2016 23rd International Conference on Pattern Recognition (ICPR)
Cancún Center, Cancún, México, December 4-8, 2016

# Camera Self-Calibration from Tracking of Moving Persons

*Zheng Tang* [1], *Yen-Shuo Lin* [2], *Kuan-Hui Lee* [1], *Jenq-Neng Hwang* [1], *Jen-Hui Chuang* [2], *Zhijun Fang* [3]

[1]Department of Electrical Engineering
University of Washington
Seattle, WA 98195, USA
{zhtang, ykhlee, hwang}@uw.edu

[2]Department of Computer Science
National Chiao Tung University
Hsinchu, Taiwan
linys.cs00g@nctu.edu.tw,
jchuang@cs.nctu.edu.tw

[3]School of Electronic and Electrical Engineering
Shanghai University of Engineering Science
Shanghai, China
zjfang@sues.edu.cn

***Abstract*—In a video surveillance system with a single static camera, tracking results of moving persons can be effectively used for camera self-calibration. However, the current methods need to depend on robustness of both tracking and segmentation procedures. RANSAC has been widely used to remove outliers in finding the vertical vanishing point and the horizon line, but the performance is degraded when the proportion of outliers is high. Last but not least, all of them require excessive simplifications in the algorithmic procedures resulting in increasing reprojection error. In this paper, a robust segmentation and tracking system is applied to provide accurate estimation of head and foot locations of moving persons. The noise in the computation of vanishing points is handled by mean shift clustering and Laplace linear regression through convex optimization. We also propose to use the estimation of distribution algorithm (EDA) to search for the local optimal solution for camera calibration that minimizes average reprojection error on the ground plane, while relaxing the assumptions on camera parameters. Promising evaluations of the performance of our proposed method on real scenes are presented.**

***Keywords*—camera self-calibration; moving persons tracking; mean shift clustering; Laplace linear regression; estimation of distribution algorithm**

## I. Introduction

Camera calibration is a very important step in many computer vision applications such as 3-D object tracking [1, 2] and people localization [3]. It is used to find the intrinsic and extrinsic parameters of the camera, so that the projection matrix from 3-D points to 2-D points can be constructed. Vanishing points of 3-D parallel lines have been proven to be useful to recover both intrinsic and extrinsic parameters [4]. Therefore, some approaches focus on finding accurate vanishing points. Generally, camera calibration methods can be classified into two categories. The first category is based on a known calibration object, from which extensive knowledge of the scene geometry or measurements of sufficient number of 3-D points in the scene can be extracted to derive the camera parameters. However, such information is not always available, and it is hard to be acquired in a large camera network that includes many cameras [5]. The second category is called camera self-calibration, which does not depend on prior knowledge of the camera scenes. Our method lies in this category.

In [6], Lv et al. first present a method to perform self-calibration based on tracking of a human object with known height. They extract the head and foot locations in different frames to compute the vertical vanishing point ($V_Y$) and the horizon line ($L_H$). Many approaches [7-11] are proposed to improve the performance based on this method. In [7], Lv et al. upgrade their approach to apply nonlinear optimization on the parameters using Levenberg-Marquardt algorithm, however, it can only optimize three variables in the projection matrix simultaneously. Krahnstoever and Mendonca [8] propose a Bayesian solution to the self-calibration problem to handle measurement noise and outliers. Junejo and Foroosh [9] use a total least squares method to solve an over-determined system of equations to reduce noise, and the outliers are removed by truncating the Rayleigh quotient. In [10], Wu et al. use RANSAC to remove outliers in the estimation of $V_Y$ and $L_H$ from given locations of heads and feet. Liu et al. [11] propose to use the prior knowledge about the distribution of relative human heights to automatically estimate camera parameters.

Although self-calibration from object tracking has been studied for years, it is still facing many challenges. In all the above methods, the performance of self-calibration is highly dependent on the accuracy of extracted head and foot locations, which is related to the robustness of segmentation and tracking approach. Furthermore, it is common to adopt RANSAC to eliminate outlier points in the estimation of $V_Y$ and $L_H$ [7, 10-11], however, due to noise in measurement, the number of outliers can overwhelm inliers in some scenarios, which will lead to failure of this method. And the threshold of RANSAC needs to be fine-tuned every time as well. Last but not least, Mohedano and Garcia [12] analyze and conclude that complete self-calibration based on estimated $V_Y$ and $L_H$ cannot be achieved if more than one of the intrinsic parameters is unknown. That is why all the mentioned works [6-11] assume that the focal length is the only parameter to be estimated in the intrinsic parameter matrix. This ambiguity in computation will lead to increasing reprojection error. Other limitations also prohibit the development of this area. The work in [6-7] requires accurate detection of leg-crossing for calibration. Hence, it cannot work well when the angle between the object moving direction and the principal axis of the camera is small. In [8], they need to assume that the objects are moving at a constant velocity, and the noise model of measurements is known. The work in [11] assumes that the variation of relative pedestrian heights in the camera's field of view (FOV) is sufficiently low. These are generally not the cases in real world.

978-1-5090-4846-5/16/$31.00 ©2016 IEEE

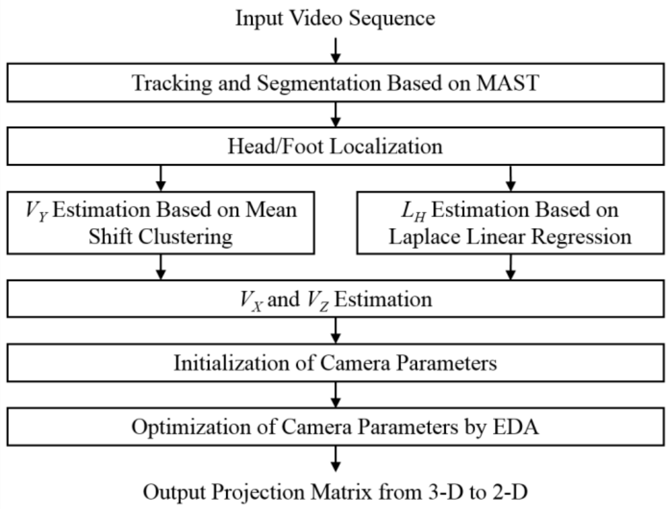

Input Video Sequence
Tracking and Segmentation Based on MAST
Head/Foot Localization
$V_Y$ Estimation Based on Mean Shift Clustering
$L_H$ Estimation Based on Laplace Linear Regression
$V_X$ and $V_Z$ Estimation
Initialization of Camera Parameters
Optimization of Camera Parameters by EDA
Output Projection Matrix from 3-D to 2-D

Fig. 1. Overview flow chart of the proposed system.

In this paper, we propose to use a robust object segmentation and tracking system to achieve accurate head/foot localization. Besides, mean shift clustering is applied to estimate $V_Y$, so that the method is less affected by large number of outliers. Also, we adopt Laplace linear regression to formulate the fitting of $L_H$ into a convex optimization problem. In this way, there is no need to set the threshold parameter to indicate inliers like RANSAC. Moreover, we formulate the problem of optimization based on minimizing the average reprojection error on the ground plane. In this innovative formulation, we do not need to know the actual heights of all walking humans, but only a rough range of the camera height. Fig. 1 shows the overview flow chart of our proposed camera self-calibration scheme.

Our method first employs the estimation of distribution algorithm (EDA) to search for the optimal parameters in camera calibration. This type of search algorithms is based on probabilistic modeling of promising solutions combined with the simulation of the induced models to guide the search. Among the category of EDAs, we adopt the Estimation of Multivariate Normal Algorithm – global ($\text{EMNA}_{\text{global}}$) [13] to optimize all camera parameters simultaneously. In this way, the assumptions of prior knowledge in intrinsic parameters can be relaxed. In our work, we only assume that the people are walking on a visible horizontal ground plane with at least three different locations not on the same straight line observable, and an approximate range of the camera height is known. Therefore, our self-calibration algorithm can be applied widely in video surveillance systems. The advantages of EDAs against most of other metaheuristics are discussed in detail in the review paper [14], including ability to adapt their operators to the structure of the problem, reduced memory requirements, etc. Because parallel computation can be adopted in sampling the population at each generation, the efficiency of EDA can be much higher compared to many other nonlinear optimization approaches.

The rest of this paper proceeds by describing the computation of vanishing points in Section 2. The self-calibration process and optimization of parameters are covered in Section 3. Section 4 presents experimental results and discussions. Finally, Section 5 concludes this paper.

## II. Computation of Vanishing Points

In this section, we first introduce the adaptive segmentation and tracking system adopted, and how the head/foot locations are determined from the results. We then illustrate the process of estimating vanishing points, in which noise and outliers are dealt with using mean shift clustering and Laplace linear regression.

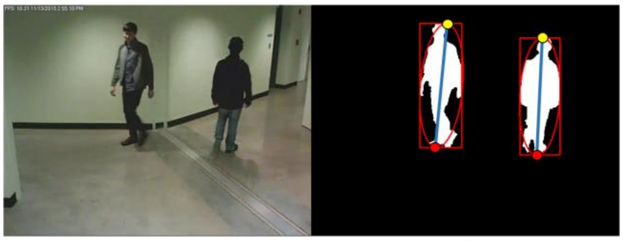
### *A. Object Tracking and Head/Foot Localization*

To find $V_Y$ and $L_H$, we approximate each human body to be a vertical pole with head and foot locations at its ends. They are extracted based on the tracking result and segmented foreground blob of each object. The accuracy of their positions will have a significant impact on the subsequent steps. Hence, it is important to use a segmentation and tracking system that can robustly track moving persons while generating accurate foreground mask.

Chu et al. [15] propose an effective human tracking algorithm, where constrained multiple kernels are utilized to deal with occlusion. However, when some parts of the objects share similar color with the modeled background, the problem of object merging will occur, resulting in failure in both segmentation and tracking. In [16], the authors propose the Multiple-kernel Adaptive Segmentation and Tracking (MAST) system to improve the tracking algorithm by adding a multiple-kernel feedback loop based on preliminary tracking results to dynamically control the decision thresholds in object segmentation. Therefore, this system is able to upgrade the performance of both segmentation and tracking.

In our paper, we improve the method in [16] by combining it with the state-of-the-art change detection algorithm named SuBSENSE [17], which introduces feedback from pixel-level background dynamics and allows increased local sensitivity, especially for regions with intermittent dynamic variations. In the original MAST system, the penalty weight computed from color similarity between current frame and background is applied to Otsu thresholding in background subtraction. Now we apply this penalty weight to the decision thresholds for both the RGB color space and local binary similarity patterns (LBSP) feature space in the SuBSENSE algorithm to preserve more foreground for supporting robust object tracking. In addition, to handle shadowing problem, a shadow detection block based on YCbCr color space [16] is added to the SuBSENSE system, which is also controlled by feedback from tracking. Note that instead of using only one single image, the kernel histograms in the background model are built by the normalized value of all background samples at each pixel location. As a result, we can benefit from the state of the art in object segmentation while maintaining the robustness of tracking.

From the segmentation and tracking results, we can determine the major axis, minor axis and centroid of the binary foreground blob of each object by its first and second moments. Hence, each extracted foreground blob can be approximated by an ellipse. Also, the bounding box for each tracked object can be derived. The head and foot locations are extracted as the intersections of the major axis of the ellipse with the borders of the bounding box (see Fig. 2). This approach has been applied in both single-camera [11] and multiple-camera [18] self-calibration algorithms. To further reduce errors, the head and foot locations that are close to the image boundaries, as well as



(a) (b)

Fig. 2. Head/foot localization example. (a) Original image. (b) Head/foot localization on foreground mask (object bounding boxes and ellipses in red, major axes in blue, head intersection points in yellow, and foot intersection points in red).

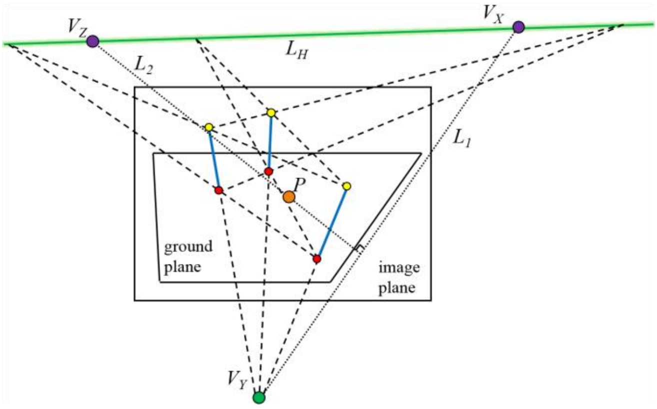


Fig. 3. Self-calibration geometry (major axes in blue, head locations in yellow, foot locations in red). Point $V_Y$ in green is the vertical vanishing point, and line $L_H$ in green is the horizon line. Points $V_X$ and $V_Z$ in purple are the two vanishing points on $L_H$. Point $P$ in orange is the initial position of principal point of the camera. Dotted lines $L_1$ and $L_2$ are auxiliary lines to find $V_Z$.

those with bounding boxes of abnormal size or aspect ratio are discarded. The object blobs that are occluded will not be considered in head/foot localization as well.

### B. Vanishing Points Estimation

As illustrated in Fig. 3, assuming that all objects are standing upright on the ground plane, all the lines passing through their head locations and corresponding foot locations should intersect at $V_Y$. Similarly, the intersection of a line passing through two head locations and another line going through two corresponding foot locations of the same object at different positions should lie on $L_H$, which can be considered as the extension of ground plane at infinity. Thus, three instances of the same object at different timings are sufficient to derive $V_Y$ and $L_H$ [6]. However, because of the existence of measurement errors in object tracking and head/foot localization, the candidates of $V_Y$ usually will not locate at a single point, and there are also many candidates of $L_H$. Therefore, a necessary approach of noise reduction is required.

In $V_Y$ estimation, common methods for centroid estimation such as RANSAC could easily fail when the number of outliers is significantly large. This is very common in camera self-calibration. It is because every extracted head/foot location is associated with all other locations, so that a small error in the localization step will be magnified in the estimation of $V_Y$. The proposed method based on mean shift clustering is explicitly described in Algorithm 1. Because only the cluster with the most candidate points is considered, the outliers that form small clusters will have little impact on the estimation.

**Algorithm 1**: Estimation of the vertical vanishing point by mean shift clustering

```
Input: The set of candidate points of $V_Y$, mean shift window bandwidth $BW$
Output: Estimated $V_Y$ position
while ∃ unvisited candidate point(s) do
    Randomly select an unvisited candidate point as the initial mean point;
    while mean shift distance > $BW * e^{-3}$ do
        Compute the mean of the cluster $c_k$;
        Add all inlier points to $c_k$;
        Mark that these points have been visited;
        Shift to the new mean point;
    end while
end while
for each $c_k$ do
    Merge with other cluster(s) whose mean point(s) are within $BW/2$
Find the cluster $c_k^{\max}$ with the most inlier points;
Set $V_Y$ as the mean of $c_k^{\max}$.
```

We propose to use Laplace linear regression [19] to estimate $L_H$. The robustness of this method arises from the heavy tails of the Laplace distribution, allowing higher likelihood to outliers without having to perturb the straight line. The likelihood model is given by

$$p(\mathbf{y}|\mathbf{x},\mathbf{w}) = \text{Laplace}(\mathbf{y}|\mathbf{w}^T\mathbf{x}) \propto \exp(-|\mathbf{y}-\mathbf{w}^T\mathbf{x}|), \quad (1)$$

where $\mathbf{x}$ and $\mathbf{y}$ denote the corresponding coordinates of the candidate points lying on $L_H$ separately, and the parameters of the horizon line that we want to find are represented by $\mathbf{w}$.

The constrained optimization problem is formulated as

$$\min_{\mathbf{w},\mathbf{r}} \sum_i r_i = \min_{\mathbf{w},\mathbf{r}^+,\mathbf{r}^-} \sum_i (r_i^+ + r_i^-),$$
$$\text{s.t. } r_i^+ \geq 0, r_i^- \geq 0, \mathbf{w}^T\mathbf{x}_i + r_i^+ - r_i^- = y_i, \quad (2)$$

where $r_i \triangleq r_i^+ - r_i^-$ is the $i$'th residual, which is split into two variables representing the $i$'th positive and negative residuals respectively, so that the objective function can be converted into a linear objective. This is a linear programming problem with the following standard formulation:

$$\min_{\boldsymbol{\theta}} \mathbf{f}^T\boldsymbol{\theta} \text{ s.t. } \mathbf{A}\boldsymbol{\theta} \leq \mathbf{b}, \mathbf{A}_{eq}\boldsymbol{\theta} = \mathbf{b}_{eq}, \mathbf{l} \leq \boldsymbol{\theta} \leq \mathbf{u}, \quad (3)$$

in which $\boldsymbol{\theta} = (\mathbf{w},\mathbf{r}^+,\mathbf{r}^-)$, $\mathbf{f} = [\mathbf{0},\mathbf{1},\mathbf{1}]$, $\mathbf{A} = []$, $\mathbf{b} = []$, $\mathbf{A}_{eq} = [\mathbf{x},\mathbf{I},-\mathbf{I}]$, $\mathbf{b}_{eq} = \mathbf{y}$, $\mathbf{l} = [-\infty\mathbf{1},\mathbf{0},\mathbf{0}]$ and $\mathbf{u} = []$. It can be solved by any convex optimization solver such as CVX [20]. Compared to other noise reduction schemes in regression models, this method does not require a predefined threshold to distinguish outliers from inliers. Because it is formulated into a linear programming problem, the speed is much faster than non-linear optimization approaches.

Following Fig. 3, the next step is to find the other two vanishing points, $V_X$ and $V_Z$, located on $L_H$. Firstly, we assume that the initial position of the principal point is at the image center indicated by point *P*. (The searching for a more accurate position of *P* through optimization will be addressed in Section 3). Then, a random point on $L_H$ is picked as $V_X$. Let $L_1$ be the line which passes through $V_X$ and $V_Y$, and $L_2$ be the line that is perpendicular to $L_1$ passing through *P*. Since the principal point has the property that it is the orthocenter of the triangle whose three vertices are $V_X$, $V_Y$ and $V_Z$, $V_Z$ can be determined as the intersection point of $L_H$ and $L_2$. This method has been adopted by many other related works [6, 8-9].

## III. Self-Calibration by Optimization

We use the general pinhole camera model in this paper. A 3-D point $(X, Y, Z)$ can be projected to the 2-D image at $(u, v)$ through a $3 \times 4$ projection matrix $\boldsymbol{P}$:

$$[u, v, 1]^T \sim \boldsymbol{P} \cdot [X, Y, Z, 1]^T, \tag{4}$$

The matrix $\boldsymbol{P}$ can be factorized into 3 matrices, including the intrinsic parameter matrix $\boldsymbol{K}$ containing 5 intrinsic parameters (focal length in x direction $f_x$, focal length in y direction $f_y$, principal point coordinate $(c_x, c_y)$, and skew $s$), the rotation matrix $\boldsymbol{R}$ formed by 3 extrinsic parameters ($roll$ angle around Z-axis, $pitch$ angle around X-axis, and $yaw$ angle around Y-axis), and the translation matrix $\boldsymbol{t}$ with the other 3 extrinsic parameters ($t_X$ along X-axis, $t_Y$ along Y-axis, and $t_Z$ along Z-axis). Their relationship is given by:

$$\boldsymbol{P} = \boldsymbol{K} \cdot [\boldsymbol{R}|\boldsymbol{t}]$$

$$\text{where } \boldsymbol{K} = \begin{bmatrix} f_x & s & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}, \boldsymbol{t} = \begin{bmatrix} t_X \\ t_Y \\ t_Z \end{bmatrix}, \text{and } \boldsymbol{R} = \boldsymbol{R}_Z \cdot \boldsymbol{R}_X \cdot \boldsymbol{R}_Y,$$

$$\boldsymbol{R}_Z = \begin{bmatrix} \cos(roll) & -\sin(roll) & 0 \\ \sin(roll) & \cos(roll) & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

$$\boldsymbol{R}_X = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(pitch) & -\sin(pitch) \\ 0 & \sin(pitch) & \cos(pitch) \end{bmatrix},$$

$$\boldsymbol{R}_Y = \begin{bmatrix} \cos(yaw) & 0 & -\sin(yaw) \\ 0 & 1 & 0 \\ \sin(yaw) & 0 & \cos(yaw) \end{bmatrix}, \tag{5}$$

Some initial values of camera parameters can be calculated, using the vanishing points derived from Section 2, based on the common method used in [6-12]:

$$roll = \tan^{-1}\left(\frac{v_{V_Z} - v_{V_X}}{u_{V_X} - u_{V_Z}}\right), \tag{6}$$

$$f_x = f_y = \sqrt{-\left(v_{V_X}^{rot} \cdot v_{V_Z}^{rot} + u_{V_X}^{rot} \cdot u_{V_Z}^{rot}\right)}$$

$$\begin{aligned} \text{where } v_{V_X}^{rot} &= \cos(roll)\left(v_P - v_{V_X}\right) - \sin(roll)\left(u_{V_X} - u_P\right), \\ v_{V_Z}^{rot} &= \cos(roll)\left(v_P - v_{V_Z}\right) - \sin(roll)\left(u_{V_Z} - u_P\right), \\ u_{V_X}^{rot} &= \cos(roll)\left(u_{V_X} - u_P\right) + \sin(roll)\left(v_P - v_{V_X}\right), \\ \text{and } u_{V_Z}^{rot} &= \cos(roll)\left(u_{V_Z} - u_P\right) + \sin(roll)\left(v_P - v_{V_Z}\right), \end{aligned} \tag{7}$$

$$pitch = \tan^{-1}\left(\frac{v_{V_X}^{rot}}{f_x}\right), \tag{8}$$

$$yaw = -\tan^{-1}\left(\frac{f_x}{\cos(pitch) \cdot u_{V_X}^{rot}}\right), \tag{9}$$

The skew is set to zero, because generally it is safe to assume rectangular pixels [12]. Without loss of generality, we can assume the origin $(0,0,0)$ located at the intersection of ground plane with its perpendicular line passing through camera, so that $t_X$ and $t_Z$ are also set to zero. An approximate range of $t_Y$, which is the negative of the camera height, is assumed to be known.

As has been analyzed in [12], the initial values computed using the above method are based on the assumptions of central principal point, unit aspect ratio and zero skew. Except for the last assumption, the other two do not hold in general situations, and it will result in increasing reprojection error. To relax the assumptions, we formulate the problem of optimization by minimizing the average reprojection error of some projected points on the ground plane using EDA. Thus, the local optimal value of each camera parameter in its corresponding initial range can be found. According to the observation in general camera calibration, the deviation ranges for these parameters are empirically set as: $0.1 \cdot f_x$ for $f_x$, $0.1 \cdot f_y$ for $f_y$, 10 pixels for $c_x$ and $c_y$, and 20 degrees for $roll$, $pitch$, and $yaw$. The detailed description of multivariate optimization is given in Algorithm 2.

**Algorithm 2**: Optimization of camera parameters by EDA

```
Input: Initial ranges of 8 camera parameters, sample size of initial population $R$, sample size of selected population $N$, maximum number of iterations $g_{\max}$, ratio threshold of reprojection error between two generations $r_{thres}$, beginning measurement point $(X_0, 0, Z_0)$, and the number of measurement points in X direction $N_X$ and in Z direction $N_Z$
Output: Local optimals of 8 camera parameters
$P(0) \longleftarrow$ Sample $R$ individuals randomly from the 8-D parameter space; $g \longleftarrow 1$;
while $\frac{d_{g-1} - d_g}{d_{g-1}} > r_{thres}$ and $g < g_{\max}$ do
    Choose a set of parameters from $P(g-1)$;
    Generate $N_X \cdot N_Z$ measurement points $p_{i,j}$ on ground plane starting from $(X_0, 0, Z_0)$ where $i = 0,1,\ldots,N_Z - 1$ and $j = 0,1,\ldots,N_X - 1$;
    Generate $N_Z$ lines $l_i^X$ passing through both $(X_0, 0, Z_i)$ and $V_X$;
    Generate $N_X$ lines $l_j^Z$ passing through both $(X_j, 0, Z_0)$ and $V_Z$;
    Compute error distance $d_{i,j}^X$ of each $p_{i,j}$ with $l_i^X$;
    Compute error distance $d_{i,j}^Z$ of each $p_{i,j}$ with $l_j^Z$;
    $S(g-1) \longleftarrow$ Select $N < R$ individuals within $P(g-1)$ that have lower average reprojection error $d_{g-1} = \frac{\Sigma\left(d_{i,j}^X + d_{i,j}^Z\right)}{N_X \cdot N_Z}$;
    Build probabilistic model $M(g) = \mathcal{N}\left(\mu_g, \sigma_g\right) \longleftarrow$ Estimate the multivariate normal density function from $S(g-1)$;
    $P(g) \longleftarrow$ Sample $R$ individuals from $M(g)$
    $g \longleftarrow g + 1$;
end while
Output $\mu_g$ of $M(g)$.
```

TABLE I. COMPARISON OF COMPUTED CAMERA PARAMETERS AND AVERAGE REPROJECTION ERROR ON TEST VIDEO SEQUENCES

| *Seq. #* | $f_x$ *(pix.)* | $f_y$ *(pix.)* | $c_x$ *(pix.)* | $c_y$ *(pix.)* | *roll (deg.)* | *pitch (deg.)* | *yaw (deg.)* | $\mu_{err}$ *(pix.)* |
|---|---|---|---|---|---|---|---|---|
| 1. Ground Truth | 731.3880 | 728.2518 | 322.1298 | 237.2676 | -3.1371 | 16.2676 | -78.3065 | N/A |
| 1. Method in [6] | 611.5239 | 611.5239 | 320.0000 | 240.0000 | 5.7439 | 22.4758 | -64.9974 | 11.7954 |
| 1. Method in [10] | 638.2676 | 638.2676 | 320.0000 | 240.0000 | 3.8800 | 23.2010 | -71.8167 | 8.7750 |
| 1. Proposed w/o EDA | 738.7650 | 738.7650 | 320.0000 | 240.0000 | 5.0689 | 17.6076 | -79.0154 | 6.0133 |
| 1. Proposed | 730.9167 | 735.9371 | 322.9955 | 236.1948 | -5.0345 | 17.4224 | -79.1491 | 2.50E-5 |
| 2. Ground Truth | 731.3880 | 728.2518 | 322.1298 | 237.2676 | -1.8887 | 11.0081 | -68.7126 | N/A |
| 2. Method in [6] | 618.7858 | 618.7858 | 320.0000 | 240.0000 | 2.3671 | 8.7161 | -71.5302 | 4.9334 |
| 2. Method in [10] | 647.4640 | 647.4640 | 320.0000 | 240.0000 | 1.8874 | 9.8994 | -71.7033 | 5.0624 |
| 2. Proposed w/o EDA | 679.6617 | 679.6617 | 320.0000 | 240.0000 | 1.7928 | 10.7818 | -70.3027 | 4.6445 |
| 2. Proposed | 727.6335 | 728.1606 | 321.4372 | 241.1506 | -2.2546 | 10.3345 | -70.3032 | 3.12E-5 |
| 3. Ground Truth | 731.3880 | 728.2518 | 322.1298 | 237.2676 | -0.3459 | 18.3846 | -63.8778 | N/A |
| 3. Method in [6] | 606.8088 | 606.8088 | 320.0000 | 240.0000 | -0.8635 | 13.2525 | -67.1697 | 2.1670 |
| 3. Method in [10] | 662.9474 | 662.9474 | 320.0000 | 240.0000 | -0.2164 | 22.4663 | -57.6830 | 0.5403 |
| 3. Proposed w/o EDA | 719.8882 | 719.8882 | 320.0000 | 240.0000 | 0.2693 | 17.4219 | -64.7125 | 0.3398 |
| 3. Proposed | 720.6649 | 729.5090 | 319.8556 | 240.6065 | -0.2658 | 17.2493 | -64.7081 | 1.17E-4 |
| 4. Ground Truth | 437.2689 | 437.8792 | 173.7693 | 142.7878 | 1.5466 | 14.1153 | -54.5257 | N/A |
| 4. Method in [6] | 406.8041 | 406.8041 | 180.0000 | 144.0000 | -0.2633 | 22.4482 | -63.5813 | 0.5051 |
| 4. Method in [10] | 432.0973 | 432.0973 | 180.0000 | 144.0000 | -0.2062 | 20.8494 | -45.6322 | 0.4321 |
| 4. Proposed w/o EDA | 440.5366 | 440.5366 | 180.0000 | 144.0000 | -0.4297 | 16.2182 | -55.8775 | 0.1858 |
| 4. Proposed | 442.4795 | 440.9664 | 176.2516 | 142.1498 | 0.4313 | 15.9846 | -55.6434 | 2.74E-5 |

This camera calibration algorithm requires only two vanishing points on the ground plane and an approximate range of the camera height as input. Therefore, it can be extended to many other applications. For instance, camera calibration can be performed in a single image by locating $V_X$ and $V_Z$ from two orthogonal pairs of parallel lines on the ground plane manually. They can be derived from common structured elements in the scene such as crosswalks or vehicles.

## IV. Experimental Results

In our experiments, we captured three video sequences with duration of 2 to 3 minutes using a common surveillance camera having resolution of 640 x 480 and frame rate of 10 fps. These videos have either single or multiple persons walking on regular ground plane, covering both indoor and outdoor environments. They were captured in natural settings, including problems such as occlusion, object merging, shadowing, and reflection. To get ground truth of camera parameters, 52, 52, and 38 3-D points were measured in the three scenes respectively to compute the projection matrices using the linear method [21]. The fourth sequence that we used to compare all the self-calibration algorithms is from the publicly available EPFL dataset [22]. It shows multiple pedestrians walking on a terrace for 3 ½ minutes with frame rate of 25 fps. The ground truth of the fourth sequence is generated by the Tsai calibration [23], and converted to our coordinate system with the same length unit.

We first apply the MAST algorithm introduced in Section 1 on each video. The tracking and segmentation results are then used as the input for self-calibration. To demonstrate the efficacy of our method, we compare the proposed method with three different approaches: (1) the method in [6] that uses a different head/foot localization scheme and is without noise reduction in vanishing points estimation, (2) the method in [10] that is based on RANSAC to eliminate outliers and is combined with the same head/foot localization scheme as ours, and (3) the proposed method without optimization by EDA.

The configuration parameters for the object segmentation and tracking system are the same as the default settings in [16] and [17]. The $BW$ in mean shift clustering is empirically set as 100 pixels, which is the same as the distance threshold of RANSAC for the method in [10]. For the optimization of camera parameters, $R$ and $N$ are chosen as 2000 and 20 respectively. The maximum iteration number $g_{\max}$ is 100, and $r_{thres}$ is given as 0.1. These parameters can be empirically determined by the user, and will not significantly affect calibration accuracy from our observations. The beginning measurement point on the ground plane is located at $(1,0,1)$, while $N_X$ and $N_Z$ are both set as 10. The total 100 points form a 10 x 10 square grid on the 3-D ground plane. The length unit used here is meter.

The evaluation results of computed camera parameters and average reprojection error $\mu_{\text{err}}$ on the four video sequences are shown in Table 1. It can be seen that the application of the noise reduction schemes in vanishing points estimation can help improve calibration accuracy. The proposed noise reduction scheme based on mean shift clustering and Laplace linear regression outperforms RANSAC because it can work well even when the number of outliers is significantly large. Another advantage of our proposed method is that there is no need for fine-tuning threshold parameter in $L_H$ estimation. The robustness of self-calibration is further strengthened through optimization using EDA, mainly because the assumptions on intrinsic camera parameters are relaxed. This algorithm takes 20, 24, 23, and 22 iterations to converge in these four sequences respectively. The projected points in EDA optimization on selected frames of the test videos are plotted in Fig. 4.

In addition, the average runtime of each iteration in EDA according to our parameters setting is 0.589 seconds. The

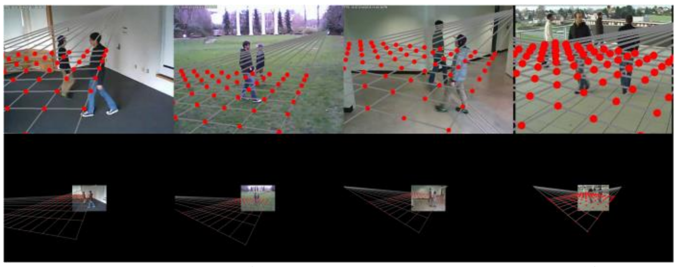


Fig. 4. Projected points in the EDA optimization of camera parameters on frames of the four test sequences (measurement points in red, and auxiliary lines in grey). The reprojection error is proportional to the distance between each measurement point and its two corresponding auxiliary lines from $V_X$ and $V_Z$. (a) *Seq. #1*. (b) *Seq. #2*. (c) *Seq. #3*. (d) *Seq. #4*.

runtime is estimated on an Intel Core i5-4210U PC with 1.70 GHz processor and 8G RAM in a Windows 8 environment. Since camera self-calibration usually only needs to be run once for each static camera, such short computation time for optimization should be acceptable. Furthermore, as is discussed in Section 1, the optimization procedure by EDA can be accelerated through parallel computation when GPU is adopted.

## V. Conclusion

In this paper, we propose a robust single camera self-calibration method based on moving persons tracking. Our contribution lies in (1) combining the state-of-the-art change detection algorithm and tracking algorithm to generate accurate head/foot localization, (2) introducing mean shift clustering and Laplace linear regression through convex optimization to the estimation of vanishing points for noise reduction, and (3) formulating the problem of camera parameters optimization into minimization of average reprojection error on the ground plane, which is supported by EDA that can relax the assumptions on unknown intrinsic parameters. From experiments on real data, it is shown that our self-calibration algorithm can accurately compute both intrinsic and extrinsic camera parameters. The reprojection error is also significantly reduced after optimization. For future development, we are going to combine this method with 3-D human tracking to further optimize the camera parameters using measurements from 3-D trajectories. Moreover, parallel computation using GPU will be implemented to improve computation efficiency.

## Acknowledgment

This research is supported by research grants from Prism Skylabs Inc. and the Ministry of Science and Technology of Taiwan 104-2917-I-009-022.

## References

[1] K.-H. Lee, J.-N. Hwang and S.-I. Chen, "Model-based vehicle localization based on three-dimensional constrained multiple-kernel tracking," *IEEE Trans. Circuits Syst. Video Technol.* (TCSVT), vol. 25, no. 1, pp. 38-50, Jun. 2014.

[2] K.-H. Lee, J.-N. Hwang, J.-Y. Yu and K.-Z. Lee, "Vehicle tracking iterative by Kalman-based constrained multiple-kernel and 3-D model-based localization," in *Proc. IEEE Int. Symp. Circuits Syst.* (ISCAS), May 2013.

[3] Y.-S. Lin, K.-H. Lo, H.-T. Chen and J.-H. Chuang, "Vanishing point-based image transforms for enhancement of probabilistic occupancy map-based people localization," *IEEE Trans. Image Processing* (TIP), vol. 23, no. 12, pp. 5586-5598, 2014.

[4] B. Caprile and V. Torre, "Using vanishing points for camera calibration," *Int. J. Computer Vision* (IJCV), vol. 4, no. 2, pp. 127-139, 1990.

[5] X. Chen, J.-N. Hwang, D. Meng, K.-H. Lee, R. L. de Querioz and F.-M. Yeh "A Quality-of-Content (QoC)-based Joint Source and Channel Coding for Human Detections in A Mobile Surveillance Cloud," *IEEE Trans. Circuits Syst. Video Technol.* (TCSVT), 2016.

[6] F.-J. Lv, T. Zhao and R. Nevatia, "Self-calibration of a camera from video of a walking human," in *Proc. 16th IEEE Int. Conf. Pattern Recognition* (ICPR), vol. 1, pp. 562-567, 2002.

[7] F.-J. Lv, T. Zhao and R. Nevatia, "Camera calibration from video of a walking human," *IEEE Trans. Pattern Analysis & Machine Intelligence* (TPAMI), vol. 9, pp. 1513-1518, 2006.

[8] N. Krahnstoever and P. R. S. Mendonça, "Autocalibration from tracks of walking people," in *Proc. British Machine Vision Conference* (BMVC), 2006.

[9] I. Junejo and H. Foroosh, "Robust auto-calibration from pedestrians," in *Proc. IEEE Int. Conf. Advanced Video and Signal Based Surveillance* (AVSS), pp. 92-97, Nov. 2006.

[10] Q. Wu, T.-C. Shao and T. Chen. "Robust self-calibration from single image using RANSAC," *Advances in Visual Computing*, Springer Berlin Heidelberg, pp, 230-237, 2007.

[11] J. Liu, R. T. Collins and Y. Liu, "Surveillance camera autocalibration based on pedestrian height distributions," in *Proc. British Machine Vision Conference* (BMVC), 2011.

[12] R. Mohedano and N. Garcia, "Capabilities and limitations of mono-camera pedestrian-based autocalibration," in *Proc. 16th IEEE Int. Conf. Image Processing* (ICIP), 2010.

[13] P. Larranaga and J. A. Lozano, *Estimation of distribution algorithms: A new tool for evolutionary computation*, Springer Science & Business Media, 2nd edition, 2002.

[14] M. Hauschild and M. Pelikan. "An introduction and survey of estimation of distribution algorithms," *Swarm and Evolutionary Computation*, vol. 1, no. 3, pp. 111-128, 2011.

[15] C.-T. Chu, J.-N. Hwang, H.-Y. Pai and K.-M. Lan, "Tracking human under occlusion based on adaptive multiple kernels with projected gradients," *IEEE Trans. Multimedia* (TMM), vol. 15, pp. 1602-1615, Jun. 2013.

[16] Z. Tang, J.-N. Hwang, Y.-S. Lin and J.-H. Chuang, "Multiple-kernel adaptive segmentation and tracking (MAST) for robust object tracking", in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing* (ICASSP), pp. 1115-1119, Mar. 20-25, 2016.

[17] P. St-Charles, G. Bilodeau and R. Bergevin, "SuBSENSE: A universal change detection method with local adaptive sensitivity," *IEEE Trans. Image Processing* (TIP), vol. 24, no. 1, pp. 359-373, 2015.

[18] T. Chen, A. D. Bimbo, F. Pernici and G. Serra, "Accurate self-calibration of two cameras by observations of a moving person on a ground plane," in *Proc. IEEE Int. Conf. Advanced Video and Signal Based Surveillance* (AVSS), pp. 129-134, Sep. 2007.

[19] K. P. Murphy, *Machine learning: A probabilistic perspective*, MIT Press, 2012.

[20] M. Grant and S. Boyd, CVX: MATLAB software for disciplined convex programming [Online]. Available at http://stanford.edu/_boyd/cvx.

[21] O. Faugeras, *Three Dimensional Computer Vision: A Geometric Viewpoint*, MIT Press, 1993.

[22] F. Fleuret, J. Berclaz, R. Lengagne and P. Fua, "Multi-camera people tracking with a probabilistic occupancy map", *IEEE Trans. Pattern Analysis & Machine Intelligence* (TPAMI), vol. 30, no. 2, pp. 267-282, Feb. 2008.

[23] R. Y. Tsai, "A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses", *IEEE J. Robotics and Automation*, vol. 3, no. 4, pp. 323-344, Aug. 1987.