A Flexible Auto Camera Calibration Technique using Vehicle Tracking

Tuan Hue Thi, Sijun Lu and Jian Zhang
National ICT of Australia
Neville Roach Laboratory
223 Anzac Parade, Kensington NSW 2052, Australia
{TuanHue.Thi, Sijun.Lu, Jian.Zhang}@nicta.com.au

Abstract

A statistical and computer vision approach using tracked moving vehicle shapes for auto-calibrating traffic surveillance cameras is presented. Various methods have been designed to estimate scene vanishing point and invertible transformation between real world and image coordinates. Results are validated against traditional methods in different traffic locations yield accurate results with much more flexibility and reliability. Given only the prior knowledge about the dimensions of typical vehicle class traveling at a particular traffic scene, it is sufficient for our approach to completely generate all information about the camera being used.

1. Introduction

1.1. Motivation

In every computer vision problem, especially those in traffic surveillance, knowledge about the camera is essential for the purpose of content retrieval. There have been around many different techniques used for obtaining those camera parameters, which is known as calibration process. Common classification divides those techniques into manual [10][8] and self-calibration [5][3]. The need for flexible and cost-effective self-calibration techniques has brought to the computer vision world many different interesting techniques, and are mostly conceptualized around the line between flexibility and accuracy. However, even with the most current popular methods [6] [3] for calibrating single camera in traffic surveillance, they still in one way of the other rely on certain assumptions about the scene location. In order to breakthrough these flexibility constraints, we propose a highly flexible self-calibration technique which can be used in all the traffic location and has been proven to produce as desirable results as the most popular self-calibration techniques do.

1.2. Related Works

The most popular manual calibration approach use square patterns in 3D (by either different orthogonal planes [8] or one plane rotating on three coordinate axes [10]). These are the most accurate approach since all the freedom degrees necessary for the transformation between image and world are obtained. However, the high cost of manually setting up the calibrating model makes these approaches hardly applicable in practical traffic surveillance.

Self-calibration techniques are often seen to use vanishing points available in the scene together with some image-world analysis to calculate camera intrinsic and extrinsic parameters [2]. These methods normally take different views of an architectural object to generate the sufficient vanishing points in all dimensions. These approach, however, have very limited application in traffic surveillance where the camera is fixed and only limited knowledge about the traffic scene is produced.

The technique described by Cathey and Dailey [3] appears to solve these constraints of uncalibrated camera using vanishing point obtained from the lane straightness on highway together with the stripe lane frequency and lane width. The interesting idea from their method is the use of a rough value of the panning angle to obtained the straightened model of the road, which is in fact proven to be good enough for generating the accurate estimation of the camera parameters. The main disadvantage of this technique, however, is the dependance on lane marking and existing stripe lane to do the calculation. This makes this technique inapplicable in surveillance locations where knowledge about the travel lanes is unknown or the stripe line is not present.

Bose and Grimson from their work in [1] introduced an interesting idea of using object tracking and prior knowledge about object dimensions to rectify images of the ground plane. Although this work is not a complete calibration process, it helps to bring up another way of solving image-world transformation using object detection and tracking from the image together with sufficient knowledge
Figure 1. Image captured by a traffic camera of the object’s physical characteristics.

The complete camera calibration work using moving objects in the scene was introduced in [4] by Fengjun et al. in which they detect and track walking human on different direction to derive three vanishing points of the scene, together with measurement from images of human at leg-crossing phases to derive the intrinsic and extrinsic parameters of the camera. This method, however, cannot be applied to our problem of traffic surveillance where normally vehicles only travel on one line, and the knowledge about the vehicles traveling on the road is much more vague than human, where typically there are many kinds of vehicles with different sizes running on the same road.

1.3. Our Approach

We propose a highly flexible approach for self-calibrating traffic surveillance camera using no other knowledge rather than the vehicle width and length. We first detect motion blobs in the traffic scenes using motion detection algorithm described in Wang et al. work [9]. Those blobs will then be passed through several filters to eliminate noises and smooth out the measurements, before being put into a Linear Kalman Filter for tracking purpose. The history of all these tracking records then helps us to produce one trajectory for each single blob using Linear Regression. A weighting algorithm is applied after that to pick the vanishing point out of the collection of all trajectory intersections, this process will be described in section 2.

The next section will then show how the actual calibration process is carried out. A similar calibration approach to Cathey and Dailey work [3] is then employed in the next stage where a preliminary straightening model is built upon the detected vanishing point. Statistics about all moving blobs corresponding to vehicles in the straightening model then facilitate us to select for each particular traffic scene the most dominant vehicle class, which we use together with the prior knowledge about the real vehicle dimensions to draw the relation between road and image planes. Eventually, the camera calibration is completed by combining all these results in the image-world transformation calculation.

Section 4 shows the successful validation of our proposed technique against the method described in [3] using different traffic surveillance video sequences. The last part of this paper will be used for conclusion, acknowledgement and references of our work.

2. Vanishing Point Estimation

2.1. Moving Blob Detection

In order to obtain valuable information about vehicles in the scene, we make use of the foreground and background segmentation algorithm described in Wang et al. paper [9]. This method will first adapt an simple motion detection to first detect the moving blobs in traffic image (figure 1), the background is also updated after every frame and after a certain amount of running time, the complete background image will be revealed as in figure 2. This approach is also very helpful by providing a way to remove shadows from those moving blobs using Markov Random Field (MRF), leaving only the meaningful pixels of vehicle images, figure 3 demonstrates how moving blobs (white pixels) are distinguished from shadows (grey pixels).

Figure 2. Generated scene background ROI

2.2. Blob Tracking with Kalman Filter

Each detected moving blob $b_i$ obtained from the foreground is passed through a filter consists of two preliminary shape constraints, namely area ($width \times height$) and ratio ($width / height$). Those shape value thresholds are
empirically chosen at this stage and will be corrected in the later phases based on statistics of the larger population of vehicles running in each particular traffic scene. A simple Linear Kalman Filter is implemented here for to keep tracks of the vehicle positions over time in both directions. Since we are only interested in the distance each vehicle has traveled rather than its velocity or acceleration, least squared distance from subsequent vehicle positions is used as the loss function to determine which vehicle belongs to which travel path. We also apply a Finite State Machine (figure 4) in this tracking model which consists of 4 separate states indicating the status of each vehicle:

\[
\begin{align*}
00: \text{initialization} \\
01: \text{noise blob} \\
10: \text{confirmed vehicle in scope} \\
11: \text{confirmed vehicle out of scope}
\end{align*}
\] (1)

where any new detected blob after ratio and area filtering will be initialized in state 00, and using the tracking results obtained from Kalman Filter, if its recorded miss_counts exceeds a threshold MAX_MISS, it will be passed to state 01, which indicates this blob as noise rather than real vehicle, whereas if its track_counts passes another threshold MIN_TRACK, that blob will be considered as a real vehicle detected in the current image, and will be sent to state 10. The last state 11 is used to capture the moment when the real vehicle goes out of the scope of the image, which is a appropriate time to finalize its travel trajectory through the image, and will be discussed in the following section 2.3.

2.3. Vanishing Point Detection Algorithm

When a vehicle finishes its route (state 11 described in the previous section), linear regression is run on all its recorded positions to generate the trajectory this vehicle has traveled through the image. The objective is to find a line \( y = ax + b \) going through all the points \( P_i(x_i, y_i) \in \tau \), the vehicle trajectory represented by a vector of all recorded positions, with

\[
a = \frac{\sum_{i=1}^{\tau} (y_i - \mu_y)(x_i - \mu_x)}{\sum_{i=1}^{\tau} (x_i - \mu_x)^2}, \quad b = \mu_y - a \times \mu_x
\]

where \( \mu_x, \mu_y \) are the means in two dimensions of all \( P_i(x_i, y_i) \in \tau \) respectively. Figure 5 shows the vehicle trajectory in green obtained from the collection of all vehicle position points in white. A legitimate vehicle trajectory route is defined by its number of recorded vehicle points and tangent value. Number of vehicle points is calculated based on the statistics of vehicles running on the road. Vehicles always contribute most to all the detected number of moving blobs on the roads (with respect to other objects like pedestrians and other foreground noises). This raises a fact that real vehicles will have more dominant tracking status compared to all the moving blobs detected. The results on figures and do actually demonstrate this, the two histograms on trajectory’s length \( \tau \) and tangent \( a \) of vehicles obtained from a long running sequence appear like two Gaussian distributions with small variances and long tails. The tails on two sides are considered as noises in our vehicle trajectory model, they might be caused by vehicles or passengers traveling across the road on directions different from the one.
we are interested in. Although they only contribute to the minor portion of the vehicle route, it is helpful to remove those noises from consideration, and it is done by rejecting the first and last quartile of both the length and tangent distribution.

After the constraints of vehicle trajectories have been defined in both terms of length and tangent, all the vehicle trajectories on different lanes will in theory meet at their vanishing point, and in practice, a large number of trajectories will converge at the same vanishing point. Using this observation, we then define a term called “trajectory intersection area”, which is basically a circular region with a defined radius, and a center generated from the intersection of two vehicle trajectories. This circular region will have an attribute of intersection count, which indicates the number of other intersections in this region. This intersection count gets incremented every time a new trajectory intersection is generated within the radius distance from the center of the considered region. When the intersection counts reach certain large values, the region with maximum intersection counts will be chosen as the vanishing points. Below is the complete snapshot of the vanishing point (thick pink circle) detection chosen from the region with maximum number of intersection count among all the trajectory intersection areas (thin brown circles) formed by all vehicle trajectories (green lines).

3. Camera Calibration

The road-image transformation in our system is presented in figure, where road and image planes are the Cartesian systems originating at $O(x, y)$ and $C_i(r_o, c_i)$ respectively. With this transformation, a particular point on the road plane $P_r(r_r, c_r)$ will find itself exactly one corresponding point on the image plane $P_i(r_i, c_i)$. The pinhole camera model with coordination $C_c(\pi, \gamma, \varpi)$ pointing down the road plane from a distance of $h$ is considered to generate this transformation. This camera-view geometry is well explained in most of computer vision literatures [6][10][8]. Basically, for a point in real world $X = (x, y, z)$, there will be a corresponding point $\tilde{X} = (\bar{x}, \bar{y}, \bar{z})$ in the camera coordinate, and the relationship between two
points is represented using an invertible affine transformation with a rotation matrix $A$ and a translation matrix $B$

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}$$

and $B = \begin{pmatrix} 0 \\ 0 \\ -h \end{pmatrix}$ where $A$ can be represented by the rotation angles $\psi$, $\theta$ (tilt), and $\phi$ (pan) about three axes $\pi$, $\gamma$ and $\sigma$ respectively of the camera coordination, and $h$ is the distance from the camera center to the road plane, all points on the road plane $O(x, y)$ will have the same height $z = 0$. Generally, traffic surveillance cameras are installed with zero roll angle, or $\psi = 0$, so $A$ normally depends only on $\theta$ and $\phi$ in $A(\theta, \phi) = \begin{pmatrix} \cos \theta \cos \phi & -\sin \theta & \cos \theta \sin \phi \\ \sin \theta \cos \phi & \cos \theta & -\sin \theta \sin \phi \\ -\sin \phi & 0 & \cos \phi \end{pmatrix}$

All points from real world going through the camera $C_r(\pi, \gamma, \sigma)$ will be shown on the image plane $C_i(u, v)$ with a transformation $u = f \frac{x}{w}$ and $v = \frac{y}{w}$ where $f$ is the distance from the camera center $C_r$ to the image plane, and is normally called focal length in pin-hole camera model. All these definitions yield us the relationship between a point $P_i(x, y, 0)$ on the road plane with its image point $P_i(u, v)$ ([3]):

$$\begin{align*}
  u &= f \frac{a_{11} x + a_{12} y + a_{13} h}{a_{11} x + a_{12} y + a_{13} h} \\
  v &= f \frac{a_{21} x + a_{22} y + a_{23} h}{a_{21} x + a_{22} y + a_{23} h}
\end{align*}$$

which is proven from their papers [3] that, these preliminary straightening values $A'$ and $f'$ in accordance with two scale factors $\beta_{h}$ on vertical and $\beta_{o}$ on horizontal lines will be sufficient to calculate the real camera parameters $f$, $\phi$, $\theta$, and $h$ via the relationships:

$$f = \sqrt{\frac{1}{2} (d^2 - 2(u_{\infty}^2 + v_{\infty}^2) + d\sqrt{d^2 - 4u_{\infty}^2})}$$

$$\phi = \arctan \frac{v_{\infty}}{f}$$

$$\theta = \arcsin \frac{-u_{\infty}}{\sqrt{f^2 + u_{\infty}^2 + v_{\infty}^2}}$$

$$h = \frac{h'}{\beta_{h} a_{11} a_{22}}$$

In terms of image calculation, the actual values we are interested in are the coordinates of the image pixels $P_i(r_i, c_i)$ where $r_i = x + v_i$ and $c_i = c_o + u_i$ (for image plane with image center at $C_i(r_o, c_o)$), and $P_i(r_r, c_r)$ where $r_r = X_{\max} - X_{\min} - x_i$ and $c_r = y_i - Y_{\min}$ (for road plane with selected rectangular region formed by $X_{\min}, X_{\max}, Y_{\min}$ and $Y_{\max}$) (figure 9).

As we can see, an ordinary camera calibration technique requires 8 variables to be solved, and normally done using checkboxes for the three rotation angles. In our system, by using the vanishing point $P_\infty(u_{\infty}, v_{\infty})$, we have reduced the number of variables with translation matrix $B^T = (0, 0, -h)$ and rotation matrix $A(f, u_{\infty}, v_{\infty})$ (obtained from letting $x \rightarrow \infty$ [3]):

$$A = \begin{pmatrix} \frac{f}{\sqrt{w^2 + u_{\infty}^2}} & \frac{u_{\infty}}{\sqrt{w^2 + u_{\infty}^2}} & \frac{v_{\infty}}{\sqrt{w^2 + u_{\infty}^2}} \\
\frac{w}{\sqrt{w^2 + u_{\infty}^2}} & \frac{w}{\sqrt{w^2 + u_{\infty}^2}} & \frac{w}{\sqrt{w^2 + u_{\infty}^2}} \\
0 & \frac{w}{\sqrt{w^2 + u_{\infty}^2}} & \frac{w}{\sqrt{w^2 + u_{\infty}^2}} \end{pmatrix}$$

where $w = \sqrt{f^2 + v_{\infty}^2}$. In addition, with the relationship $v_{\infty} = f \tan \phi$, we can either use $f$ or $\phi$ for $A$. Our system also makes use of the technique described in [3], where they make a hypothesis straightening based on a guess value of $\phi'$, an arbitrary value of $h'$, and an estimation of $P_\infty(u_{\infty}, v_{\infty})$ to calculate $A'(a_{11} \cdots a_{33})$ and $f'$. It is proven from their papers [3] that, these preliminary straightening values $A'$ and $f'$ in accordance with two scale factors $\beta_{h}$ on vertical and $\beta_{o}$ on horizontal lines will be sufficient to calculate the real camera parameters $f$, $\phi$, $\theta$, and $h$ via the relationships:

Using these equations, we then go on approach camera calibration in a similar manner, except that the values $P_\infty(u_{\infty}, v_{\infty})$ (described in the previous section) and $\beta_{h}$ and $\beta_{o}$ are obtained in a statistical method which is proven to be more flexible and transparent from the physical characteristics of the traffic scenes.

3.1. Preliminary straightening for vehicle image shape statistics

The preliminary straightening model are built based on a feasible value of $\phi = -10$ (which is proven not to be
necessarily accurate [3]), a camera height value \( h \) (preferably dependent on how big we want the straightening width region to be), and finally the value of the vanishing point we obtained from the previous part. The straightened image obtained is shown in figure . Using this straightening model, our objective is to obtain the two ratios on both vertical and horizontal directions \( \beta_h \) and \( \beta_v \) respectively. This is done using a statistical analysis on the data of the moving vehicles \( \nu_i(\text{area}_i, \text{width}_i, \text{length}_i) \) detected in the image scene and transformed into straightening model \( \nu'_i(\text{area}'_i, \text{width}'_i, \text{length}'_i) \). These vehicle image parameters together with their actual values \( \nu(\text{AREA, WIDTH, LENGTH}) \) will help to derive the ratios \( \beta_h = \frac{\text{length}'_i}{\text{LENGTH}} \) and \( \beta_v = \frac{\text{width}'_i}{\text{WIDTH}} \) which are eventually used to update our final camera calibration according to (9). The decision on which kind of vehicles to pick for their parameters is made during the analysis of traffic surveillance data provided by the Road and Transport Authority (RTA), which indicates the dominant population of the medium vehicle class on the road considering a sufficient long period of surveillance. This helps us to make the decision of selecting this class of vehicle for analysis. The corresponding values for these vehicles in straightening model are easily chosen using the analysis of vehicle shape (area and ratio) distributions, which are shown in figures 11 and 12. These two distribution graphs in fact confirm the fact about the dominance of mid-size car class, hence help us to narrow down the range of input vehicle data into only those peak areas (middle quartiles of the Gaussian distributions on both area and ratio). The method to pick width and length values out of straightened vehicle image are done using the polygon model of vehicle shape 13.

With the method described in [7], the vehicle model consists of 6 principal measures in 3 dimensions, in which two of them are the width and length. All width and length of the interesting vehicles (which meet the requirement of area and ratio) are then put into 2 vectors. The medians of these two sorted arrays are the final values \( \text{length}'_i \) and \( \text{length}'_i \) that we used to estimate \( \beta_h \) and \( \beta_v \), hence used to come up with the final camera calibration according to formula (9).

\[ \beta_h = \frac{\text{length}'_i}{\text{LENGTH}}, \quad \beta_v = \frac{\text{width}'_i}{\text{WIDTH}} \]

\[ \text{Figure 10. Straightening Model} \]

\[ \text{Figure 11. Vehicle Shape Area Distribution} \]

\[ \text{Figure 12. Vehicle Shape Ratio Distribution} \]

\[ \text{Figure 13. Vehicle Models (a) in 3D (b) straightened and (c) fitted convex hull image} \]

\section{4. Result Validation}

This method is validated against the approach described in [3] in different traffic surveillance sequences obtained from the New South Wales Road and Traffic Authority (RTA), the results are summarized in the following table with results from our proposed method are shown in the left column, while the right column is left for the method described in [3], and the vanishing point is almost the same in all cases and are displayed once only. These results show that our proposed method yield the same results as the existing ones. In addition, this method is much more flexible and independent on the specific traffic location, and regardless of the lane existing, the values obtained will be the same.
Table 1. Result Validation

<table>
<thead>
<tr>
<th>Seq</th>
<th>M4 Hwy C1</th>
<th>M4 Hwy C2</th>
<th>M4 Hwy C3</th>
<th>Anzac Pde</th>
<th>Elizabeth St</th>
<th>City Rd</th>
<th>Liverpool St</th>
<th>Bourke St</th>
<th>Oxford St</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P_{\infty}$</td>
<td>(391; 28)</td>
<td>(355; 1)</td>
<td>(353; 26)</td>
<td>(379; 24)</td>
<td>(368; 48)</td>
<td>(335; 12)</td>
<td>(337; 116)</td>
<td>(316; 64)</td>
<td></td>
</tr>
<tr>
<td>$h$</td>
<td>11.5</td>
<td>11.4</td>
<td>11.2</td>
<td>11.7</td>
<td>11.5</td>
<td>7.5</td>
<td>11.6</td>
<td>11.1</td>
<td>7.6</td>
</tr>
<tr>
<td>$f$</td>
<td>356</td>
<td>359</td>
<td>329</td>
<td>481</td>
<td>1607</td>
<td>1592</td>
<td>431</td>
<td>444</td>
<td>680</td>
</tr>
<tr>
<td>$</td>
<td>\theta</td>
<td>$</td>
<td>18</td>
<td>17.9</td>
<td>18</td>
<td>19.9</td>
<td>18</td>
<td>6.6</td>
<td>9.6</td>
</tr>
<tr>
<td>$</td>
<td>\phi</td>
<td>$</td>
<td>18</td>
<td>17.9</td>
<td>23.5</td>
<td>24.9</td>
<td>5.1</td>
<td>5.2</td>
<td>6.83</td>
</tr>
</tbody>
</table>

5. Conclusion

In our paper, a statistical and computer vision approach in camera calibration specifically for traffic surveillance has been proposed and validated. Given only the prior knowledge about the dimensions of typical vehicle class traveling at a particular traffic scene, it is sufficient for our approach to completely generate all information about the camera being used. This fully auto-operated approach can be used flexibly on different traffic locations to help reduce the burden of manual calibration.

Acknowledgement

We would like to thank Yang Wang and Getian Ye for their help in the Motion Detection part, and Kostia Robert in his camera calibration data from the Traffic Surveillance System. This work is part of a collaboration project between National ICT of Australia (NICTA) and the Road and Traffic Authority of New South Wales (RTA). NICTA is funded by the Australian Government’s Backing Australia’s Ability initiative, in part through the Australian Research Council.

References


