

Camera Calibration from Video of a Walking Human

Fengjun Lv, *Member, IEEE*,
Tao Zhao, *Member, IEEE*, and
Ramakant Nevatia, *Fellow, IEEE*

Abstract—A self-calibration method to estimate a camera's intrinsic and extrinsic parameters from vertical line segments of the same height is presented. An algorithm to obtain the needed line segments by detecting the head and feet positions of a walking human in his leg-crossing phases is described. Experimental results show that the method is accurate and robust with respect to various viewing angles and subjects.

Index Terms—Camera calibration, self-calibration, vanishing point, vanishing line, human tracking.

1 INTRODUCTION

1.1 Motivation

ACCURATE camera calibration is essential in many computer vision tasks such as stereo, metrology, and reconstruction. There are also many tasks that can benefit from the knowledge of camera parameters but have less requirements for accuracy; for example, camera models have been used in tracking systems to accommodate change in object scale and infer the depth order of multiple objects in occlusion [15]. Camera models are also necessary in a class of methods called "analysis by synthesis" where 3D objects are projected onto 2D to match with the image observations [8], [16]. It may be impractical to use conventional calibration techniques due to lack of calibration objects or static structure in the scene. This motivates us to develop a more flexible method which exploits object motion for calibration.

1.2 Related Work and Our Approach

Camera parameters can be computed by classical methods if measurements of enough 3D points and their image correspondences are available, as in [6]. This usually requires a time-consuming site survey. Zhang proposed flexible techniques for calibration, which require the camera to observe a 2D planar pattern [13], or a 1D object consisting of multiple collinear points [14] shown at a few different orientations. These techniques still require calibration objects.

Calibration objects are not always available, which has inspired self-calibration methods. Vanishing points of parallel lines in 3D have proven useful for this task. Caprile and Torre [2] described a method to use vanishing points to recover intrinsic parameters from a single camera and extrinsic parameters from a pair of cameras. Liebowitz et al. [9] have developed a method to estimate intrinsic parameters by Cholesky decomposition and applied it to a scene reconstruction problem; they do not explicitly compute the extrinsic parameters. Cipolla et al. [3] presented a method to compute both intrinsic and extrinsic parameters from three vanishing points and

two reference points from two views. Deutscher et al. [5] used vanishing points in a Manhattan world to calibrate cameras for visual tracking. All these methods use the vanishing points computed from static scene structures such as buildings.

In the absence of scene structures, these vanishing point-based calibration methods are not applicable. We present an automatic approach to compute the vertical vanishing point and the horizon line from the motion of a walking human and use this information to compute the camera parameters. Some recent methods have also exploited tracked object motion to avoid the need for such scene structures: Bose and Grimson [1] achieved affine and metric rectification of the ground plane by tracking moving objects, Stau et al. [11] normalized measurable properties such as size, length and height of tracked objects that lie primarily on a ground plane in far-field tracking scenarios. However, these methods are for rectification and/or weak calibration while our method attempts to estimate all intrinsic/extrinsic parameters.

We first apply a robust method to compute the vertical vanishing point and the horizon line given noisy measurements of the images of vertical poles of the same height (Section 2.1). We approximate human bodies as vertical poles and automatically extract the heads and feet at leg-crossing phases from a video sequence (Section 2.2). We use two auxiliary lines to obtain the other two vanishing points. We then directly estimate all camera parameters based on the above information plus the height of the human. Finally, we apply an optimization approach to refine the computed parameters (Section 3).

We show experimental results and error analysis for different viewing angles (Section 4). The results are not as accurate as those obtained by using calibration objects, which is not surprising given the nature of the input; however, the results are consistent and robust *with regard to* different viewing angles and subjects and their accuracy is comparable to those computed from real vertical poles. Moreover, in absence of camera object and scene structure, this approach might be the only way to do calibration under the given conditions and its utility has been verified by our previous works in tracking and human motion analysis [15], [16].

Although the main contribution of this paper is to estimate camera parameters from a walking human, the techniques in Section 2.1 and Section 3 are general and apply to any other object which can provide multiple observations of a vertical pole or multiple vertical poles of the same height. Examples include vertical structure on a moving vehicle or a scene with a few instances of the same object (e.g., lamp poles, columns, and furniture).

2 COMPUTATION OF THE VERTICAL VANISHING POINT AND THE HORIZON LINE

In this section, we first provide a solution to a more general problem of computing the vertical vanishing point V_Y and the horizon line V_L from the images of multiple vertical poles of the same height (Section 2.1.). We then provide an automatic approach to extract human head and feet pair as such poles (Section 2.2).

2.1 Compute V_Y and V_L from the Images of Vertical Poles

If we can observe N vertical poles of the same height in 3D, V_Y can be fixed just by finding the intersection of two (or more) such poles. For any two poles, the lines connecting the top and bottom of the poles intersect at a point on V_L . Therefore, three (or more) noncoplanar poles fix V_L , as illustrated in Fig. 1. Denote the poles as $\{(\mathbf{h}_i, \mathbf{f}_i)\}_{i=1, \dots, N}$, where \mathbf{h}_i and \mathbf{f}_i are the image positions of the i th head and feet, and $\{(\Sigma_{h_i}, \Sigma_{f_i})\}_{i=1, \dots, N}$ are the associated covariance matrices. Because of possible noise and outliers in the measured head and feet locations, a robust method is required to compute V_Y and V_L .

• F. Lv and R. Nevatia are with the Institute for Robotics and Intelligent Systems, University of Southern California, Los Angeles CA 90089. E-mail: {flv, nevatia}@usc.edu.

• T. Zhao is with Sarnoff Corporation, 201 Washington Rd., Princeton, NJ 08543. E-mail: tzhao@sarnoff.com.

Manuscript received 9 Feb. 2005; revised 29 Nov. 2005; accepted 3 Jan. 2006; published online 13 July 2006.

Recommended for acceptance by L. Quan.

For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-0080-0205.

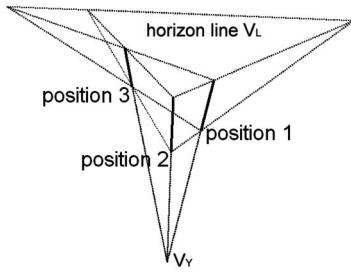


Fig. 1. Three noncoplanar poles can fix V_Y and V_L .

We use the method described in [9] to compute V_Y : Denote \mathbf{m}_i as the midpoint of \mathbf{h}_i and \mathbf{f}_i ; V_Y is the point \mathbf{v} that minimizes the sum of distances from \mathbf{h}_i and \mathbf{f}_i to the line linking \mathbf{m}_i and \mathbf{v} :

$$V_Y = \arg \min_{\mathbf{v}} \sum_{i=1}^N \left(\frac{|\mathbf{w}_i^T \mathbf{h}_i - b_i|}{(\mathbf{w}_i^T \boldsymbol{\Sigma}_i \mathbf{w}_i)^{1/2}} + \frac{|\mathbf{w}_i^T \mathbf{f}_i - b_i|}{(\mathbf{w}_i^T \boldsymbol{\Sigma}_i \mathbf{w}_i)^{1/2}} \right), \quad (1)$$

where (\mathbf{w}_i, b_i) is the line determined by \mathbf{m}_i and \mathbf{v} .

Two head-feet pairs can potentially contribute a point on V_L . Assume the i th point $\mathbf{x}_i = [x_i, y_i]^T$ ($i = 1, \dots, M$) is contributed by two head-feet pairs $(\mathbf{h}_j, \mathbf{f}_j)$ and $(\mathbf{h}_k, \mathbf{f}_k)$ by the intersection of line $\mathbf{h}_j \mathbf{h}_k$ and line $\mathbf{f}_j \mathbf{f}_k$. The covariance matrix $\boldsymbol{\Sigma}_i$ of \mathbf{x}_i is computed by the Jacobian as

$$\boldsymbol{\Sigma}_i = \mathbf{J} \cdot \text{diag}(\boldsymbol{\Sigma}_{h_j}, \boldsymbol{\Sigma}_{h_k}, \boldsymbol{\Sigma}_{f_j}, \boldsymbol{\Sigma}_{f_k}) \cdot \mathbf{J}^T; \quad \text{where} \quad (2)$$

$$\mathbf{J} = \frac{\partial \mathbf{x}_i}{\partial [\mathbf{h}_j^T, \mathbf{h}_k^T, \mathbf{f}_j^T, \mathbf{f}_k^T]^T}.$$

V_L (with equation $\mathbf{w}_{V_L}^T \mathbf{x} = b_{V_L}$, where \mathbf{w}_{V_L} is a unit vector) is determined as

$$(\mathbf{w}_{V_L}, b_{V_L}) = \arg \min_{(\mathbf{w}, b)} \sum_{i=1}^M \frac{|\mathbf{w}^T \mathbf{x}_i - b|}{(\mathbf{w}^T \boldsymbol{\Sigma}_i \mathbf{w})^{1/2}}. \quad (3)$$

As many lines are nearly parallel and thus their intersection is ill-posed, we adopt RANSAC [7] to robustly estimate both V_Y and V_L from a set of data contaminated by a large portion of outliers. Incorporation of the covariance matrices here is critical because the points have very different uncertainties. In practice, points with diagonal elements in a covariance matrix larger than half of the image dimension are useless and are discarded before executing the above estimation.

2.2 Extract Head and Feet Positions from Video Sequence of a Walking Human

Humans are roughly vertical while they stand or walk. In absence of other useful scene structures, they can be used as vertical poles. However, because human walking is an articulated motion, the shape and height of the human vary in different walking phases. The

phase at which the two feet cross each other (*leg-crossing*) is of particular interest in that the feet position is relatively easy to locate and the shape is relatively insensitive to viewpoint. Thus, we aim to extract the head and feet locations at leg-crossing phases. We first detect a walking human from a video sequence by change detection. Then, we extract the leg-crossing phases by temporal analysis of the object shape. Finally, we compute the principal axes of the human's body and locate the human's head and feet positions at those phases.

Moving foreground pixels can be extracted fairly accurately using a statistical background model (e.g., [12]). When the moving objects are sparse in the scene and there is no strong shadow, the foreground blobs (i.e., connected components of foreground pixels) correspond well to the moving objects. The moving blobs can be tracked with a blob tracker by spatial proximity and size similarity [15].

Let $(\mathbf{e}_{1,t}, \mathbf{e}_{2,t})$, $(v_{1,t}, v_{2,t})$ be the first and second eigenvectors and eigenvalues of the covariance matrix of a tracked object at frame t , respectively. (See Figs. 2a and 2b for two examples.) We define a quantity $q_t = v_{2,t}/v_{1,t}$; it reaches its maximum when the walker's two legs are most apart and minimum when the legs cross. q_t is also invariant to the image size and insensitive to the orientation of the walker. Across time, q_t exhibits periodic change (Fig. 2c). The principal frequency of q_t is computed by Fourier transformation.

The periodicity of q_t is less salient when the human walks in a direction close to the viewing angle. However, since a long sequence is considered globally, the frequency is still correct with the contributions from the side-view observations. The requirement for phase accuracy is also less because the shape change is small.

The head and feet positions at the leg-crossing frames are then computed. The center and $\mathbf{e}_{1,t}$ of the blob gives the principal axis of the body at frame t . The head and feet are assumed to be located on the principal axes. The object blob is projected onto the principal axis. As shown in Fig. 2d, the head and feet are located by finding two endpoints along the principal axis whose projection count is above a threshold (set as 1 percent of the blob size for the head and 2 percent for the feet).

This method is not restricted to one human track. Tracks of multiple walkers can be easily used to compute the vertical vanishing point. To compute the horizon line, each track contributes one set of intersections and multiple sets are combined in the line fitting process. When a human walks fast in a side-view, the forward shift of his/her center of gravity may cause the principal axes to lean forward slightly. This could result in inaccuracy of the vertical vanishing point. Considering multiple walkers in opposite directions can alleviate this problem.

One question about the above approach arises: How much does the accuracy depend on the viewing angle (especially the tilt angle)? When the viewing angle is parallel to the ground plane, V_Y tends to infinity; on the contrary, when viewed with a large tilt angle, it is difficult to accurately locate head and feet positions. We

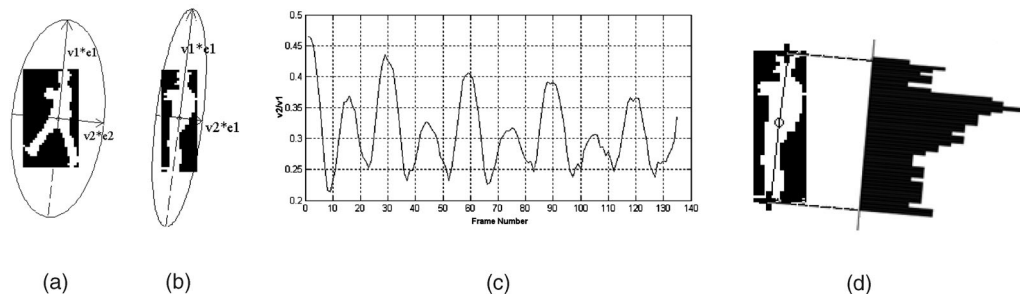


Fig. 2. Extracting human leg-crossing phases from a video sequence. (a) and (b) Eigen analysis on the human shape at two different phases. The arrows are $\sqrt{v_{1,t}}\mathbf{e}_{1,t}$ and $\sqrt{v_{2,t}}\mathbf{e}_{2,t}$, respectively. (c) Plot of $q_t = v_{2,t}/v_{1,t}$ over time. (d) Head and feet positions are located by finding two endpoints along the principal axis.

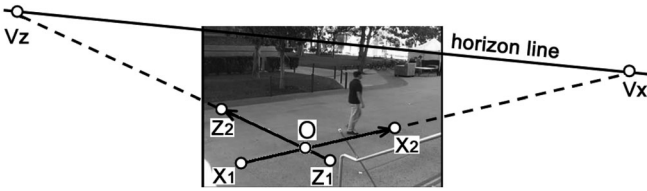


Fig. 3. Given the horizon line, two auxiliary lines X_1X_2 and Z_1Z_2 are needed to obtain the other two vanishing points. X_1X_2 and Z_1Z_2 are parallel to the ground plane and orthogonal to each other.

analyze the robustness of our approach *with regard to* two different viewing angles in Section 4.

3 CALIBRATION ALGORITHM

3.1 Camera Model

In a general pinhole camera model, the relationship between a 3D point $[X, Y, Z, 1]^T$ and its 2D image projection $[u, v, 1]^T$ can be represented by a 3×4 projection matrix M :

$$[u, v, 1]^T \sim M \cdot [X, Y, Z, 1]^T. \quad (4)$$

M is determined by five intrinsic parameters (focal length f , principal point (u_P, v_P) , aspect ratio a , and skew s) and six extrinsic parameters corresponding to the transformation between the World Coordinate System (WCS) and the Camera Coordinate System (CCS). Here, we conveniently place the origin of WCS on the ground plane and specify the transformation between WCS and CCS as follows: CCS is initially aligned with WCS . Then, it is translated to T_C , followed by a rotation around the Y -axis by angle pan , then a rotation around the X -axis by angle $tilt$ and, finally, a rotation around the Z -axis by angle $roll$. The corresponding rotation matrices are R_Y , R_X , and R_Z , respectively. Accordingly, M can be represented by the product of the following three matrices:

$$M = A \cdot R \cdot [I - T_C] \quad \text{where } A = \begin{pmatrix} f & s & u_P \\ 0 & f \cdot a & v_P \\ 0 & 0 & 1 \end{pmatrix} \quad \text{and } R = R_Z \cdot R_X \cdot R_Y. \quad (5)$$

3.2 The Algorithm

Our calibration algorithm consists of the following five steps:

Step 1. Obtain three vanishing points: The vertical vanishing point V_Y and the horizon line V_L are computed in Section 2. In order to obtain the other two vanishing points V_X and V_Z , we need to specify two auxiliary lines X_1X_2 and Z_1Z_2 , which are parallel to the ground plane and mutually orthogonal to each other. The directions of $\overrightarrow{X_1X_2}$ and $\overrightarrow{Z_1Z_2}$ correspond to the directions of the X -axis and the Z -axis of WCS . The intersections of the two lines and V_L are V_X and V_Z , as illustrated in Fig. 3.

Step 2. Locate the principal point: Under the assumption of zero skew ($s = 0$) and unit aspect ratio ($a = 1$), the orthocenter of the triangle with the three vanishing points as vertices is the principal point [2]. We assume that $s = 0$ and a is known. All v coordinates of the original image are first multiplied by a scale factor $1/a$ so that the aspect ratio becomes one. In Step 5, we will adjust the parameters that are affected by this scale factor.

Step 3. Compute focal length and rotation angles: $[1, 0, 0, 0]^T$ is the homogeneous 3D coordinate of the point at infinity whose 2D image is V_X . Apply (4) and (5) to $[1, 0, 0, 0]^T$ and we get

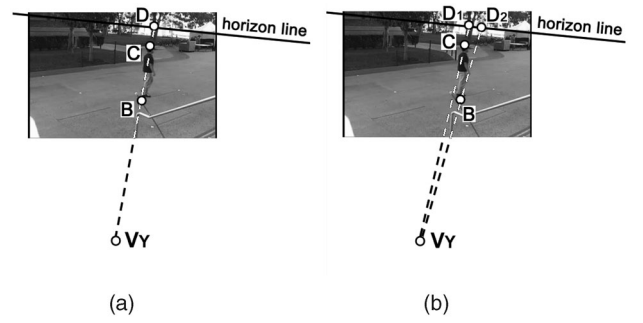


Fig. 4. Computing camera height using cross ratio invariance. (a) D is the intersection of the horizon line and the line passing V_Y , feet B and head C of the walking human. (b) If V_Y, B , and C are not collinear, there are two intersections (D_1 and D_2).

$$u_{V_X} = fR_{11}/R_{31} + u_P = f \cos(roll) \cot(pan) / \cos(tilt) + f \sin(roll) \tan(tilt) + u_P, \quad (6)$$

$$v_{V_X} = fR_{21}/R_{31} + v_P = -f \sin(roll) \cot(pan) / \cos(tilt) + f \cos(roll) \tan(tilt) + v_P. \quad (7)$$

Similarly, for $[0, 1, 0, 0]^T$, we get

$$u_{V_Y} = fR_{12}/R_{32} + u_P = -f \sin(roll) \cot(tilt) + u_P, \quad (8)$$

$$v_{V_Y} = fR_{22}/R_{32} + v_P = -f \cos(roll) \cot(tilt) + v_P, \quad (9)$$

where R_{11}, \dots, R_{33} are elements of the rotation matrix R . (u_{V_X}, v_{V_X}) is the image coordinate of V_X and (u_{V_Y}, v_{V_Y}) is the image coordinate of V_Y . Equations (8) and (9) give us

$$roll = \tan^{-1} \left(\frac{u_{V_Y} - u_P}{v_{V_Y} - v_P} \right). \quad (10)$$

From (6)-(10), we get

$$f = \sqrt{(\sin(roll)(u_{V_X} - u_P) + \cos(roll)(v_{V_X} - v_P))(\sin(roll)(u_P - u_{V_Y}) + \cos(roll)(v_P - v_{V_Y}))}. \quad (11)$$

Although, theoretically, the term under the square root sign in (11) should always be positive, it can be negative, in practice, which indicates an error in estimating V_Y or V_L . In such cases, we adopt a remedy similar to [10]. We increase the inlier threshold in the RANSAC algorithm (Section 2.1) to remove more outliers and repeat all previous steps until f is a real number.

Consequently, from (6), (7), (10) and (11), we get

$$tilt = \tan^{-1}((\sin(roll)(u_{V_X} - u_P) + \cos(roll)(v_{V_X} - v_P))/f). \quad (12)$$

From (6), (7), (10), (11), and (12), we get

$$pan = \tan^{-1}(f / (\cos(tilt)(\cos(roll)(u_{V_X} - u_P) - \sin(roll)(v_{V_X} - v_P))). \quad (13)$$

For computational convenience, the orientation of CCS as well as WCS is set such that the positive direction of the Y axis points down (same as the v -direction in the image coordinate system) and the positive direction of the Z axis points into the screen. Under this configuration, $roll$ and $tilt$ computed using (10) and (12) have no ambiguity because $roll$ usually has a small value and $tilt$ is within the range of $[-\pi/2, \pi/2]$ (when the camera points straightly down or up), in which the tangent has a single value. Computing pan by using (13) can give two values within the range of $[-\pi, \pi]$, but this ambiguity can be easily eliminated by checking if $\overrightarrow{X_1X_2}$, the positive direction of X -axis in Fig. 3, is from left to right or otherwise.

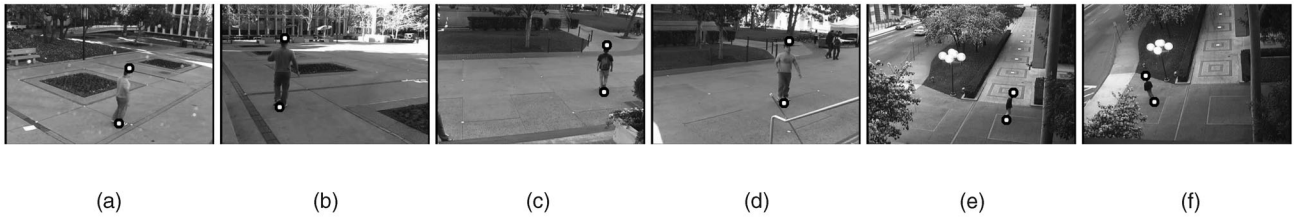


Fig. 5. One detected leg-crossing frames of each scene.

Step 4. Compute the translation vector T_C : If H , the height of the walking human, is known, the cross ratio invariance [4] can be used to infer camera height H_C , as shown in (14).

$$\frac{H}{H_C} = 1 - \frac{d(C, D) \cdot d(B, V_Y)}{d(B, D) \cdot d(C, V_Y)}. \quad (14)$$

Here, as illustrated in Fig. 4a, assuming V_Y , feet B , and head C of the walking human are collinear, D is the intersection of the horizon line and the line passing through V_Y , B , and C . Function $d(P1, P2)$ denotes the distance between two points $P1$ and $P2$ on a $2D$ image. When V_Y , B , and C are not collinear, we compute two intersections $D1$ and $D2$, as shown in Fig. 4b, and we use the midpoint of $D1D2$ as D .

Since each vertical line segment gives us a camera height, we use the median of these camera heights as the final value of H_C . There is no way to recover the real length of a $3D$ line segment from a single image if no metric information is given. So, if H is unknown, its value can be arbitrarily set to an assumed value H' ; for example, the average height of humans. All $3D$ world coordinates are then subject to a scale factor H/H' .

Since $T_{C2} = -H_C$, the only unknown parameters now are T_{C1} and T_{C3} . If we have a reference point, say O , with known $3D$ world coordinates and corresponding image coordinates (u_O, v_O) , T_{C1} and T_{C3} can be fixed by solving the two projection equations in (4). If we conveniently specify O as the origin of WCS (as shown in Fig. 3), T_{C1} and T_{C3} are then given by

$$\begin{pmatrix} T_{C1} \\ T_{C3} \end{pmatrix} = \begin{pmatrix} (u_O - u_P)R_{31} - fR_{11} & (u_O - u_P)R_{33} - fR_{13} \\ (v_O - v_P)R_{31} - fR_{21} & (v_O - v_P)R_{33} - fR_{23} \end{pmatrix}^{-1} \begin{pmatrix} (u_O - u_P)R_{32} - fR_{12} \\ (v_O - v_P)R_{32} - fR_{22} \end{pmatrix}. \quad (15)$$

Step 5. Refine the parameters by optimization: As mentioned earlier in Step 2, the v coordinate of all pixels in the original image are first multiplied by a scale factor $1/a$. Suppose principal point and aspect ratio computed in the previous steps are (u_P, v_P) and 1 , it can be easily verified that those values of the original camera are $(u_P, v_P \cdot a)$ and a . All other parameters remain the same.

Up to now, we assume that a is known, $s = 0$, V_Y , V_L (automatically computed from walking motion) and user's inputs $\{O, X_1, X_2, Z_1, Z_2\}$ contain no errors. But, this is not always the case in practice. By taking the errors into account, we can further refine the results. We apply an optimization approach here. Suppose the head and feet positions of N vertical line segments are (u_{h_i}, v_{h_i}) , (u_{f_i}, v_{f_i}) , $i = 1, 2, \dots, N$. The X and Z world coordinates of N feet positions can be computed by solving the projection equations in (4). If feet positions are $(X_i, 0, Z_i)$, head positions should be (X_i, H, Z_i) . H is the height of the walking human. We project (X_i, H, Z_i) back onto the $2D$ image. Suppose the reprojected head positions are (u'_{h_i}, v'_{h_i}) , $i = 1, 2, \dots, N$ and the sum of the square distance between the original and these reprojected head positions is $Dist = \sum_{i=1}^N ((u_{h_i} - u'_{h_i})^2 + (v_{h_i} - v'_{h_i})^2)$. $Dist$ depends on the computed projection matrix M and, thus, reflects the accuracy of M . Therefore, the result can be refined by finding M that minimizes $Dist$. We do not have to search all 12 elements of M ;

$Dist$ depends on only three variables: M_{31} , M_{32} , and M_{34} . The other variables can be determined by the following constraints.

$$\begin{aligned} M_{31}^2 + M_{32}^2 + M_{33}^2 &= 1, \\ M_{11}/M_{31} = u_{V_X}, M_{21}/M_{31} = v_{V_X}, M_{12}/M_{32} = u_{V_Y}, M_{22}/M_{32} = v_{V_Y}, \\ M_{13}/M_{33} = u_{V_Z}, M_{23}/M_{33} = v_{V_Z}, M_{14}/M_{34} = u_O, M_{24}/M_{34} = v_O. \end{aligned} \quad (16)$$

So, the problem can be stated as computing $\arg \min_{(M_{31}, M_{32}, M_{34})} Dist(M_{31}, M_{32}, M_{34})$. We use Levenberg-Marquardt, a widely used nonlinear minimization algorithm to solve this problem. The initial values of M_{31} , M_{32} , and M_{34} are given by the previous steps. Intrinsic and extrinsic parameters are then recovered using the method described in [13].

Theoretically, this optimization procedure cannot guarantee the reduction of true reprojection error because true $2D$ - $3D$ correspondences are unknown. In addition, the nonlinear optimization algorithm (Levenberg-Marquardt) may only find a local minimum. Nonetheless, our experiments show that results are improved after applying this procedure.

4 EXPERIMENTAL RESULTS

For our experiment, six scenes were shot from three places with two viewing angles each. Each scene contains walking sequences of two subjects. Fig. 5 shows one detected leg-crossing frames of each scene. Each video was recorded in Microsoft DV (NTSC) format. The resolution is 360×240 and the aspect ratio is 0.9. To get ground truth, we measured some $3D$ points and used the linear method [6] to compute the ground truth projection matrices. The actual number of $3D$ points measured in those six scenes is 20, 20, 16, 16, 20, 24, respectively.

4.1 Evaluation of Head/Feet Extraction

First, we evaluate the effectiveness of our approach to extract the head and feet locations and, thus, the validity of using such locations as vertical poles. As it is difficult to obtain real "vertical poles" at the locations that the human passed, we use the following criteria to provide an approximate yet meaningful evaluation. For each pair of head and feet locations, we place a **virtual** vertical pole on the

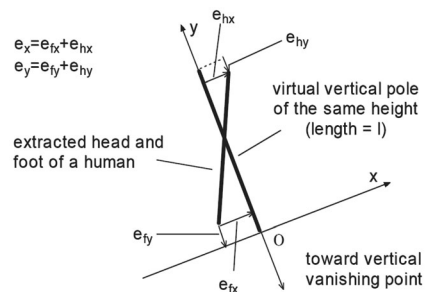


Fig. 6. Definition of evaluation errors of head/feet extraction: e_x reflects the deviation from being vertical; e_y reflects the deviation from the ground truth height.

TABLE 1
Statistics of Evaluation Errors in Head/Feet Extraction (in Percentage)

Seq. #	1.a	1.b	2.a	2.b	3.a	3.b	4.a	4.b	5.a	5.b	6.a	6.b	all
mean(ex)	3.25	2.79	1.08	1.96	1.71	3.04	1.94	1.82	0.13	-1.19	-2.11	-2.14	0.46
median(ex)	3.88	2.84	1.61	1.83	1.92	3.72	2.5	2.16	-0.08	-1.12	-1.3	-1.56	0.46
std(ex)	4.14	4.82	4.51	4.16	4.8	4.98	5.2	4.68	3.21	3.72	5.09	5.27	4.97
mean(ey)	6.82	7.48	3.02	1.57	2.7	3.85	5.39	5.71	-4.73	-7.65	-7.36	-8.27	-1.04
median(ey)	6.95	6.94	2.91	1.11	2.67	3.92	4.96	5.55	-4.4	-8.09	-6.89	-8.19	-0.44
std(ey)	2.71	3.01	2.63	2.29	2.85	2.81	2.94	3.76	2.86	3.63	4.05	4.63	6.97

TABLE 2
Calibration Results

Seq. #	f (pixel)	a	(u_p, v_p) (pixel)	s (pixel)	tilt (deg)	pan (deg)	roll (deg)	T_c (inch)	μ_{err} (pixel)	σ_{err} (pixel)
1.linear	408.79	0.915	(172.96, 111.92)	2.97	-13.5447	-38.9268	-4.3201	(191.84, -120.28, -131.81)	0.763	0.326
1.a	420.37	0.900	(212.51, 76.90)	0.00	-7.3912	-35.1911	-3.5122	(205.73, -119.61, -139.11)	3.980	1.520
1.a+opt	425.18	0.906	(197.53, 77.91)	1.62	-7.5287	-36.2052	-3.4893	(202.76, -118.61, -142.34)	2.807	1.567
1.b	417.53	0.900	(201.52, 85.73)	0.00	-8.5428	-35.8500	-3.4435	(202.95, -119.23, -139.25)	3.519	1.646
1.b+opt	419.26	0.904	(195.78, 86.40)	0.65	-8.6173	-36.2453	-3.4320	(198.54, -116.72, -138.20)	2.948	1.629
2.linear	447.10	0.902	(182.49, 82.56)	0.19	-5.4889	26.2357	-1.3465	(-264.57, -75.80, -276.17)	0.725	0.295
2.a	420.09	0.900	(145.76, 100.49)	0.00	-8.1360	22.7522	-0.8251	(-269.20, -75.61, -262.21)	5.113	2.883
2.a+opt	430.29	0.912	(156.55, 104.93)	-1.78	-8.4511	23.4913	-0.8193	(-261.39, -72.77, -263.87)	2.657	1.099
2.b	460.26	0.900	(202.59, 73.24)	0.00	-3.5351	27.4905	-1.2777	(-269.66, -76.23, -298.42)	5.551	2.020
2.b+opt	467.23	0.927	(212.78, 75.83)	-1.28	-3.6956	28.1208	-1.2433	(-264.11, -72.91, -300.22)	3.135	2.389
3.linear	411.49	0.909	(160.71, 98.50)	-3.32	-17.0398	15.3896	-2.7559	(5.81, -136.10, -216.56)	0.816	0.417
3.a	416.84	0.900	(173.99, 59.92)	0.00	-10.3018	17.1830	-3.4664	(0.02, -136.23, -227.64)	3.164	1.635
3.a+opt	409.74	0.905	(169.31, 58.07)	0.58	-10.1815	16.8450	-3.4435	(0.02, -132.82, -218.85)	2.408	2.023
3.b	437.04	0.900	(164.12, 61.45)	0.00	-10.9951	15.0287	-2.1486	(2.55, -135.61, -231.75)	4.000	2.735
3.b+opt	424.46	0.917	(157.11, 59.60)	0.76	-10.8747	14.5531	-2.1085	(2.63, -137.06, -231.15)	1.698	1.322
4.linear	440.87	0.908	(166.15, 106.27)	-2.76	-11.1899	39.2934	-1.5355	(-142.55, -92.07, -121.10)	0.703	0.293
4.a	437.17	0.900	(157.87, 68.18)	0.00	-5.9702	39.0299	-2.0283	(-144.62, -91.72, -119.10)	4.486	1.634
4.a+opt	440.98	0.881	(177.37, 67.78)	-1.35	-5.8957	40.3019	-2.0741	(-140.77, -92.87, -121.30)	2.572	1.497
4.b	438.73	0.900	(196.63, 73.92)	0.00	-6.3197	42.0093	-2.4694	(-140.06, -91.56, -124.15)	3.914	1.531
4.b+opt	437.13	0.908	(184.00, 73.88)	1.08	-6.3541	41.1785	-2.4465	(-142.38, -91.10, -122.57)	1.632	1.357
5.linear	431.78	0.854	(211.25, 84.63)	-0.50	-16.9252	79.6812	-0.9397	(-568.16, -329.22, -135.92)	0.481	0.262
5.a	426.87	0.900	(216.20, 123.52)	0.00	-21.1536	80.9188	-1.6100	(-614.63, -328.89, -142.36)	5.970	1.266
5.a+opt	468.13	0.894	(199.53, 156.93)	7.00	-23.5658	79.8188	-1.6215	(-627.41, -296.85, -129.34)	3.156	1.723
5.b	485.66	0.900	(181.48, 133.57)	0.00	-20.8098	77.4754	-0.5329	(-689.23, -329.02, -141.09)	4.879	1.418
5.b+opt	505.70	0.919	(169.78, 159.49)	4.53	-22.4599	76.7592	-0.5214	(-699.16, -305.93, -135.10)	3.791	2.257
6.linear	447.97	0.891	(155.27, 84.95)	-0.11	-29.1063	74.4559	9.6028	(-603.83, -464.41, -130.78)	0.426	0.233
6.a	411.55	0.900	(159.42, 112.30)	0.00	-31.1689	75.0689	8.8178	(-599.53, -434.44, -118.31)	6.213	1.795
6.a+opt	396.49	0.897	(165.82, 83.41)	-7.03	-28.9859	75.9685	8.8751	(-523.58, -415.99, -110.25)	3.757	1.443
6.b	408.73	0.900	(175.51, 101.96)	0.00	-30.1261	77.7962	7.3052	(-588.75, -430.79, -117.19)	6.033	1.891
6.b+opt	395.77	0.894	(179.85, 77.77)	-4.95	-28.2583	78.4093	7.3625	(-518.55, -410.53, -108.89)	3.850	1.513

ground with the known height of the walking human and optimize its location on the ground as

$$\mathbf{x}^* = \arg \min_{\mathbf{x}} ((\mathbf{h}_i - \mathbf{t}(\mathbf{x}))^T \Sigma_{\mathbf{h}_i}^{-1} (\mathbf{h}_i - \mathbf{t}(\mathbf{x})) + (\mathbf{f}_i - \mathbf{b}(\mathbf{x}))^T \Sigma_{\mathbf{f}_i}^{-1} (\mathbf{f}_i - \mathbf{b}(\mathbf{x}))),$$

where \mathbf{t} and \mathbf{b} are the image locations of the top and the bottom of the pole projected by the ground truth projection matrix. We compute the statistics of the deviations of the extracted head and feet locations from the top and the bottom of the pole, as shown in Fig. 6. We parameterize the error in the local coordinate system whose y axis aligns with the vertical direction (towards the vertical vanishing point). Therefore, e_x reflects the deviation from being vertical and e_y reflects the deviation from a constant height. We retain the signs of the errors to show the trend of bias. We normalize

the error by the length of the image pole length to make different observations more comparable, $\tilde{e}_i = [\tilde{e}_{i,x}, \tilde{e}_{i,y}] = [e_{i,x}/l_i, e_{i,y}/l_i]$. The mean, median, and standard deviation of the error for all the 12 sequences is summarized in Table 1. *Seq.# i.a* and *i.b* denote the sequences of the first and second subject in the i th scene.

The statistic of e_x shows that the human are roughly vertical. There are slight biases in different sequences. The bias usually comes from the leaning of the body when walking. The statistic of e_y shows larger bias but smaller noise.

4.2 Evaluation of the Calibration Algorithm

Table 2 shows the computed camera parameters, the mean, μ_{err} , and the standard deviation, σ_{err} , of reprojection error given 2D-3D correspondences which were used to get ground truth. The five rows for each scene denote the linear method [6] (Parameters were recovered using the method described in [13]) and our method (with

and without optimization) on the first and second subject, respectively.

Results show that our method is robust *with regard to* various viewing angles. Our experiments cover a wide range of tilt angle from 5 degrees (See Fig. 5b, shot on a tripod on the ground) to 30 degrees (See Fig. 5f, shot from the fourth floor of a building). The range covers tilt angle of a typical camera setup for applications such as visual surveillance. As expected, the result is more accurate in the midrange, but it is also acceptable when the tilt angle is either very small or large.

Results on different walking persons (three in total in all experiments) show that our method is also insensitive *with regard to* various subjects. Note the backpack in Fig. 5c, which changes the walking human's silhouette. But, it did not affect the result much.

The differences between results before and after applying the optimization can be clearly seen from Table 2. Although errors of some parameters become larger after applying the optimization, the average projection error (μ_{err}) decreases in all experiments. In some cases (e.g., 2.b and 3.a), δ_{err} turns out to be larger after applying the optimization. This is because, in these cases, the reprojected points are homogeneously biased *with regard to* the ground truth before optimization. Although the overall bias can be compensated for by the optimization procedure, some points are not affected by the optimization, which results in a larger variance among the reprojection errors of all points.

The proposed method has been successfully tested on many other data sets and has been used as a component in our tracking systems ([15], [16]).

5 CONCLUSION

We propose a self-calibration method based on the geometric properties of vanishing points. Our method differs from other methods that use parallel lines in three mutually orthogonal directions in that we use only vertical line segments of the same height and minimal amount of information available in the scene (two auxiliary lines) to obtain all three vanishing points. We specifically present a method to obtain the needed line segments by detecting the head and feet positions of a walking human in his leg-crossing phases. The results are satisfactory *with regard to* various viewing angles and subjects for the task of human motion analysis.

ACKNOWLEDGMENTS

This research was supported, in part, by the Advanced Research and Development Activity of the US government under contract No. MDA904-03-C1786. The authors would like to thank Bo Wu and Xuefeng Song for their help in collecting data.

REFERENCES

- [1] B. Bose and E. Grimson, "Ground Plane Rectification by Tracking Moving Objects," *Proc. IEEE Workshop Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, pp. 94-101, 2003.
- [2] B. Caprile and V. Torre, "Using Vanishing Points for Camera Calibration," *Int'l J. Computer Vision*, vol. 4, pp. 127-140, 1990.
- [3] R. Cipolla, T. Drummond, and D.P. Robertson, "Camera Calibration from Vanishing Points in Images of Architectural Scenes," *Proc. British Machine Vision Conf.*, vol. 2, pp. 382-391, 1999.
- [4] A. Criminisi, I. Reid, and A. Zisserman, "Single View Metrology," *Int'l J. Computer Vision*, vol. 40, no. 2, pp. 123-148, 2000.
- [5] J. Deutscher, M. Isard, and J. MacCormick, "Automatic Camera Calibration from a Single Manhattan Image," *Proc. European Conf. Computer Vision*, pp. 175-188, 2002.
- [6] O. Faugeras, *Three Dimensional Computer Vision: A Geometric Viewpoint*. MIT Press, 1993.
- [7] M.A. Fischler and R.C. Bolles, "Random Sample Consensus: A Paradigm for Model Fitting with Application to Image Analysis and Automated Cartography," *Comm. ACM*, vol. 24, no. 6, pp. 381-395, 1981.

- [8] M. Isard and J. MacCormick, "BraMBLE: A Bayesian Multiple-Blob Tracker," *Proc. Int'l Conf. Computer Vision*, vol. 2, pp. 34-41, 2001.
- [9] D. Liebowitz, A. Criminisi, and A. Zisserman, "Creating Architectural Models from Images," *Proc. EuroGraphics*, vol. 18, pp. 39-50, 1999.
- [10] A. Nakatsuji, S. Takahashi, Y. Sugayay, and K. Kanatani, "Stabilizing the Focal Length Computation for 3-D Reconstruction from Two Uncalibrated Views," *Proc. Asian Conf. Computer Vision*, vol. 1, pp. 1-6, 2004.
- [11] C. Stau, K. Tieu, and L. Lee, "Robust Automated Planar Normalization of Tracking Data," *Proc. IEEE Workshop Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, pp. 1-8, 2003.
- [12] C.R. Wren, A. Azarbayejani, T. Darrell, and A. Pentland, "Pfinder: Real-Time Tracking of the Human Body," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 780-785, July 1997.
- [13] Z. Zhang, "A Flexible New Technique for Camera Calibration," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 22, no. 11, pp. 1330-1334, Nov. 2000.
- [14] Z. Zhang, "Camera Calibration with One-Dimensional Objects," *Proc. European Conf. Computer Vision*, vol. 4, pp. 161-174, 2002.
- [15] T. Zhao, R. Nevatia, and F. Lv, "Segmentation and Tracking of Multiple Humans in Complex Situations," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, vol. 2, pp. 194-201, 2001.
- [16] T. Zhao, R. Nevatia, "Tracking Multiple Humans in Crowded Environment," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, vol. 2, pp. 406-413, 2004.

► For more information on this or any other computing topic, please visit our Digital Library at www.computer.org/publications/dlib.