Scale-Accurate 3D Vehicle Point Cloud Extraction from Single-Camera Traffic Video

Jędrzej Kowalczuk, Eric T. Psota, and Lance C. Pérez
Department of Electrical Engineering, University of Nebraska, Lincoln, NE, U.S.A.

Abstract—Reliable data extraction is essential to achieving a high-level understanding of the processes ongoing in the traffic environment. The ability to extract 3D structural properties of vehicles enables advanced traffic analysis such as vehicle classification and the detection of traffic rule violations, accidents, and near-collisions. A novel method is presented for extracting 3D structural properties of moving vehicles using a single camera. This method operates by tracking features of the vehicle as it travels through the camera’s field of view. Two-frame structure from motion is then used to extract a 3D point cloud representing the structure of the vehicle. In addition, a robust pose estimation algorithm is given for relating the geometry of the street surface to the position of the camera with minimal user interaction. It is shown that the resulting 3D point cloud can be used to accurately approximate the dimensions of vehicles to within a half foot of their true dimensions.

Keywords: Traffic camera, 3D reconstruction, structure from motion, pose estimation, bundle adjustment.

1. Introduction

Image processing is often used to detect low-level traffic events. These low-level events include vehicle velocity and traffic flow rate estimation by counting cars that enter and exit the camera’s field of view. Methods for low-level traffic event detection typically rely on some form of background subtraction and frame-to-frame tracking of binary connected components. While basic traffic data can be extracted from a 2D projection of the video frames to the street plane, extracting information regarding the 3D structure of vehicles is a much more challenging problem to computer vision researchers.

Within the intelligent transport systems (ITS) research field, a large volume of research has been focused on automated event detection and data extraction from traffic video [1]. As the cost of video-capture hardware continues to decrease, there has been an increased number of traffic video monitoring systems deployed for both red light enforcement and traffic flow analysis. These systems, combined with the decreasing cost and increasing capacity of digital storage, have made it possible to obtain an abundance of traffic video. However, analysis and understanding of traffic behaviors create a need for reliable data extraction from traffic video, which preferably requires minimal user interaction. Examples of data that are commonly extracted from traffic video include traffic flow rate, vehicle velocities, traffic violations, and the presence of roadway debris.

To address the need for automated high-level data extraction towards sophisticated traffic video analysis, a novel method is presented for recovering scale-accurate structural properties of moving vehicles in the form of 3D point clouds. This method integrates the constraints governing vehicular motion into the process of structure recovery and motion estimation from single-camera traffic video. It is expected that the proposed method will lead to more robust traffic video analysis, a better understanding of traffic behavior, and improved driver safety.

2. Background

A variety of methods exist for extracting data from traffic video. These methods are thoroughly reviewed in the excellent survey paper by Buch et al. [1]. Most often, traffic video is processed using a common series of stages, arguably the most important of which is background subtraction used to isolate moving vehicles from a relatively stable environment. Several factors make background subtraction a challenging task including the slowly varying angle of sunlight, cloud movement, shadows created by vehicles, non-rigid objects (e.g., foliage), image sensor noise, and photometric variations [1], [2], [3]. Methods that have been shown to be effective for estimating background include the approximation of slowly varying pixel-level Gaussian intensity distributions [4], and the detection and removal of shadows [5].

Given traffic video captured from a small angle of incidence with respect to the street plane, detecting the presence of vehicles and estimating their velocities is a relatively straightforward task once foreground and background have been separated. In contrast, urban environments are more difficult to analyze since their associated traffic video is typically obtained from a larger angle of incidence [1]. In addition, tasks such as automatic classification of vehicles, detection of illegal turns, lane changes, and accidents are difficult to achieve using vision-based traffic analysis. Many approaches to performing vehicle classification use precomputed 3D models of the vehicle types they are trying to identify [6]. The most common approach is to orient a projection of the 3D model in the scene in such a way that it accurately matches the silhouette provided by background subtraction. Along this line, simple wireframe approximations of automobile types have recently been proposed to simultaneously identify vehicle type and 3D pose from single-camera video [7].

One significant limitation of the model-based vehicle classification schemes is that they rely on precomputed 3D models and exhaustive orientation search in order to fit the model to
the vehicle extracted from the video sequence. The method presented in [8] addresses this limitation by incorporating unsupervised learning in order to adapt basic vehicle templates to specific vehicle models from video sequences. It has been suggested that stereo video can be used to improve vehicle classification due to the fact that stereo video processing enables 3D structure recovery [1]. Results given in [9] demonstrate that reliable vehicle classification can be achieved using a stereo vision-based feature extraction method for computing 3D properties of vehicles. The primary advantage of stereo video lies in the fact that 3D structure can be computed at a single instant in time and that structural data can be fit to known models using common 3D-to-3D registration techniques, such as the orthogonal Procrustes algorithm [10] and iterative closest point [11].

3. Method

The proposed method assumes that the camera has been calibrated, i.e., internal camera parameters (focal length and principal point offset) are known prior to processing of video feedback. It is also assumed that the user has defined a plane of known geometry within the view of the camera associated with the surface of the street. These two inputs are necessary to recover the relative position and orientation (pose) of the camera with respect to the street. After recovering the pose of the camera, a sequence of operations including blob detection and tracking followed by feature detection and matching, and two-frame structure and motion estimation with bundle adjustment are performed to extract a scale-accurate 3D point cloud representing the sparse structure of each vehicle moving through the field of view. The complete traffic video processing pipeline is shown in Figure 1. In the following, the processing pipeline and its stages are described in detail.

3.1 Camera Pose Estimation

Knowledge of a precise mapping between the camera coordinate system and the world coordinate system is essential for reliable recovery of 3D structure of vehicles within the view of the camera. This mapping is used to resolve the scale ambiguity inherent in single camera structure from motion, under the assumption that vehicle motion occurs along the street plane and the camera position and orientation are fixed.

To obtain the mapping from world to camera coordinates, the user must first select a set of four coplanar street points with known world coordinates, as illustrated in Figure 2(a).

Using the four street plane coordinates and their corresponding image points, it is possible to accurately approximate the pose of the camera relative to the street plane, where the orientation of the camera is represented by a $3 \times 3$ rotation matrix $R$ and its position is represented by a 3D vector $C$. The pose is derived using an extension of the perspective 4-point algorithm introduced in [12].

Given the internal camera calibration matrix $K$, the homogeneous image points $x = [x, y, 1]^T$, and their corresponding locations on the street plane $X = [X, Y, 0, 1]^T$ (under the assumption that the street plane lies on the $XY$-plane with $Z = 0$), the mapping from world coordinates to image points is given by

$$x \equiv K[R \mid -RC]X = K[r_1 \ r_2 \ | -RC]X$$

where $R$ is the rotation matrix with columns $r_1$, $r_2$, $r_3$ representing the orientation of the camera, vector $C$ holds the 3D coordinate of the camera center, and $X = [X, Y, 1]^T$ is the world coordinate vector with $Z$ removed. Left-multiplying both sides of Equation (1) by $K^{-1}$ results in

$$K^{-1}x = [r_1 \ r_2 \ | -RC]X$$

where the $3 \times 3$ transformation matrix $T = [r_1 \ r_2 \ | -RC]$ can be solved for using the DLT algorithm (Algorithm 4.1 in [13]). Under ideal conditions, the first two columns $r_1$ and $r_2$ should both have unit norm and be orthogonal to each other. However, since the method given in [12] does not guarantee that $T$ will maintain these properties under realistic conditions, a valid orthogonal rotation matrix $R$ can be constructed using

$$T \leftarrow \frac{T}{\|r_1\|} \quad (3)$$

$$r_3 = \frac{r_1 \times r_2}{\|r_1 \times r_2\|} \quad (4)$$

$$r_2 = r_3 \times r_1 \quad (5)$$

$$R = [r_1 \ r_2 \ r_3] \quad (6)$$

and the associated camera center can then be computed as $C = -R^T r_3$. Due to point selection error and image coordinate quantization, the pose parameters obtained using this method require further refinement.

The initial pose estimate is refined using the Levenberg-Marquardt algorithm to find $R$ and $C$ that minimize the reprojection error from $X$ to $x$. The algorithm requires the Jacobian matrix relating the output $x$ to the input $X$ with respect to the
unknown pose parameters. Rather than evaluating the Jacobian directly, the projection process is decomposed into a series of steps such that the Jacobian can be computed by back-propagating and chaining the intermediate derivatives [14].

To handle linearization of the 3D rotation, an incremental rotation matrix

$$\Delta R(v) = \begin{bmatrix} 1 & -v_z & v_y \\ v_z & 1 & -v_x \\ -v_y & v_x & 1 \end{bmatrix}$$  (7)

is defined where $v = (v_x, v_y, v_z)$ is a minimal rotation vector, such that the direction of $v$ specifies the axis and its magnitude specifies the angle of rotation. Note that the definition of $\Delta R(v)$ in Equation (7) is only valid for small rotations. In this implementation $\Delta R(v)$ is allowable since an incremental update to the initially estimated rotation matrix given in Equation (6) is sought. Using this definition of incremental rotation, projection is decomposed into

$$X_C = X - C$$  (8)

$$X_R = RX_C$$  (9)

$$X_v = \Delta R(v)X_R$$  (10)

$$\hat{X} = [X_v/Z_v, Y_v/Z_v, 1]^T,$$  (11)

where $\hat{X} = K^{-1}x$ since the internal camera parameters are fixed and do not affect minimization. The stages of projection along with the associated partial derivatives are shown in Figure 3. Combining the partial derivatives associated with the stages shown in Figure 3 leads to the formulation of the Jacobian

$$\frac{\partial \hat{X}}{\partial (C, v)} = \begin{bmatrix} \partial \hat{X} \\ \partial \hat{C} & \partial \hat{X} \end{bmatrix}$$

$$= \begin{bmatrix} \partial \hat{X} & \partial X_x & \partial X_R & \partial X_C \\ \partial X_v & \partial X_R & \partial X_C & \partial C & \partial \hat{X} \partial X_v & \partial \hat{C} \partial \hat{X} \partial X_v & \partial \hat{C} \partial \hat{X} \partial X_v \end{bmatrix},$$  (12)

which is used for updating the pose parameters in the Levenberg-Marquardt algorithm. Note that, to allow for large rotation adjustments, the camera rotation matrix is recalculated using $R \leftarrow \Delta R(v)R$ after each iteration of minimization.

Orthogonality of the rotation matrix is then forced and $v$ is set to zero prior to the next iteration.

In addition to determining the camera pose, a planar homography transformation $H$, such that $X = Hx$, is derived from the four user-defined image points. This homography allows for the recovery of 3D world coordinates of image points located on the street surface, as illustrated in Figures 2(b) and 2(c). Later on, when generating vehicle point clouds, this transformation is used to remove scale ambiguity associated with single-camera structure from motion.

3.2 Blob and Feature Detection and Tracking

The vehicles are identified in the video using adaptive background subtraction, where pixel intensity distributions (approximated as Gaussians) are estimated independently for every pixel location. This method is known to mitigate the effects of camera shake and other persistent intensity variations within the scene [4]. Foreground pixels are defined as those with distance from the mean value that is greater than some multiple of the variance. The output of adaptive background subtraction is a binary image where foreground objects are assigned a value of 1 and the background is set to 0. A sample frame of the video sequence and the corresponding mean image are given in Figures 4(a) and 4(b).

The binary image obtained through background subtraction is subjected to median filtering and morphological closing (dilation followed by erosion), which are necessary to eliminate noise artifacts resulting from thresholding and to fill gaps in the image objects. Once the filtering is complete, standard 4-connected neighborhoods of the binary image are examined in order to identify, label, and characterize blobs that correspond to vehicles. For each of the blobs, a set of key properties is recovered, which include area, bounding box, centroid, and blob shape in the form of a binary mask. Smaller blobs are excluded from further analysis, since they are due to noise or do not represent vehicles. An example of blob detection is shown in Figure 4(c).

To allow for vehicle tracking, the system maintains a list of vehicles present in the current view and updates this information as new frames become available. Correspondences of blob centroids are recovered between successive frames through
Fig. 3: Decomposition of the projection process for camera pose refinement using the Levenberg-Marquardt algorithm. The reprojection error is minimized with respect to the camera center $C$ and rotation adjustment $\Delta R(v)$.

\[
\begin{align*}
\frac{\partial X_C}{\partial C} &= -I \\
\frac{\partial X_R}{\partial X_C} &= R \\
\frac{\partial X_v}{\partial X_R} &= I, \quad \frac{\partial X_v}{\partial v} = \begin{bmatrix} 0 & Z_R & Z_R & 0 \\ -Z_R & 0 & Y_R & -Y_R \\ 0 & -Y_R & -X_R & 0 \\ 1 & Z_v & 0 & \frac{1}{Z_v} & -\frac{X_v}{Z_v^2} & -\frac{Y_v}{Z_v^2} \end{bmatrix}
\end{align*}
\]

3.3 Vehicle Structure and Motion Estimation

The movement of the vehicle through the scene can be interpreted in two ways: displacement of the vehicle with a fixed camera, or displacement of the camera with a fixed vehicle. The proposed method aims to recover the displacement of the camera under the assumption that the vehicle is fixed. The planar nature of vehicle motion constrains the relative movement of the camera between views, in that the camera center is translated along the $XY$-plane and rotated around the $Z$-axis of the world coordinate system. Therefore, relative camera motion in the world coordinate system is parameterized geometrically constrained nearest neighbor matching, leading to the recovery of frame-to-frame motion vectors associated with particular vehicles. In case a match cannot be established for a given blob centroid, the corresponding vehicle is either entering or exiting the view, which is handled appropriately. An illustration of vehicle tracking is given in Figure 4(d).

In addition to vehicle tracking, features are detected and matched as the vehicle moves through the scene. This feature detection and matching is necessary for computing the 3D point cloud of the vehicle. The SURF feature detection method is used to detect features and assign feature descriptors since it is invariant to changes in intensity, scale, and rotation, and handles small affine distortions [15]. Features are matched using a nearest neighbor distance ratio threshold of 0.6 and consistency is enforced to identify pairs of uniquely matched features. The results of feature matching between two views of the same vehicle is shown in Figure 5.

Fig. 4: The vehicles are identified in the image using a combination of adaptive background subtraction and blob detection. The vehicles are then tracked using the assumption of small frame-to-frame movement.

Fig. 5: A set of consistent matches obtained using SURF with a nearest neighbor distance ratio (NNDR) threshold of 0.6
Fig. 6: Decomposition of the projection process for two-frame bundle adjustment using the Levenberg-Marquardt algorithm. The camera translation $C'$ and camera rotation $R(\theta)$ represent the relative movement of the camera assuming the vehicle position is fixed. The parameters $C'$ and $R(\theta)$ are set to zero and identity, respectively, in one view and optimized for the other view with respect to reprojection error. Note that the 3D points $X$ are also adjusted along with rotation and translation between views.

by the displacement vector $C' = [c'_x, c'_y, 0]^T$ and the rotation

$$ R(\theta) = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad (13) $$

where $\theta$ defines the angle of rotation around the Z-axis.

With a set of matched vehicle feature points between two views, bundle adjustment is used to estimate the relative motion of the camera and the 3D points corresponding to features. Analogously to the pose estimation method described in Section 3.1, motion parameters are acquired through non-linear optimization using the Levenberg-Marquardt algorithm to minimize the reprojection error of 3D structure coordinates to 2D image points. Whereas pose estimation minimizes reprojection error in a single view, bundle adjustment aggregates the reprojection error in a pair of views while simultaneously adjusting the 3D structure coordinates. The decomposition of the projection process is given in Figure 6.

In order to eliminate outliers from the set of feature matches, RANSAC [16] is applied. At each iteration of RANSAC, a sampling of eight features is randomly chosen to estimate $C'$ and $R(\theta)$. Using the estimated camera motion parameters, 3D vehicle coordinates are triangulated (Equation 7.4 in [14]) from the entire set of feature matches. The 3D coordinates are then projected into both views and outliers are identified as those that exceed an empirically chosen reprojection error. Next, the chosen set of inliers is processed by bundle adjustment to derive the final structure and motion parameters.

4. Results

To evaluate the accuracy of the proposed method, four sample vehicles with different body types (sedan, van, SUV, and a pickup truck) were processed to extract 3D point clouds. The four vehicles with their features in both views are

Fig. 7: Vehicles used to verify the accuracy of the proposed method along with the features that were identified in each of the frames.
shown in Figure 7. The accuracy of the method is evaluated by comparing vehicle dimensions manually extracted from the 3D points clouds with the actual dimensions. Figure 8 demonstrates the extraction of vehicle dimensions from the 3D point cloud corresponding to the van.

![Fig. 8: Point cloud of a van obtained using the proposed method along with measurement points used to obtain the length, width, and height of the van. Note that the height measurement requires only one sample since the vehicle lies on the plane at $Z = 0$.](image)

Table 1: Dimensions of various vehicles recovered using the proposed method along with their actual dimensions and the corresponding average absolute measurement errors.

<table>
<thead>
<tr>
<th>Vehicle</th>
<th>Actual dimensions (ft)</th>
<th>Extracted dimensions (ft)</th>
<th>Avg. error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sedan</td>
<td>15.42 5.80 4.35</td>
<td>15.33 6.15 3.88</td>
<td>0.30</td>
</tr>
<tr>
<td>Van</td>
<td>15.48 5.71 6.42</td>
<td>15.00 5.86 5.98</td>
<td>0.36</td>
</tr>
<tr>
<td>SUV</td>
<td>15.10 5.65 5.57</td>
<td>15.07 5.19 5.33</td>
<td>0.24</td>
</tr>
<tr>
<td>Truck</td>
<td>19.61 6.58 6.17</td>
<td>20.08 6.87 5.94</td>
<td>0.33</td>
</tr>
</tbody>
</table>

A comparison between actual vehicle dimensions and those extracted using the proposed method is given in Table 1. The results show that the method is capable of recovering the dimensions of vehicles with average absolute measurement error of less than 0.36 feet. In addition, the maximum error in a single dimension is 0.48 feet.

The availability of sparse, accurate 3D point clouds creates the opportunity to recover precise vehicle trajectories, which also enables high-level detection of events such as lane changes, traffic violations, accidents, and near-collisions. The points can also be used to accelerate model fitting for automated vehicle classification. Altogether, the resulting enhancements to traffic data collection are expected to help identify abnormal and unsafe driver behaviors, which is important to improving intersection safety.

5. Conclusion

A method has been presented for obtaining scale-accurate 3D point cloud representations of vehicles from a single traffic camera. With only two inputs provided by the user - internal camera calibration and four user-defined street surface points with known dimensions - the six-dimensional pose of the camera relative to the street plane can be reliably estimated prior to 3D point cloud extraction. As vehicles pass through the camera’s field of view, blob detection and tracking is used to separate the vehicles from the background, and two sufficiently different perspectives of the vehicle are obtained. Correspondences between the two views are acquired using SURF feature detection and matching. The correspondences are then processed using two-frame structure from motion with bundle adjustment to estimate the 3D point cloud coordinates along with the motion of the vehicle under the assumption of translation on the $XY$-plane and rotation around the $Z$-axis in the world coordinate system. Outliers in the 3D point cloud are further eliminated using RANSAC to minimize the reprojection error of triangulated feature matches.

Results demonstrate that the proposed method is capable of accurately obtaining structural properties of passing vehicles. It has been demonstrated that vehicle length, width, and height can be recovered with an average error within 0.36 feet when compared to their actual values. For added robustness to varying environmental conditions, methods for efficient
detection and removal of shadows from the images after background subtraction should be integrated with the proposed method. Moreover, incorporating additional views into the structure recovery process is expected to improve the accuracy of the resulting point clouds. Alternative geometric primitives, including lines and image segments, may be identified and matched between views in order to obtain more detailed vehicle models. This method also lends itself well to dense reconstruction techniques like stereo matching, allowing for robust model-fitting for the purpose of vehicle classification.

References


