Speed Estimation Thanks to Two Images from One Stationary Camera

Charles Beumier

Signal & Image Centre, Royal Military Academy, 1000 Brussels, Belgium
Charles.beumier@elec.rma.ac.be

Abstract. This paper presents speed estimation of a moving object thanks to two images captured within a known time interval from one stationary uncalibrated camera. The development is currently applicable to rigid objects animated by a pure translation and requires the localization of corresponding points in both images and the specification of one real dimension. An interesting solution based on an equivalent stereo problem is suggested. It considers the object stationary and searches for the virtual camera motion which would produce the same images. The mathematical formulation is simple using 3D vectors and the camera parameters: focal length, CCD size and pixel size. The developed software package was tested for vehicle speed second assessment of the velocity captured by the LIDAR system LMS-06 distributed by secuRoad SA.

Keywords: Speed estimation, 3D translation, vehicle speed, LIDAR.

1 Introduction

More and more control systems are based on cameras. These generally offer flexibility, limited costs, easier installation and maintenance compared to alternative solutions. Images possibly contain additional information directly interpretable by any analyst. The use of camera for control has also been observed in traffic surveillance [1], application domain of the present work.

In the literature about vehicle speed estimation with one camera, most techniques track individual objects over a sequence of video images and derive a displacement in pixel. This displacement is converted into speed thanks to the known time interval and a conversion pixel to meter taking into account the homography which maps 3D world to image coordinates [2, 3, 4]. Refer to [2] for a precise explanation of the homography and its automatic estimation. In [3], the camera is automatically calibrated from image lane markings to get the world to image projection. In those references, real distances about lane marking are used. Strictly speaking, as mentioned in [2], features should be tracked at the road level since the homography, derived from lane markings, is only valid at that level.

In the present work, we propose to estimate the speed of any rigid object animated by a translation from two images with timestamps captured with a stationary uncalibrated camera. Compared to studies concerned with traffic surveillance for which the camera tilt is important, our camera is pointed to capture the frontal or rear licence

L. Alvarez et al. (Eds.): CIARP 2012, LNCS 7441, pp. 854–861, 2012.

plate for vehicle identification and speed estimation. The high camera resolution allows for a precise estimation of the 3D object motion thanks to triangulation and licence plate dimensions. The program developed has been applied to images captured by the cameras of a LIDAR system for vehicle speed second assessment.

In what follows, section 2 states the problem of speed estimation with camera and gives associated hypotheses. Section 3 details the mathematical background used for motion estimation. Section 4 describes the application for vehicle speed estimation and section 5 presents motion and speed estimation. Section 6 concludes the paper.

2 Problem Statement

The presented approach has been originally developed for vehicle speed estimation as second assessment for the LIDAR speed system LMS-06 used for law enforcement. For certification in Belgium, the deviation between speeds measured by each method must not exceed 10 %.

The LIDAR system LMS-06 consists of a sweeping laser which extracts horizontal profiles of range distances. Speed is estimated from profile range values over time. Compared to Doppler radar, the LIDAR is much more precise in speed and localisation. Several road lanes may be supervised simultaneously with their own speed limit. A LMS-06 system can control traffic in both driving directions and is equipped with two cameras pointing in opposite directions (Fig. 1.). A typical functional setup is to measure LIDAR speed when the vehicle is approaching and if this speed exceeds the limit, a set of images are captured. Up to two images can be taken by each camera, the first camera capturing the vehicle front and the second one its rear.

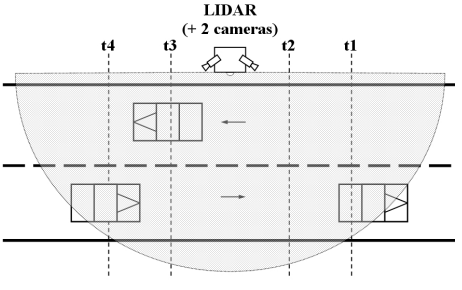

Fig. 1. LIDAR system with two cameras

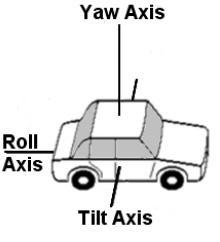

Fig. 2. Vehicle angles

The speed measured by camera is estimated from the distance travelled in a time interval. The time interval, of about 250 ms in our application (due to camera latency time between shots), is known with a precision of +/- 1 ms. Motion estimation is derived from points localised in two images. In our application, 3 hypotheses simplify the approach and its implementation.

First, the camera is expected to be stationary, as claimed by the LIDAR system operator. This simplifies the approach which can focus on object displacement, without estimating the camera motion. The stationary condition is easily confirmed. Secondly,

motion can be approximated by a translation. This is certainly true for vehicles on a straight lane. Vehicle flows are generally approximated by lines as mentioned in the introduction. Considering the three possible angles of a vehicle (Fig. 2.), tilt mainly corresponds to braking or acceleration, yaw to lane changes and roll intervenes when in a curve. These rotations are likely to be negligible in our application. We finally suppose that rigid object points are visible in both images and that at least one real dimension is available. In the vehicle application, the license plate corners are valid candidate points since the plate is used for vehicle identification and is normally of known size.

3 3D Motion Estimation

We propose 3D motion estimation thanks to object points whose projections are specified in both images. The simple pinhole model (3.1) is used to derive 3D lines from both images that should intersect (3.3) at best if the proper motion is found (3.4).

3.1 Camera Model

The camera model used here to solve 3D motion estimation is the classical pinhole model. The perspective projection depends on the focal length and the CCD pixel size (derived from the CCD size and image resolution).

Due to the quality of the image sensors (CCD of Nikon D70S and D90) and the relatively large focal length values (initially 35 or 50mm), distortion parameters could be neglected. In such a situation, a simple relation links the 3D coordinates of a point (X,Y,Z) and its projection in the images (x,y), especially if the optical centre of the camera is used as object coordinate centre Oc (0,0,0).

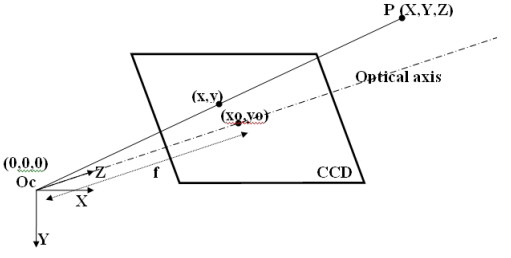

Fig. 3. Axis system and pinhole model

Instead of the analytical formulation linking x,y and X,Y,Z, we preferred a geometrical view which simply need vectors. The 3D line through a scene point P(X, Y, Z) and Oc will intersect the image plane at coordinates (Kx*(x-xo), Ky*(y-yo), f), all expressed in mm in our real world coordinate system. Kx and Ky are the pixel size in mm and (xo, yo) are the image coordinates of the optical centre (supposed to be at the image centre). Conversely, an image point (x, y) directly defines the 3D line specified by point Oc = (0,0,0) and the 3D vector (Kx*(x-xo), Ky*(y-yo), f).

3.2 Triangulation

Triangulation refers to the localization of a 3D point thanks to two projections of known point of view.

We adopted a reverse approach to object motion. We tried to identify the virtual camera motion which would compensate for object motion, allowing for a static description of object 3D points. The problem is then equivalent to stereo computing, where a moving camera captures two images of a stationary object (Fig. 4.).

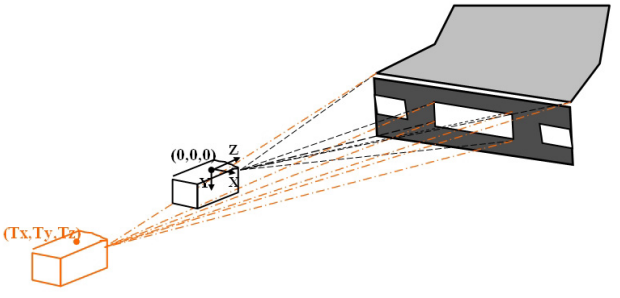

Fig. 4. Triangulation for a virtual camera motion and stationary object

Using triangulation enables to enforce that image points from both images are consistent projections of the 3D object. The problem defined as such is ill-posed, as similar objects of different scales at different distances can produce the same projections. To solve this scale indeterminate, one distance on object has to be specified.

Triangulation simply consists of 3D line intersection. More specifically, for each pair of corresponding points in images, a 3D line is constructed with optical centre 0c (0,0,0) for image1 and optical centre (Tx,Ty,Tz) for image2. The virtual motion of the camera is formalized by the translation of the optical centre of image 2.

The (stationary) coordinates of 3D object points are found at the intersection of two 3D lines. To follow with our geometrical formalism based on vector processing, we designed an elegant and fast algorithm detailed in the next section.

3.3 3D Line Intersection

3D line intersection is not as simple as for planar line intersection since two 3D lines are more likely to have no intersection (even if not parallel). The geometrical algorithm presented here first computes the intersection error which is the minimal distance between the two lines. Then it returns the intersection position which is the midpoint of the shortest segment separating the two 3D lines.

Consider the two 3D lines of Fig. 5, each specified by a point and a vector (p1, **v1** and p2, **v2**). **v1** and **v2** define a normal vector **pln**, perpendicular to **v1** and **v2** and simply obtained by vector product **v1*v2** and normalisation to have a unit vector.

From the family of parallel planes with perpendicular direction **pln**, the distance between planes is in the difference of the scalar product **p.pln**, where p has the coordinates of a point belonging to a plane. The distance between the two 3D lines is the distance between the planes containing p1 and p2 respectively, hence *dist* = (**p1-p2).pln**. Mention that the distance 'dist' is signed.

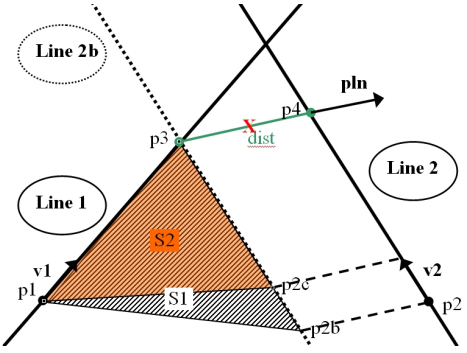

Fig. 5. 3D line intersection

The coordinates of the intersection needs a definition because two 3D lines only intersect if 'dist' is 0. We decided to look for the mid point where the distance between the two lines is minimal. Consider the dashed line 2b in Fig. 5., parallel to line2 and translated by the signed distance 'dist' obtained previously. Line 2b, in the plane (p1, **v1**), intersects line 1 at point p3. Surface S1 of triangle (p2b p1 p3) is given by half of ((**p1**–**p2b**)*k**v1**).**pln** (k, unknown, will disappear). For point p2c, which is p2b shifted by **v2**, we have surface S2 = ((**p1**-**p2c**)*k**v1**).**pln**. The ratio of S2 and S1 allows in one operation to get where point p3 is located: from p2b, along **v2** and at a distance of S1/(S1-S2) (independent of k).

Similarly point p4 could be estimated, but the desired midpoint, halfway between p3 and p4 is directly obtained from p3, at a distance dist/2 in the **pln** direction. Special conditions, like parallel or coinciding lines, are easily detected in our algorithm.

3.4 Translation Estimation

To sum up, object speed estimation requires in our approach the virtual camera translation which makes both images the valid projections of the 3D object. The object is materialized by a set of 3D points localized by the user in both images. Each pair of corresponding image points defines two 3D lines whose intersection is computed to derive an intersection position and error.

The virtual camera translation (Tx,Ty,Tz) associated to image 2 is obtained from the minimization of the error E consisting of two terms:

$$E = w1*RMS (dist(pt)) + w2*RMS\_plate\_size$$

The first term concerns the reconstruction error of all 3D points *pt* and is computed as the root mean square (RMS) value of the intersection errors (*dist* in Fig. 5). The second term, associated with scaling, is the RMS value of the differences between known distances (at least one is required) and reconstructed distances (distance between intersection positions). Both terms can be weighed according to their relative importance (we used a balanced weight: w1 = w2 = 0.5).

The minimization procedure to find the smallest error E follows a coarse-to-fine approach. The Tx, Ty and Tz values span their range with some increment at a level and their best estimates (minimum of E) are used as central values for the next level which will be evaluated in a smaller extent and with finer increments. In order to limit

computation time, it is advised to restrict the ranges of each parameter, as explained below for our application.

After optimization, the algorithm returns the residual error E and the individual deviations for all points and given distances.

4 Vehicle Speed Estimation

4.1 Graphical Interface

The developed system has a graphical user interface displaying two images and superimposed points which can be manipulated (added, moved or deleted).

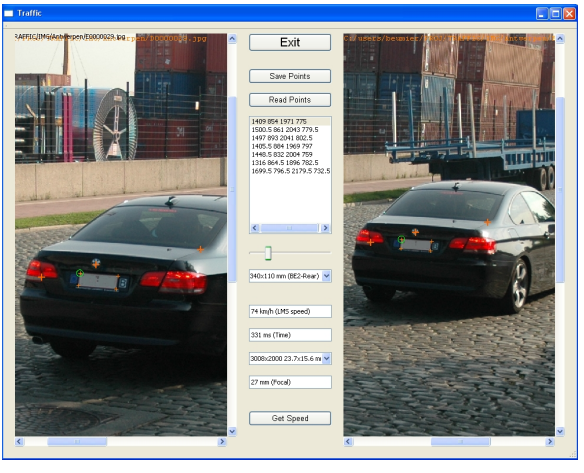

Fig. 6. Graphical interface: 2 images with points and camera parameters

The four licence plate corners must be first localised manually and the plate dimensions have to be specified (country, front/rear, or values typed in). A few additional points, preferably at another range (car bonnet, window corners) make speed estimation more robust. Manual point localisation represents a small time overhead which satisfies the client compared to the risk of failure of a fully automatic solution. Finally, camera details have to be specified (focal length, CCD size in pixel and mm).

Once all the information has been entered, the system can search the optimal translation and derive the vehicle speed and its deviation with the measured LIDAR speed. The residual error E is displayed for all the coarse-to-fine levels to highlight possible convergence problems. For the best solution, the residual error for each point and for the four plate sides is listed. This enables the detection of bad point localisation.

4.2 Uncontrolled Deviation

Our camera based speed estimation has been compared with LIDAR measurements considered as ground truth. Some uncontrolled influences explain their difference.

First, the LMS-06 LIDAR system gets its velocity estimation from the average of measures taken frontally before any image is acquired. The instants of LIDAR speed measurement and camera capture may differ from a few tenths of second to more than one second.

Secondly, the focal length is returned by the camera as an integer value in mm. For 25 mm, this may represent +/-2 % error on speed estimation.

Thirdly, point localization is crucial to the optimization. Although the operator is warned by the reported point residuals, the precision is limited by the object size in pixel. This is particularly true for small focal values.

Fourthly, licence plate dimensions may differ from the official size.

5 Results

Two types of tests were conducted with a vehicle plate as object of interest.

The first test considered a stationary vehicle placed at known relative distances on a line. In the following table, the measured and estimated distance between successive positions are given. The first position P0 is about 10m from the camera and P6 at about 36m. After 36m, the plate has only 80x15 pixel and the lack in resolution leads to imprecise estimation notified by a very large residue. This test made us confident to perform speed measurements in real traffic situations.

Table 1. Estimated distance in the case of a stationary vehicle moved along a line

Distance	**P0-P1**	**P1-P2**	**P2-P3**	**P3-P4**	**P4-P5**	**P5-P6**	**P6-P7**
Measured	3.00	3.00	5.00	5.00	5.00	5.00	5.00
Estimated	3.17	3.02	5.01	5.03	4.79	4.74	3.10
Residue	3.68	5.63	6.51	6.37	3.91	4.96	31.84

The second test concerned vehicle speed estimation in real circumstances. Analysing the convergence data, we could check that the estimated translation vectors have a similar direction, arguing for a rectilinear movement. The global residue ranged roughly from 3 to 10 mm after point refinement. Individual intersection error maxima amounted to 5 mm for worse points after refinement although the majority of point errors lied below 2 mm.

The image based speeds were compared to the LIDAR measurements. Although influences mentioned in 4.2 were expected, the large majority of the tests revealed a speed deviation inferior to 10%, as required by the client. Worst deviations were observed when the driver braked, what can be checked in rear pictures thanks to the brake lights. There was no large positive speed deviation (like a strong acceleration). The speed deviation average of more than 100 tests in different conditions (different cameras, focal length, speed and plate type) for acceptable situations (no brake) is about -3%.

More difficult cases concern vehicle slow down due to road or traffic conditions (hill climbing, road crossing). There is little evidence of such conditions if the driver simply releases gas without braking.

6 Conclusions

This paper has presented an algorithm for speed estimation based on two images with known timestamps captured by a stationary camera.

The solution has been developed so far for pure translation movements. It is based on the 3D reconstruction of points manually localised in the images with at least one known distance. It has been applied to vehicle speed estimation on straight lanes. Many tests confirmed the translational motion hypothesis and showed that the estimated velocity lies within 10% of a LIDAR measurement in normal conditions.

We hope to get test data with better speed stability or with a closer ground truth in order to get better statistical results about speed deviation.

References

1. Kastrinaki, V., Zervakis, M., Kalaitzakis, K.: A survey of video processing techniques for traffic applications. In: Image and Vision Computing, vol. 21, pp. 359–381. Elsevier, Amsterdam (2003)
2. Grammatikopoulos, L., Karras, G., Petsa, E.: Automatic estimation of vehicle speed from uncalibrated video sequences. In: Proc. of Int. Symposium on Modern Technologies, Education and Professional Practice in Geodesy and Related Fields, Sofia, pp. 332–338 (2005)
3. Tocino Diaz, J.C., Houben, Q., Czyz, J., Debeir, O., Warzée, N.: A Camera Auto-Calibration Algorithm for Real-Time Road Traffic Analysis. In: Proc. of VISAPP 2009 Int. Conf. on Computer Vision Theory and Applications, Lisboa, Portugal, pp. 625–631 (2009)
4. Beymer, D., McLauchlan, P., Coifman, B., Malik, J.: A Real-Time Computer Vision System for measuring Traffic Parameters. In: Proceedings of Computer Vision and Pattern Recognition, pp. 495–501. IEEE (1997)
5. Beumier, C.: Vehicle Speed Estimation from Two Images for LIDAR Second Assessment. In: Proc. of VISAPP 2012 Int. Conf. on Computer Vision Theory and Applications, Roma, Italy (2012)