting by 3m away from left wall going up to 6m (a), with respective height mean and std dev in (c), different depth in corridor in (b), with respective height mean and std dev in (d).

(a) Various camera tilt angles



(b) Height mean and std dev. vs tilting



(c) Std dev. vs. tilting

Figure 11: Different tilting as it can be seen in (a) from 0° to 30°, respective estimated heights in (b) and their uncertainty in (c).

mendation systems, embedded in the user interfaces, helping the height estimation in captured images from surveillance cameras.

REFERENCES

[1] M. Andriluka, S. Roth, and B. Schiele. Pictorial structures revisited: People detection and articulated pose estimation. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1014–1021, June 2009.

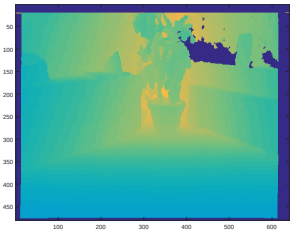
[2] A. Aziz and H. Karara. Direct linear transformation into object space coordinates in close-range photogrametry. In *Proc. of the Symposium on Close-Range Photogrammetry.*, pages 1–18, 1971.

[3] M. Berhanu. *The Polynomial Eigenvalue Problem*. Ph.d. thesis, University of Manchester-School of Mathematics, 2005.

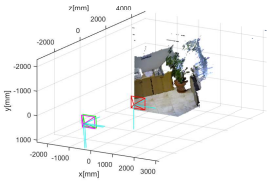
[4] Jean-Yves Bouguet. Camera calibration toolbox for matlab. http://www.vision.caltech.edu/bouguetj.

(a) Image of the camera to calibrate

(b) RGBD image data

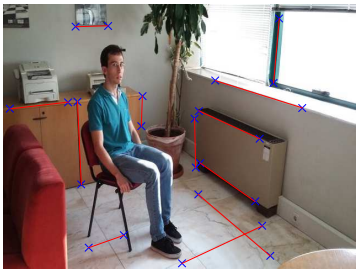


(c) RGBD depth map



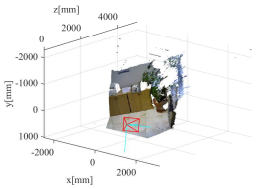
(d) Point cloud and cameras

Figure 12: Calibration of a fixed surveillance camera using a mobile color-depth (RGBD) camera. (a) RGB image of calibrated camera to calibrate. (b) and (c) show color and depth images acquired by a Microsoft Kinect. (d) Results showing the location of the color-depth camera (red) and estimates of the locations of surveillance camera, 1st estimate in green and 2nd estimate in magenta, all drawn over the point cloud acquired by the color-depth camera.

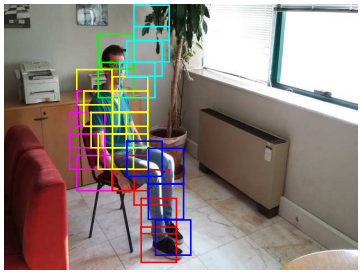


(a) RGB

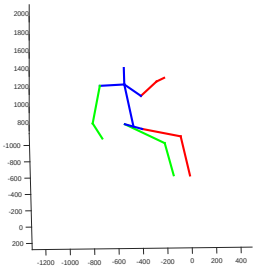
(b) RGBD



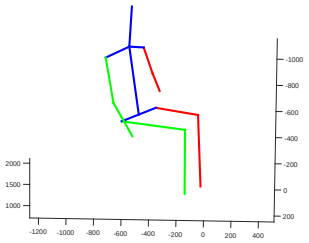
(c) PointCloud



(d) 2D pose estimate

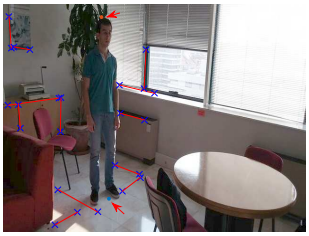


(e) 3D model

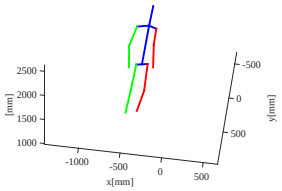


(f) 3D model using user aid in hip click

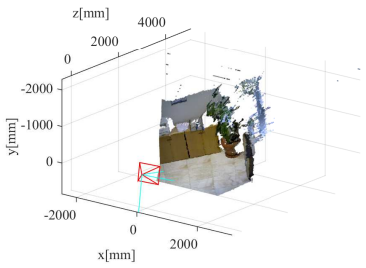
Figure 13: Height Estimation of person sitting using RGB image obtained with mobile phone camera and calibration data from Assited Matching Lines method (a).Depth-color image obtained with Microsoft Kinectic (b).Results showing estimated mobile phone camera position over the pointcloud (c). 2D person members detection in (d). 3D predicted model in (e) from results in (d). 3D model predicted with 2D joints positions aided by the user (f).



(a) RGB with calibration data



(b) 3D model



(c) PointCloud

Figure 14: Height Estimation of person standing using RGB image obtained with mobile phone camera and calibration data from Assited Matching Lines method (a).3D predicted model from person in image in (b).Results showing estimated mobile phone camera position over the pointcloud (c).

[5] Ching-Hang Chen and Deva Ramanan. 3d human pose estimation = 2d pose estimation + matching. *CoRR*, abs/1612.06524, 2016.

[6] W3 Consortium. Virtual reality modeling language. http://www.w3.org/MarkUp/VRML/.

[7] F. Endres, J. Hess, N. Engelhard, J. Sturm, and W. Burgard. Openslam. http://openslam.org/rgbdslam.html. Accessed in 2012-10-22.

[8] A. Fitzgibbon. Simultaneous linear estimation of multiple view geometry and lens distortion. In *Proc. IEEE Conf. Comp. Vision and Pattern Recognition.*, volume 1, pages 125–132, 2001.

[9] Ricardo Galego, Agustin Ortega, Ricardo Ferreira, Alexandre Bernardino, Juan Andrade-Cetto, and JosÃ© Gaspar. Uncertainty analysis of the dlt-lines calibration algorithm for cameras with radial distortion. *Computer Vision and Image Understanding*, 140:115 – 126, 2015.

[10] M. Hansard, R. Horaud, M. Amat, and S. Lee. Projective alignment of range and parallax data. In *Proc. IEEE Conf. Comp. Vision and Pattern Recognition.*, pages 3089–3096, 2011.

[11] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.

[12] R. I. Hartley. In defense of the eight-point algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(6):580–593, Jun 1997.

[13] J. Heikkila and O. Silven. A four-step camera calibration procedure with implicit image correction. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1106–1112, Jun 1997.

[14] V. Ila, J. Andrade-Cetto, R. Valencia, and A. Sanfeliu. Vision-based loop closing for delayed state robot mapping. In *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3892–3897, Oct 2007.

[15] Istvan Kispal. *HUMAN HEIGHT ESTIMATION USING A CALIBRATED CAMERA*, 01 2008.

[16] N. Leite. Calibração de uma rede de câmaras baseada em odometria visual. Master's thesis, UTL - Instituto Superior Técnico, 2009.

[17] D. Marr and H. K. Nishihara. Representation and recognition of the spatial organization of three-dimensional shapes. *Proceedings of the Royal Society of London B: Biological Sciences*, 200(1140):269–294, 1978.

[18] Manuel Silva, Ricardo Ferreira, and JosÃ© Gaspar. *Camera Calibration using a Color-Depth Camera: Points and Lines Based DLT including Radial Distortion*, 08 2017.

[19] R. Tsai. A versatile camera calibration technique for high accuracy 3d machine vision metrology using off-the-shelf tv cameras. *IEEE J. Robot. Automat.*, 3(4):323–344, Aug 1987.

[20] Yue Wang. *Non-contact Human Body Measuring Technology Based on Camera Calibration Technique*, 12 2015.

[21] Y. Yang and D. Ramanan. Articulated pose estimation with flexible mixtures-of-parts. In *CVPR 2011*, pages 1385–1392, June 2011.

[22] Zhengyou Zhang. A flexible new technique for camera calibration, 2002.