VEHICLE TRACKING FROM DISPARATE VIEWS

Lin Yang, John Johnstone, Chengcui Zhang

Computer and Information Sciences
University of Alabama at Birmingham
{galabing, jj, zhang}@cis.uab.edu

ABSTRACT

Most approaches to vehicle tracking have adopted a single calibrated camera for the task, which leads to an under-conditioned problem. We present a surveillance system for on-line vehicle tracking based on two cameras and structure from motion (SfM). Our surveillance system starts by tracking feature points. A novel matching scheme is proposed that allows a subset of feature points to be corresponded across disparate views. Based on the reconstructed subset, the full set of feature points are reconstructed in 3D and segmented into different vehicles by solving a multiple labeling problem.

Index Terms— vehicle tracking, structure from motion

1. INTRODUCTION

Automatic extraction of vehicle parameters (e.g., world position and orientation) is a fundamental problem in traffic monitoring. Most video-based approaches have adopted a single calibrated camera for this task [1, 2, 3]. However, recovering 3D information from a 2D view is inherently an under-conditioned problem. Thus, they all require some heuristic to serve as the missing constraint, which most often comes from image foreground analysis, and is also most likely to become the major bottleneck of their systems.

In [3], image foreground blobs are compared to a database of pre-generated 2D projections of vehicle models from a range of viewpoints, so as to gain a constraint on the vehicle orientation. Therefore its performance is dependent on the quality of foreground extraction and noise (shadows and reflections) removal. In [2], Lou *et al.* iteratively minimize the residual between foreground edge points and the projection of pre-defined 3D wire-frame models to estimate the pose of a vehicle in world coordinates. However, this approach has difficulty handling vehicle-to-vehicle occlusions, because the merged foreground edges lead to unexpected local minima.

A feature based approach, which forms the basis of our work, is gaining more attention due to its ability to track under partial occlusions. Its early application in vehicle tracking is explored by Beymer *et al.* [4] and later improved by Kanhere *et al.* [1] to handle the low-angle scenario. Nevertheless, in order to gain any 3D information, they still must provide the missing constraint by inferring the height of vehicles using vehicle foreground analysis, which is sensitive to vehicle shadows and occlusions.

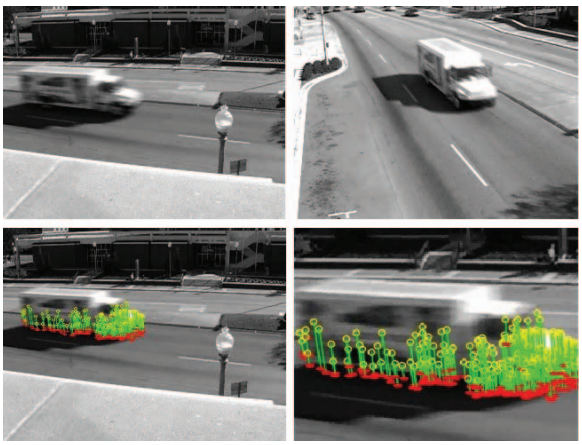

Fig. 1. Reconstructing vehicle features from two views: the top row shows two camera views of the vehicle, and the bottom row shows the reconstruction results (left) and a zoomed-in view (right) for clarity. The world coordinates are illustrated with their ground projections (the red bars).

Adding a second camera removes the bottlenecks of these methods and attacks the under-conditioning problem directly, because the additional view provides another two constraints and boosts the problem to an over-conditioned one. This idea formalizes our approach in a structure from motion (SfM) framework, where 3D reconstruction from two views is already well studied [5] (see Figure 1).

Now the major problem becomes finding point correspondences between the two camera views. The corner-based local descriptors [6] widely used in image matching do not work well here, because many of the corners are blurred away due to vehicle motion, and they cannot handle the case when the two cameras have very different viewpoints.

In this paper, we explore a new reconstruction scheme which allows the two cameras to be installed disparately, yet is still able to reconstruct a rich set of vehicle features. Our contributions are twofold:

1. We propose a matching algorithm to correspond feature points between two video streams where two

cameras can be installed with a large difference in viewpoint, focal length, and image quality.

2. We reduce the full reconstruction of vehicle features to a multiple labeling problem and show that, by adjusting the energy functions, it is possible to solve the problem of reconstruction, grouping of vehicles, and outlier removal at the same time.

2. RECONSTRUCTION OF VEHICLE FEATURES

Prior to online reconstruction of vehicle features, the two video streams are synchronized and the two cameras are calibrated using Zhang's toolkit [7]. We grab a pair of corresponding frames from both cameras and compute the fundamental matrix [5]. The fundamental matrix defines the epipolar geometry between the two views, which will serve as an important constraint during on-line reconstruction.

The on-line process starts by detecting and tracking KLT [8] feature points within each image sequence. Then a small set of feature points are corresponded between two views and reconstructed (Section 2.1), followed by a full reconstruction of all the feature points. Grouping and outlier removal are handled at the same time (Section 2.2).

2.1. Two-view matching

During the first stage, a small set of correspondences between the two views is established on their KLT features and their world coordinates triangulated [5], which serves as the basis for reconstructing the rest of the feature points.

As mentioned previously, direct matching using corner-based local descriptors can be highly unreliable. Besides the unknown image quality, the two cameras will be configured with different orientations. As pointed out by Moreels *et al.* [9], no state-of-the-art detector-descriptor combination performs well with a change of viewpoint more than 25−30°.

However, the intuition is that these algorithms should be able to find good correspondences between consecutive frames of one camera given that there is little change of viewing angle relative to the moving vehicle, and no change in image quality. Thus, our core strategy is to convert the two-view matching problem to matching between frames within a single view.

We denote a feature point in consecutive frames as $\{x_0 \rightarrow x_1\}$. The goal is to find their correspondences x_0' and x_1' in the other view. Instead of finding $\{x_0 \leftrightarrow x_0'\}$ and $\{x_1 \leftrightarrow x_1'\}$ directly, we first find a set of candidates for $\{x_0' \rightarrow x_1'\}$ using local descriptors, and verify their validity using motion constraints (see Figure 2).

With known epipolar geometry, candidates for x_0' and x_1' are restricted to their corresponding epipolar lines. A brute force method would be to compute and match local descriptors for all the points lying on the line, but this would be inefficient given that most regions in a traffic scene have too little texture to form a distinctive descriptor. In [10], Ma *et al.* discover that the most repeatable vehicle features lie on the edges. Thus, we apply the Canny edge detector on every frame and greatly suppress the candidate set to those edge points on the epipolar line.

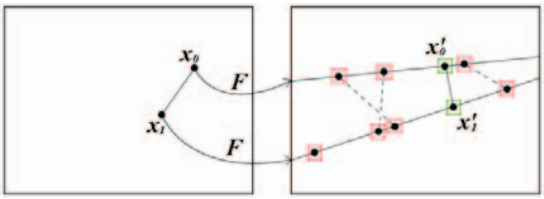

Fig. 2. Two-view matching combining epipolar and motion constraints. The epipolar geometry restricts the search space to the corresponding epipolar lines. A set of candidate matches are found between the two lines, and their validity is verified against the motion constraint that exists in traffic applications to find the correct match.

We adopt a modified SIFT [11] descriptor to represent the candidate points. While the original algorithm seeks extrema in scale space in order to make the descriptors scale-invariant, and rotates the neighborhood to the dominant orientation to achieve rotation-invariance, we observe that inter-frame motion of a vehicle can be well modeled by a pure translation (*i.e.*, there is little variance in scale or orientation), so we discard those stages and compute the descriptor directly by accumulating gradient values in the neighborhood into a $n \times n \times r$ histogram.

Since the candidate set is much smaller than in regular image matching tasks, a simplified histogram is adopted, where $n = 2$ and $r = 4$: the gradient values are accumulated in the neighboring 2×2 sub-regions. Within each sub-region, there are 4 orientation bins, each spanning $\pi / 4$: polarities are discarded as suggested by [10]. This gives us a 16-vector for each candidate point, and matching is done by finding its first and second nearest neighbors in the Euclidean space and thresholding on their ratios. For all the experiments in this paper, we use 0.6 as the maximum threshold for a good match.

The above procedure locates a set of $\{x_0' \rightarrow x_1'\}$, and thus gives a corresponding set of $\{x_0 \leftrightarrow x_0'\}$ and $\{x_1 \leftrightarrow x_1'\}$. In order to find the correct one, their validity is verified by a special motion constraint of traffic: all vehicle features should move in parallel to the ground plane. That is, the triangulated 3D points X_0 and X_1 should have little variance in their Z-component. After applying this constraint, if the remaining correspondence is unique, then it is highly probable that it is a correct match.

The proposed matching algorithm combines local descriptors, epipolar constraints and a motion constraint that exists in traffic applications. A screen shot from our experiments is shown in Figure 3.

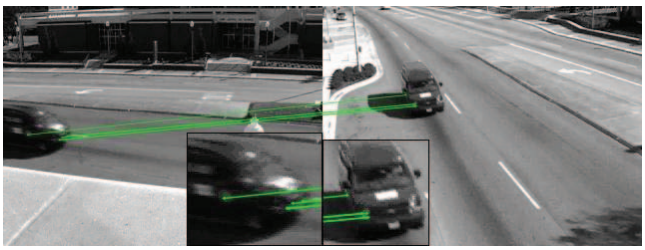

Fig. 3. Matching with changes in viewpoints. The dual constraints work even when the two cameras have very different viewpoints. In this frame, the matched points in one view are barely visible (either on the side of the vehicle, or blurred due to vehicle motion) in the other.

2.2. Reconstruction using motion constraints

The first stage is typically able to reconstruct several points for each vehicle. Based on these points, we recover the world coordinates for the rest of the feature points using the *relative height constraint* [1].

Let C denote the camera center with known height h. X_0 and X_1 denote the world coordinate of an arbitrary feature point X with height z at consecutive time stamps, and $P(X_0)$, $P(X_1)$ the corresponding ground projections along the camera ray. We can derive the following equation from similar triangles

$$\frac{\Delta X}{\Delta P(X)} = \frac{h-z}{h} \tag{1}$$

where $\Delta X = \|X_1 - X_0\|, \Delta P(X) = \|P(X_1) - P(X_0)\|$ are the Euclidean distances that the point and its ground projection have moved during this period of time. Notice that both $P(X_0)$ and $P(X_1)$ have 2 *dof* ($z = 0$) and can be solved using just one camera.

If we have another world point X' that undergoes the same motion (which is true for features on the same vehicle), then taking the ratio on Equation (1) gives us:

$$z' = h - \frac{\Delta P(X)}{\Delta P(X')}(h-z) \tag{2}$$

Therefore, if one of the points is already reconstructed (z), we can compute the height of the other (z'). Once the height is known, the world coordinate of the other point can be solved using one camera. However, an important assumption of (2) is that X and X' must share the same motion between two time stamps, otherwise (2) does not hold.

We propose an algorithm to group points into equivalent classes with the same motion. Since, for each unknown feature point, there are typically several points already reconstructed in the first stage that lie on the same vehicle, we can redefine the problem of grouping points with the same motion as a multiple labeling problem, where the set of labels ℓ contains the reconstructed points in the first stage, and all the remaining features must be assigned a label. Because some of the KLT features are actually outliers, we also include a special label ϕ for feature points that do not belong to any known motions (see Figure 4).

We adopt graph cuts [12] to optimize multiple labeling. Each unknown feature point is a vertex in the graph, and the connectivity is defined by a Delaunay triangulation in the image coordinate (Figure 4), which connects nearby features while maintaining a low vertex degree.

The energy function minimized by graph cuts consists of two terms, a "data" term measuring the compatibility between a vertex and its label, and a "smoothness" term, measuring the consistency of labeling between adjacent vertices. In our application, the cost functions are based on the geometric constraints derived above. In particular, we set two criteria to optimize the labeling:

1. A feature point should be close to its label, with similar 3D motion.

2. The labeling should be piecewise constant. That is, features close to each other should be assigned the same label.

Thus, the cost of a labeling $L \in \ell$ is defined as

$$E(L) = E_{data}(L) + E_{smooth}(L) \tag{3}$$

$$E_{data}(L) = \sum_{X \in \chi} D(X, L(X)) \tag{4}$$

$$E_{smooth}(L) = \sum_{X,Y \in \gamma} V_{XY}(L(X), L(Y)) \tag{5}$$

where χ is the set of unknown features, γ contains all the connected pairs in χ, and the cost functions D and V_{XY} are defined below.

Given a point X and its label $L(X)$, the data term D in our system consists of two costs: one measures the difference between their motion vectors, and the other measures their distance in 3D space.

$$D(X, L(X)) = \begin{cases} \sigma & L(X) = \phi \\ \|\Delta L(X) - \Delta(X)\| + \lambda \left\| \overline{L(X)} - \overline{X} \right\| & L(X) \neq \phi \end{cases} \tag{6}$$

where Δ has the same meaning as in (1), and $\overline{X} = (X_0 + X_1)/2$ is the midpoint X's motion path.

For the smoothness term V_{XY}, we adopt the Potts model with static cues [12]. We want to punish two adjacent points assigned with different labels, especially when the two points move in a similar direction. Assuming that a feature point X is not an outlier, it moves parallel to the ground plane, and the direction of its movement coincides with that of its ground projection $P(X)$, which is solvable using one camera. So the smoothness term is defined as:

$$V_{XY}(L(X), L(Y)) = \begin{cases} 0 & L(X) = L(Y) \\ U_{XY} & L(X) \neq L(Y) \end{cases} \tag{7}$$

Fig. 4. Grouping feature points into different vehicles. Left: Delaunay triangulation of unknown feature points. Right: results from multiple labeling. Different groups are color-coded with bold circles being the corresponding label. Yellow points are outliers.

where the static cue U_{XY} adds a larger penalty when X and Y are moving in the similar direction but assigned different labels:

$$U_{XY} = \begin{cases} \alpha K & \angle(X,Y) \le \theta \\ K & \angle(X,Y) > \theta \end{cases} \quad (8)$$

where $\alpha > 1$, K is a constant provided by the user (see Table 1 for assignments), and $\angle(X,Y)$ measures the angle between X and Y's motion path.

$$\angle(X,Y) = \arccos\left(\frac{\Delta P(X) \cdot \Delta P(Y)}{\|\Delta P(X)\| \|\Delta P(Y)\|}\right) \quad (9)$$

For feature-based approaches, the reconstructed points are not very useful unless they are grouped into different vehicles. Given that their 3D coordinates are known in each frame, this problem is often expressed as segmenting a graph into connected components based on spatial and motion similarities [4, 1].

Notice that our energy function is minimized upon the same criteria, so we can gain the grouping directly from the labeling results. The only problem is that there might exist more than one label for each vehicle, and the associated feature points must be merged together.

Since the points associated with a vehicle share the same motion vector, merging can be achieved by adjusting the parameters of our data term: punishing inconsistent motion more, but spatial distance less (by suppressing λ), so that two vehicles can still be separated (because of different motions), but within a single vehicle, one unique label will dominate.

On our experimental data, we find that the breaking point is given by the following assignment of parameters:

Table 1. Parameters for multiple labeling as defined in Equation (6-9).

σ	λ	α	K	θ
0.8	0.2	2	0.4	0.1

3. EXPERIMENTAL RESULTS

We tested our system on two camera installments, with the cameras oriented 60° and 30° to each other respectively. Each surveillance video is about 10*min* in length and contains approximately 300 vehicles. For each vehicle, its tracking result is manually inspected when it reaches a pre-defined point on the road and classified into the following categories:

1. Not matched (NM): if during the first stage, no correspondence is established on this vehicle.
2. Under-segmented (US): if the labeling optimization does not differentiate it from adjacent vehicles.
3. Over-segmented (OS): if more than one label exists on the vehicle after optimization.
4. Correct (C): if it does not fall into any of the failure cases above.

Table 2. Results with different camera configurations.

Relative viewing angle	NM	US	OS	C
60°	2.2%	11.1%	2.2%	84.4%
30°	10.0%	0.0%	0.0%	90.0%

Our system gives comparable performance to other feature-based approaches, without using cues from foreground classification or processing a block of frames. However, our system is ready to incorporate any of those methods if higher performance is required. For example, in our first test, many under-segmented situations happen because, when a vehicle reaches the pre-defined point, it is moving along with another vehicle in the adjacent lane, at an almost identical speed. If we extend our energy function to include information from more than one previous frame, this problem is likely to go away.

4. REFERENCES

[1] N. Kanhere, S. Pundlik, and S. Birchfield, "Vehicle segmentation and tracking from a low-angle off-axis camera," *CVPR*, 2:1152–1157, 2005.

[2] J. Lou, T. Tan, W. Hu, H. Yang, and S.J. Maybank, "3-d model-based vehicle tracking," *IEEE Transactions on Image Processing*, 14(10):1561–1569, 2005.

[3] X. Song and R. Nevatia, "A model-based vehicle segmentation method for tracking," *ICCV*, 2:1124–1131, 2005.

[4] D. Beymer, P. McLauchlan, B. Coifman, J. Malik, "A real-time computer vision system for measuring traffic parameters," *CVPR*, 1997.

[5] R. I. Hartley and A. Zisserman, Multiple View Geometry in Computer Vision, *Cambridge University Press*, second edition, 2004.

[6] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *PAMI*, 27(10):1615–1630, 2005.

[7] Z. Zhang, "A flexible new technique for camera calibration," *PAMI*, 22(11):1330–1334, 2000.

[8] J. Shi and C. Tomasi, "Good features to track," CVPR, 593–600, 1994.

[9] P. Moreels and P. Perona, "Evaluation of features detectors and descriptors based on 3d objects," *IJCV*, 73(3):263–284, 2007.

[10] X. Ma and W.E.L. Grimson, "Edge-based rich representation for vehicle classification," *ICCV*, 2:1185–1192, 2005.

[11] D.G. Lowe, "Distinctive image features from scale-invariant keypoints," *IJCV*, 60(2):91–110, 2004.

[12] Y. Boykov, O. Veksler, and R. Zabih., "Fast approximate energy minimization via graph cuts," *PAMI*, 23(11):1222–1239, 2001.