Video-based Height Measurements of Multiple Moving Objects

Mingxin Jiang[1,2], Hongyu Wang[1] , Tianshuang Qiu[1]
[1] School of Information & Communication Engineering,Dalian University of Technology, Dalian, Liaoning, 116024, China.
[2] School of Information & Communication Engineering, Dalian Nationalities University, Dalian, Liaoning, 116600, China
[e-mail: jiangmingxin@126.com,e-mail:whyu@dlut.edu.cn,qiutsh@dlut.edu.cn]
*Corresponding author: Hongyu Wang

Received October 10, 2013; revised December 23, 2014; revised February 17, 2014; accepted March 12, 2014; published September 30, 2014

Abstract

This paper presents a novel video metrology approach based on robust tracking. From videos acquired by an uncalibrated stationary camera, the foreground likelihood map is obtained by using the Codebook background modeling algorithm, and the multiple moving objects are tracked by a combined tracking algorithm. Then, we compute vanishing line of the ground plane and the vertical vanishing point of the scene, and extract the head feature points and the feet feature points in each frame of video sequences. Finally, we apply a single view mensuration algorithm to each of the frames to obtain height measurements and fuse the multi-frame measurements using RANSAC algorithm. Compared with other popular methods, our proposed algorithm does not require calibrating the camera, and can track the multiple moving objects when occlusion occurs. Therefore, it reduces the complexity of calculation and improves the accuracy of measurement simultaneously. The experimental results demonstrate that our method is effective and robust to occlusion.

Keywords: projective geometric constraint; Codebook; height measurements; multi-target tracking

This research described in this paper was supported by National Natural Science Foundation of China(61172058, 61403060). This work was supported by Project funded by China Postdoctoral Science Foundation (2014M551081).

http://dx.doi.org/10.3837/tiis.2014.09.014

1. Introduction

Metrology, the measurement of real world metrics, has been investigated extensively in computer vision for many applications. The technique of measuring the geometric parameters of objects from video has been developed as an interesting issue in computer vision field in recent years [1-2]. With the increasing use of video surveillance systems [3], more and more crimes and incidents have been captured on video. When the incidents have been captured, we need to gain an understanding of the events or identify a particular individual.

As height is an important parameter of a person, some methods have been presented for estimating height information from video [4-5]. They can be roughly divided into two categories: absolute measurement and relative measurement. Absolute measurement requires fully calibrating camera, which is a complicated process [6]. Relative measurement only requires the minimal calibration. Guo and Chellappa [7] presented a video metrology approach using an uncalibrated single camera that is either stationary or in planar motion. This paper also leverages object motion in videos to acquire calibration information for measurement. No constant velocity motion is assumed. Furthermore, it incorporates all the measurements from individual video frames to improve the accuracy of final measurement.

Several automatic mensuration algorithms have been developed to take advantage of tracking results from video sequences. Renno et al. [8] used projected sizes of pedestrians to estimate the vanishing line of a ground plane. Bose and Crimson [9] proposed a method that uses constant velocity trajectories of objects to derive vanishing lines for recovering the reference plane and planar rectification. The basic idea of their algorithm is to use an additional constraint brought by the constant-velocity assumption, which is not always available in surveillance sequences. Shao et al. [10] proposed a minimal-supervised algorithm based upon monocular videos and uncalibrated stationary cameras. The author recovered the minimal calibration of the scene based upon tracking moving objects, then applied the single view metrology algorithm to each frame, and finally fused the multi-frame measurements using the LMedS as the cost function and the RMSA as the optimization algorithm.

However, most of the existing approaches are direct extension of image-based algorithms, which have not considered the occlusions between objects and lack robustness. Reliable tracking of multiple objects in complex situations is a challenging visual surveillance problem since the high density of objects results in occlusion. When occlusion between multiple objects is common, it is extremelly difficult to perform the tasks of height measurements of objects.

In this paper, we propose a new method for height measurements of multiple objects based on rubust tracking. Firstly, the foreground likelihood map is obtained by using the Codebook background modeling algorithm. Secondly, tracking of multiple objects are performed by a combined tracking algorithm. Then, the vanishing line of the ground plane and the vertical vanishing point are computed, and the head feature points and the feet feature points are extracted in each frame of video sequences. Finally, we obtain height measurements of multiple objects according to the projective geometric constraint, and the multiframe measurements are fused using RANSAC algorithm.

Compared with other popular methods,our proposed algorithm does not require calibrate the camera, and can track the multiple moving objects in crowded scenes.Therefore, it reduces the complexity and improves the accuracy simultaneously. The experimental results demonstrate that our method is effective and robust in the occlusion case..

The organization of this paper is as follows. In Section 2, we introduce the multi-target detecting and tracking algorithm.Section 3 addresses video-based height measurements of multiple moving objects. Section 4 presents experimental results. Section 5 concludes this paper.

2. Multi-Target Detecting and Tracking Algorithm

2.1 Multi-Target Detecting Algorithm

The capability of extracting moving objects from a video sequence captured using a static camera is a typical first step in visual surveillance. A common approach for discriminating moving objects from the background is detection by background subtraction [11-12]. The idea of background subtraction is to subtract or difference the current image from a reference background model. The subtraction identifies non-stationary or new objects. The generalized mixture of Gaussians (MOG) has been used to model complex, non-static backgrounds. MOG does have some disadvantages. Backgrounds having fast variations are not easily modeled with just a few Gaussians accurately, and it may fail to provide sensitive detection.

In this paper, codebook algorithm has been used to model backgrounds. The algorithm is an adaptive and compact background model that can capture structural background motion over a long period of time under limited memory. This allows us to encode moving backgrounds or multiple changing backgrounds. At the same time, the algorithm has the capability of coping with local and global illumination changes.

A quantization/clustering technique is adopted to construct a background model in the codebook algorithm. Samples at each pixel are clustered into the set of codewords. The background is encoded on a pixel by pixel basis.

Let $\mathbf{X}=\{\mathbf{x}_1,\mathbf{x}_2,...,\mathbf{x}_N\}$ be a training sequence for a single pixel consisting of N RGB-vectors. Let $\mathbf{C}=(\mathbf{c}_1,\mathbf{c}_2,...,\mathbf{c}_L)$ represent the codebook for the pixel consisting of L codewords. Each pixel has a different codebook size based on its sample variation. Each codeword $\mathbf{c}_i(i=1,...,L)$ consists of an RGB vector $\mathbf{v}_i=(\overline{R_i},\overline{G_i},\overline{B_i})$ and a 6-tuple $\mathbf{aux}_i=(\check{I}_i,\hat{I}_i,f_i,\lambda_i,p_i,q_i)$.The tuple $\mathbf{aux}_i$ contains intensity (brightness) values and temporal variables described below.

$\check{I}_i,\hat{I}_i$: the min and max brightness accepted for the codeword *i* respectively;

f_i : the frequency for the codeword *i* occurring;

λ_i : the maximum negative run-length (MNRL) defined as the longest interval during the training period that the codeword has NOT recurred;

p_i , q_i :the first and last access times, respectively, that the codeword has occurred.

In the training period, each value, $\mathbf{x}_t$, sampled at time t is compared to the current codebook to determine which codeword $\mathbf{c}_m$ (if any) it matches (m is the index of matching codeword). We use the matched codeword as the sample's encoding approximation. To determine which codeword will be the best match, we employ a color distortion measure and brightness bounds.

When we have an input pixel $\mathbf{x}_t = (R, G, B)$ and a codeword $\mathbf{c}_i$ where $\mathbf{v}_i = (\overline{R_i}, \overline{G_i}, \overline{B_i})$, $\|\mathbf{x}_t\|^2 = R^2 + G^2 + B^2$, $\|\mathbf{v}_i\|^2 = \overline{R_i}^2 + \overline{G_i}^2 + \overline{B_i}^2$, $\langle \mathbf{x}_t, \mathbf{v}_i \rangle^2 = (\overline{R_i}R + \overline{G_i}G + \overline{B_i}B)^2$.

The color distortion can be calculated by

$$colordist(\mathbf{x}_t, \mathbf{v}_i) = \frac{\|\mathbf{x}_t\|^2 \|\mathbf{v}_i\|^2 - \langle \mathbf{x}_t, \mathbf{v}_i \rangle^2}{\|\mathbf{v}_i\|^2} \tag{1}$$

The logical brightness function is defined as

$$brightness(I, <\check{I}, \hat{I}>) = \begin{cases} true, if & I_{low} \le \|\mathbf{x}_t\| \le I_{hi} \\ false, & otherwise \end{cases} \tag{2}$$

The detailed algorithm of constructing codebook is given in [11].

We segment foreground using subtracting the current image from the background model. When we have a new input pixel $x_i = (R, G, B)$ and its codebook M. The subtraction operation $BGS(x_i)$ for the pixel is defined as:

Step1. Compute the brightness $I = R + G + B$. Define a boolean variable $match$.

Step2. Find the codeword C_m matching to x based on two conditions:

$$colordist(x_i, v_m) \le \varepsilon \tag{3}$$

$$brightness(I, <\check{I}_m, \hat{I}_m>) = true \tag{4}$$

if the codeword C_m is found, let $match = 1$, else let $match = 0$.

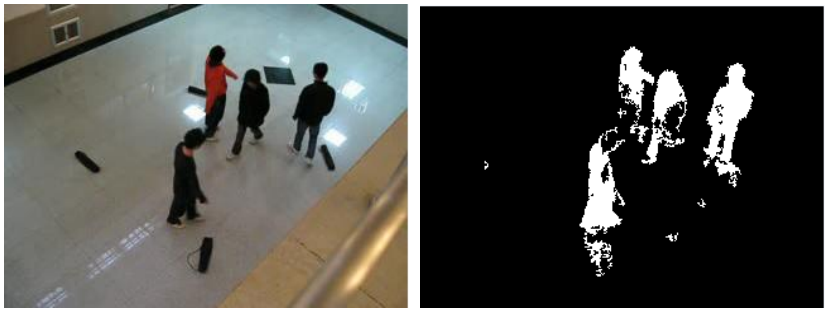
Step3. Determine the foreground moving object pixel:

$$BGS(x_i) = \begin{cases} foreground & match = 0 \\ backround & match = 1 \end{cases} \tag{5}$$

Step4. The likelihood of observation x_i belonging to the foreground:

$$L(x_i) = \begin{cases} 1, & \check{I}_m \le \|x_i\| \le \hat{I}_m \\ 0, & otherwise \end{cases} \tag{6}$$

Fig. 1 depicts the results of comparison of foreground likelihood maps obtained using different methods for an indoor data set. **Fig. 1(a)** is an image extracted from an indoor video. **Fig. 1(b)** depicts the foreground likelihood map of the image using mixture of Gaussians algorithm. **Fig. 1(c)** depicts the foreground likelihood map of the image using Codebook-based method.



(a) Original video (b) Mixture of Gaussians

(c)Codebook

Fig. 1. Comparison of foreground likelihood maps obtained using different methods

2.2 Multi-Target Tracking Algorithm

Tracking multiple people accurately in cluttered and crowded scenes is a challenging task primarily due to occlusion between people [13-14]. Particle filter can work well when the object gets an occlusion, but it has difficulty in satisfying the requirement of real-time computing. Meanshift can solve this problem easily, while it has poor robustness during mutual occlusion. Aiming at all above problems, this section proposes a robust multi-target tracking algorithm by combining the particle filter with meanshift method.

Particle filters, provide an approximative Bayesian solution to discrete time recursive problem by updating an approximative description of the posterior filtering density [15].

At time k, when a measurement z_k becomes available, $z_{1:k} = \{z_1, z_2, ..., z_k\}$. Assume that probability distribution function $p(x_{k-1}|z_{1:k-1})$ is available at time $k-1$. According to the Bayes' rule, the posterior probability function of the state vector can be calculated using the following equations.

$$p(x_k|z_{1:k-1}) = \int p(x_k|x_{k-1})\, p(x_{k-1}|z_{1:k-1}) dx_{k-1} \tag{7}$$

This is the prior of the state x_k at time k without the knowledge of the measurement z_k, i.e. the probability given only previous measurements. Update step combines likelihood of current measurement with predicted state.

$$p(x_k|z_{1:k}) = \frac{p(z_k|x_k)\, p(x_k|z_{1:k-1})}{p(z_k|z_{1:k-1})} \tag{8}$$

$p(z_k|z_{1:k-1})$ is a normalizing constant. It can be calculated by:

$$p(z_k|z_{1:k-1}) = \int p(z_k|x_k)\, p(x_k|z_{1:k-1}) dx_k \tag{9}$$

Because $p(z_k|z_{1:k})$ is a constant, (8) can be written as:

$$p(x_k|z_{1:k}) \propto p(z_k|x_k)p(x_k|z_{1:k-1}) \tag{10}$$

Supposing that at time step k there is a set of particles, $\{x_k^i, i=1,...,N\}$ with associated weights $\{\omega_k^i, i=1,...,N\}$ randomly drawn from importance sampling, where N is the total number of particles. The weight of particle i can be defined as:

$$\omega_k^i \propto \omega_{k-1}^i \frac{p(x_k^i|x_{k-1}^i)p(z_k|x_k^i)}{q(x_k^i|x_{k-1}^i, z_{1:k})} \tag{11}$$

We use the transition prior $p(x_k|x_{k-1})$ as the importance density function $q(x_k^i|x_{k-1}^i, z_{1:k})$. Then, we can simplify (11) as:

$$\omega_k^i \propto \omega_{k-1}^i p(z_k|x_k^i) \tag{12}$$

Furthermore, if we use Grenander's factored sampling algorithm, Eq.(16) can be modified as:

$$\omega_k^i \propto p(z_k|x_k^i) \tag{13}$$

The particle weights then can be normalized by using:

$$\omega_k^{*i} \propto \frac{\omega_k^i}{\sum_{i=1}^N \omega_k^i} \tag{14}$$

to give a weighted approximation of the posterior density in the following form:

$$p(x_k|z_{1:k}) \approx \sum_{i=1}^N \omega_k^{*i} \delta(x_k - x_k^i) \tag{15}$$

where δ is the Dirac's delta function.

Meanshift algorithm was first analyzed in [16] and developed in [17]. Meanshift is a non-parametric statistical approach that seeks the mode of a density distribution in an iterative procedure [18]. Let X denote the current location, then its new location X' after one iteration is :

$$X' = \frac{\sum_{i=1}^M a_i \omega(a_i) g\left(\left\|\frac{a_i - X}{h}\right\|^2\right)}{\sum_{i=1}^M \omega(a_i) g\left(\left\|\frac{a_i - X}{h}\right\|^2\right)} \tag{16}$$

where $\{a_i, i=1,...,N\}$ are normalized points within the rectangle area specified by the current location X, $\omega(a_i)$ is the weight associated to each pixel a_i, and $g(x)$ is a kernel profile function, and h is window radius to normalize the coordinate a_i.

In our tracking algorithm, we assume that the dynamic of state transition corresponds to the following second order auto- regressive process.

$$x_k = Ax_{k-1} + Bx_{k-2} + C\mathrm{n}_k, \mathrm{n}_k \sim N(0, \Sigma) \tag{17}$$

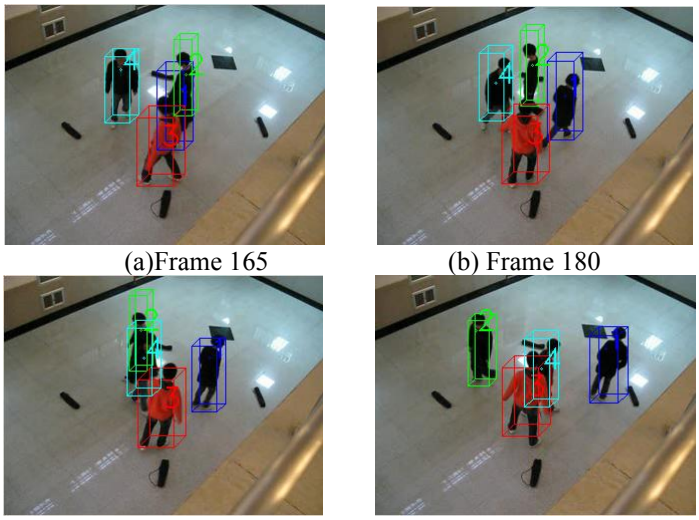
where A, B, C are the autoregression coefficients, n_k is the Gaussian noise .

We use HSV color histogram to build the observation model. Given the current observation z_k, the candidate color histogram $Q(x_k)$ is calculated on z_k in the region specified by x_k.

The similarity between $Q(x_k)$ and the reference color histogram Q^* by Bhattacharyya coefficient $d(,)$.The likelihood distribution is evaluated as

$$p(z_k|x_k) \propto e^{-\lambda d^2[Q^*,Q(x_k)]} \tag{18}$$

In our method, the meanshift algorithm is applied on every sample in sample set, this will greatly reduce the computational time of particle filtering. It might not be able to capture the true location of the objects during mutual occlusion. The particle filter can improve the robustness of the algorithm. We propagate particle $\{x_{k-1}^i, i=1,...,N\}$ according to the dynamic of state transition to get $\{\tilde{x}_k^i, i=1,...,N\}$. The samples set $\{\tilde{x}_k^i, i=1,...,N\}$ is the first transition to get $\{\bar{x}_k^i, i=1,...,N\}$ by meanshift according to Eq.(16). With meanshifted samples $\{\bar{x}_k^i, i=1,...,N\}$,we update their weights $\{\omega_k^i, i=1,...,N\}$ according to Eq.(14). The likelihood distribution $p(z_k|x_k^i)$ is given by Eq.(18). Then we resample $\{\bar{x}_k^i, \omega_k^i\}_{i=1,...N}$ and generate unweighted sample set $\{x_k^i, 1/N\}_{i=1,...N}$. In **Fig. 2** the tracking results are demonstrated for outdoor video sequences in different frames.



(a)Frame 165 (b) Frame 180

(c) Frame 195 (d) Frame 223

Fig. 2. Tracking results for test video sequences

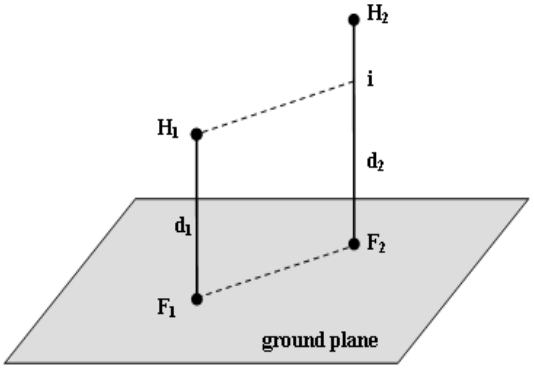
3. Video-based Height Measurements of Multiple Moving Objects

3.1 Projective Geometry

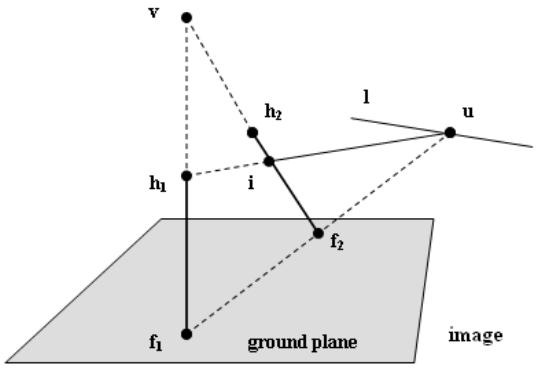
In this section, we introduce the main projective geometric ideas and notation that are required for understanding our measurement algorithm well. We use upper case letters to indicate points in the world system and the corresponding lower case letters for their images.

Fig. 3 shows the basic geometry of the scene. A line segment in space, orthogonal to the ground plane and identified by its top point H_i and base point F_i, is denoted by H_iF_i, and its length is denoted by $d(H_i,F_i)$. H_iF_i is projected onto the image plane as the line segment

$h_i f_i$ .The line l is the vanishing line of the ground plane, and v the vertical vanishing point. Given one reference height $d(H_1, F_1) = d_1$ in the scene, the height of any object on ground plane(e.g. d_2)) can be measured using geometry method shown in **Fig. 3(b)**.



(a)3D geometry



(b)image geometry

Fig. 3. Basic geometry of the scene

The measurement is achieved with two steps. At the first stage, we map the length of the line segment $h_1 f_1$ onto the other $h_2 f_2$. The intersection of the line through the two base points f_1 and f_2 with the vanishing line l determines the point u, and the intersection of the line through h_1 and u with the line through v and f_2 determines the point i. Because v and u are vanishing points, $h_1 f_1$ and $h_1 i$ are parallel to if_2 and $f_1 f_2$ respectively. h_1, i, f_2, and f_1 forms a parallelogram with $d(h_1, f_1) = d(i, f_2)$. We now have four collinear points v, h_2, i, f_2 on an imaged scene line and thus there is a cross ratio available. The distance ratio $d(h_2, f_2) : d(i, f_2)$ is the computed estimate of $d_2 : d_1$ by applying a 1-D projective transformation. At the second stage, we compute the ratio of length on the imaged scene line using cross ratio [19].

The ratio between two line segments $h_2 f_2$ and if_2 can be written by:

$$r = \frac{d(h_2, f_2)}{d(i, f_2)} = \frac{d(h_2 f_2) d(v, i)}{d(v, h_2) d(i, f_2)} \tag{19}$$

with $d_2 = rd_1$. The height of any object can be measured using this method.

With the assumption of perfect projection, e.g. with a pinhole camera, a set of parallel lines in the scene is projected onto a set of lines in the image which meet in a common point. This point of intersection, perhaps at infinity, is called the vanishing point. Different approaches are adopted to detect vanishing points for the reference direction, according to the environments of video data sets.

In pinhole camera model, the vanishing line of the ground plane can be determined as the line through two or more vanishing points of the plane. If we have N vertical poles of same height in the perspective view, the vertical vanishing point V_Y can be computed just by finding the intersection of two (or more) poles. And the vanishing line of the ground plane V_L is the line consisted by the points of intersection of the lines connecting the top and bottom of the poles. Thus, we can fix the vanishing line through three (or more) non-coplanar poles. In this paper, we denote the poles by $\{(t_i, b_i)\}_{i=1,2...N}$, where t_i, b_i represent the image positions of the top and bottom of the poles, respectively. $\{(\Sigma_{t_i}, \Sigma_{b_i})\}_{i=1,2,...N}$ are the associated covariance matrices. V_Y can be fixed by finding the point v that minimizes the sum of distances from t_i and b_i to the line linking m_i and v. Where m_i is the midpoint of t_i and b_i, (w_i, b_i) is the line determined by m_i and v.

$$V_Y = \arg\min_{v} \sum_{i=1}^{N} \left(\frac{\left| w_i^T h_i - k_i \right|}{(w_i^T \Sigma_{t_i} w_i)^{1/2}} + \frac{\left| w_i^T b_i - k_i \right|}{(w_i^T \Sigma_{b_i} w_i)^{1/2}} \right) \tag{20}$$

V_L can be computed by

$$(w_{V_L}, k_{V_L}) = \arg\min_{(w,k)} \sum_{i=1}^{N} \left(\frac{\left| w^T x_i - k \right|}{(w^T \Sigma_i w)^{1/2}} \right) \tag{21}$$

where w_{V_L} is the unit vector of V_L and b_{v_L} is a point on vanishing line.

The point x_i is the intersection of line $t_j b_j$ and line $t_k b_k$. The covariance matrix Σ_i of x_i can be computed by using Jacobian as

$$\Sigma_i = J \cdot diag(\Sigma_{t_j}, \Sigma_{t_k}, \Sigma_{b_j}, \Sigma_{b_k}) \cdot J^T \tag{22}$$

where $J = \dfrac{\partial x_i}{\partial (t_j^T, t_k^T, b_j^T, b_k^T)^T}$.

3.2 Extracting head and feet feature points from moving objects

Humans are roughly vertical while they stand or walk. However, because human walking is an articulated motion, the shape and height of the human vary in different walking phases. As shown in **Fig. 4**, at the phase which the two legs cross each other, height of the human we measured from the video sequence is highest, and is also the most appropriate height to represent human's static height.

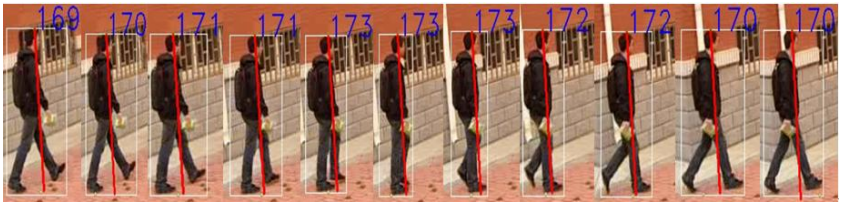


Fig. 4. The height of human varies periodically during walking cycle

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.

To every single frame t, the head feature point h_i^t of the object i $\{i = 1, 2, ...N\}$ can be obtained using the following steps.

Step 1. Construct the target likelihood matrix L_i^t corresponding to the foreground blobs B_i^t (w_i, h_i), where w_i and h_i denote the width and height of foreground blob B_i^t, respectively.

Step 2. Compute the covariance matrix C_i^t of target likelihood matrix L_i^t. The covariance matrix C_i^t can computed as

$$C_i^t(m,n) = E\{[L_i^t(m) - \overline{L}_i^t(m)][L_i^t(n) - \overline{L}_i^t(n)]\} \tag{23}$$

Where $L_i^t(m)$ and $L_i^t(n)$ denote the m and n column of foreground target matrix at frame t.

Step 3. Compute the first eigenvectors e_i^1 of covariance matrix C_t^i .The centroid and e_i^1 of the blob give the principal axis P_t^i of the target's body. The head feature point is assumed to be located on the principal axis.

Step 4. Project target blob B_t^i onto its corresponding principal axis P_t^i. Locate the head feature point h_t^i by finding the first end point along the principal axis whose projection count is above a threshold along the principal axis from the top to the bottom.

Humans are roughly vertical at different phase of a walking cycle. This means that the head feature point, the feet feature points and vanishing point are collinear. We obtain the feet feature points of target f_i by applying the collinear constraint. The f_i can be computed as $f_i = (h_i \times V_Y) \times l_{b(i)}$. h_i denotes head feature point of object i. V_Y denotes the vertical vanishing point. $l_{b(i)}$ denotes the bottom line of blob.

3.3 Multiple frame fusion

The measurements from multiple frames include outlier observations due to bad tracking errors, articulated motion, and occlusions. It makes using mean of the multiframe estimates not robust. The RANSAC technique has the well-known property of being less sensitive to

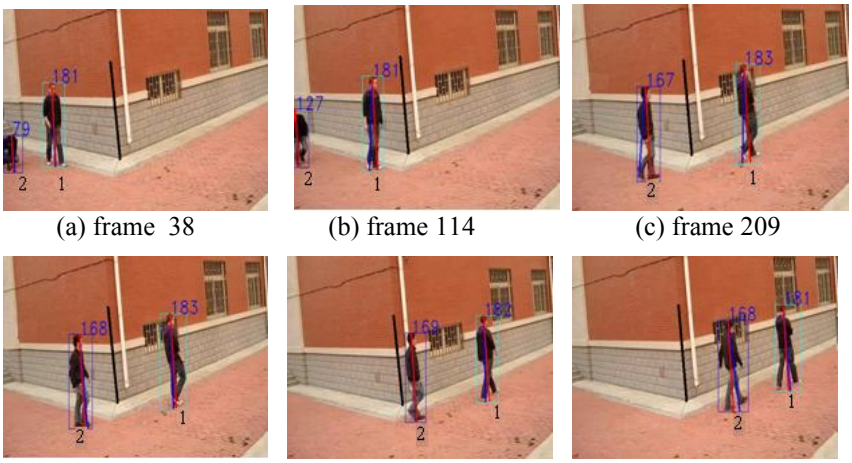
outliers. Thus, in this paper, we use RANSAC algorithm to estimate the height from a set of data contaminated by outliners.

4. Experimental Results and Analysis

In order to show the robustness of the height measurement algorithm discussed in this paper, we conducted several experiments with the data that we collected from stationary cameras under different environments. Moving objects include vehicles and humans. Given the limited space, in this section we only list two of them to show the experimental results and the forms of data statistics. The number of particles used for our method is 300 for all experiments.

The implementation of the algorithm is based on Windows 7 operating system and using MATLAB as the software platform. The configuration of the computer is AMD Athlon (TM) X2DualCore QL-62 2.00GHz,1.74GB memory.

The results of height measurements for test video 1 are shown in **Fig. 5**.



(a) frame 38 (b) frame 114 (c) frame 209

(d) frame 215 (e) frame 257 (f) frame 312

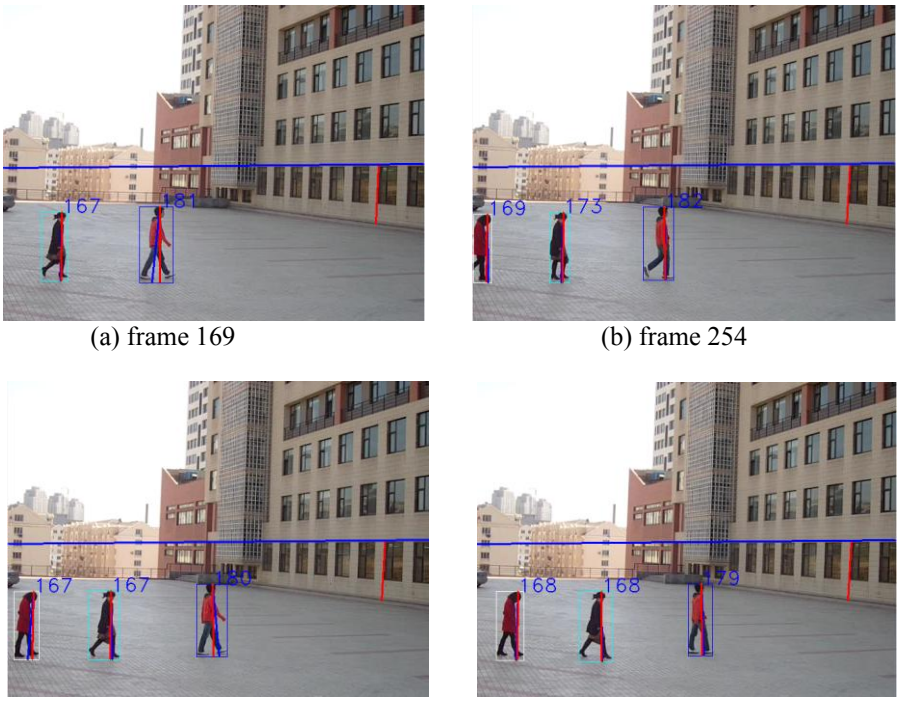
Fig. 5. The results of height measurement for test video 1

Statistics of the measured value for test video 1 are shown in **Table 1**.

Table 1. Statistics of the measured value for test video 1

Moving object	Actual height (cm)	The average measured value (cm)	The average measured value at leg-crossing phases (cm)	Variance at leg-crossing phases	Variance of [10]
object 1	181.5	178.2	181.8	0.42	2.39
object 2	168.0	145.4	168.3	0.51	2.62

The results of height measurement for test video 2 are shown in **Fig. 6**. The tracking blobs are object 1, object 2, object 3 from right to left respectively. The heights of objects are shown on the top of the blobs.



(a) frame 169 (b) frame 254

(c) frame 269 (d) frame 305

Fig. 6. The results of height measurement for test video 1

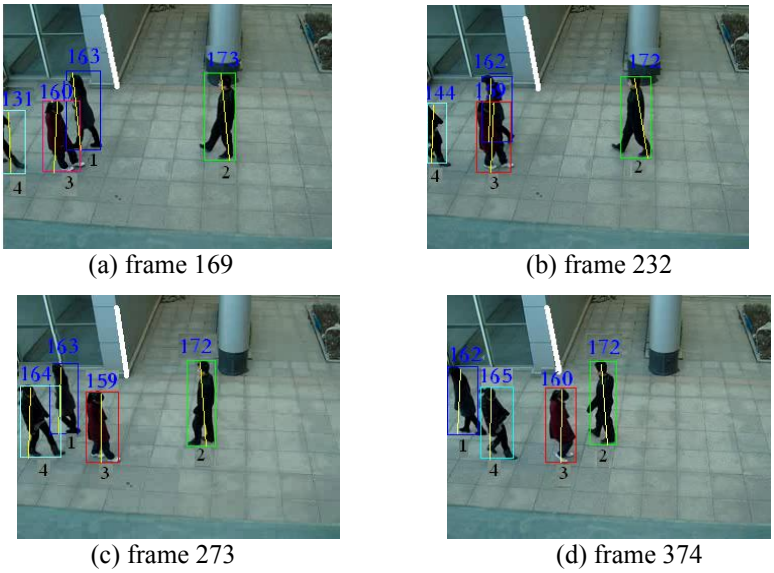
Statistics of the measured value for test video 1 are shown in **Table 2**.

Table 2. Statistics of the measured value for test video 2

Moving object	Actual height (cm)	The average measured value (cm)	The average measured value at leg-crossing phases (cm)	Variance at leg-crossing phases	Variance of [10]
object 1	180.0	180.4	182.3	0.51	2.12
object 2	168.5	169.1	171.6	0.48	2.23
object 3	168.0	168.3	169.8	0.44	2.13

From the experimental results, we can see that our algorithm shows better accuracy and robustness than algorithm proposed in [10].

The results of height measurement for test video 3 are shown in **Fig. 7.**



(a) frame 169 (b) frame 232

(c) frame 273 (d) frame 374

Fig. 7. The results of height measurement for test video 1

Statistics of the measured value for test video 1 are shown in **Table 3**.

Table 3. Statistics of the measured value for test video 3

Moving object	Actual height (cm)	*The average measured value (cm)*	The average measured value at leg-crossing phases (cm)	Variance at leg-crossing phases	Variance of [10]
object 1	162.0	*160.1*	162.3	0.41	2.34
object 2	172.5	*170.2*	172.9	0.41	2.33
object 3	159.5	*157.5*	159.8	0.45	2.45
object 4	165.0	*162.9*	165.4	0.43	2.40

From the experimental results, we can see that our proposed algorithm does not require calibrating the camera, and can track the multiple moving objects when occlusion occurs. Therefore, it reduces the complexity of calculation and improves the accuracy of measurement simultaneously.

5. Conclusion

We have presented a new algorithm for estimating height of multiple moving objects. We first compute the vanishing line of the ground plane and the vertical vanishing point. Secondly, detect and track multiple moving objects. Then, the head feature points and the feet feature points are extracted in each frame of video sequences. The height measurements of multiple objects are obtained according to the projective geometric constraint. Finally, the multi-frame measurements are fused by using RANSAC algorithm. The experimental results demonstrate that our method is effective and robust to the occlusion.This is a preliminary study and further work is required to do.

References

[1] J. Cai, R. Walker, "Height estimation from monocular image sequences using dynamic programming with explicit occlusions," *IET Comput. Vis.*, vol. 4, no.4, pp. 149–161, 2010. Article (CrossRef Link)

[2] A. Criminisi, "Accurate visual metrology from single and multiple uncalibrated images," *Distinguished dissertation*. New York: Springer-Verlag, Sep. 2001.

[3] W. Hu, T. Tan, L. Wang, and S. Maybank, "A survey on visual surveillance of object motion and behaviors," *IEEE Trans. on Systems, Man, and Cybernetics—Part C: Applcations and Reviews*, vol.34, no.3, pp. 334–352,2004. Article (CrossRef Link)

[4] A. Criminisi, I. Reid, and A. Zisserman, "Single view metrology," *Int. J. Comput. Vis.,* vol.40, no.2, pp. 123–148, 2000. Article (CrossRef Link)

[5] I. Reid, A. Zisserman J., "Goal-derected video metrology," in *Proc. of 4th European Conference on Computer Vision (ECCV)*, pp.647–658, April 15–18, 1996.

[6] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, vol.22, no.11, pp. 1330 – 1334, 2000. Article (CrossRef Link)

[7] F. Guo and R. Chellappa, "Video metrology using a single camera," *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 32, no.7, pp. 1329-1335, 2010. Article (CrossRef Link)

[8] J. Renno, J. Orwell, and G. Jones, "Learning surveillance tracking models for the self-calibrated ground plane," in *Proc. of British Machine Vision Conf.*, pp. 607–616, Sep. 2002.

[9] B. Bose and E. Grimson, "Ground plane rectification by tracking moving objects," in *Proc. of Joint IEEE International Workshop on Visual Surveillance and Performence Evaluation of Tracking and Surveillance,* 2003.

[10] J. Shao, S. K. Zhou, and R. Chellappa, "Robust height estimation of moving objects from uncalibrated videos," *IEEE Trans. On Image Processing*, vol. 19, no.8, pp.2221-2232, 2010. Article (CrossRef Link)

[11] K. Kim, T.H. Chalidabhongse, D. Harwood, and L.S. Davis. "Real-Time foreground-background segmentation using codebook model," *Real-Time Imaging*, vol.11, no.5, pp.167-256, 2005.

[12] B. K. Bao, G. Liu, C. Xu, S. Yan. "Inductive Robust Principal Component Analysis," *IEEE Transactions on Image Processing*, vol.21,no. 8, 3794-3800,2012. Article (CrossRef Link)

[13] Z.H. Khan, I.Y.-H Gu. "Nonlinear dynamic model for visual object trackingon grassmann manifolds with partial occlusion handling," *IEEE Trans. on Cybernetics*, vol. 43, no.6, pp. 2005–2019, 2013. Article (CrossRef Link)

[14] S.H. Khatoonabadi, I.V. Bajic. "Video object tracking in the compressed domain using spatio-temporal markov random fields," *IEEE Trans. On Image Processing*, vol. 22, no.1, pp.300-313, 2013. Article (CrossRef Link)

[15] M. Du, X. M. Nan, L. Guan. "Monocular Human Motion Tracking by Using DE-MC Particle Filter," *IEEE Trans. On Image Processing*, vol. 22, no.10, pp.3852-3865, 2013. Article (CrossRef Link)

[16] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol.24, no.5, pp. 603–619, 2002. Article (CrossRef Link)

[17] L. F. Wang, H. P. Yan, H. Y. Wu, C. H. Pan. "Forward–backward mean-shift for visual tracking with local-background-weighted histogram," *IEEE Trans. On Intelligent Transportation Systems*, vol. 14, no.3, pp. 1480–1489, 2013. Article (CrossRef Link)

[18] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel based object tracking," *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 25, no.5, pp.564–577, 2003. Article (CrossRef Link)

[19] R. Hartley, A. Zisserman. *Multiple view geometry in computer vision*. 2nd Edition, Cambridge University Press, Cambridge, 2003.

Mingxin Jiang received the M.S. degree in communication and information system from Jilin University,Changchun,China,in 2005. She received the Ph.D. in Dalian University of Technology in 2013. Her research interests include multi-object tracking, video content analysis and visual metrology.

Hongyu Wang received his B.S. degrees from Jilin University of Technology, Changchun, China, in 1990,and M.S. degrees fromGraduate School of Chinese Academy of Sciences, Changchun, China, in 1993, both in Electronic Engineering. He received the Ph.D. in Precision Instrument and Optoelectronics Engineering from Tianjin University, Tianjin, China, in 1997. Currently, he is a Professor in the institute of Information Science and Communication Engineering, Dalian University of Technology, China. His research interests include computer vision, video coding and wireless video sensor networks.