Abstract

This paper presents a scheme of camera ego-motion estimation through locating the focus of expansion (FOE). We showed that the bilinear constraint [2] leads to a sub-optimal solution of motion parameters in the sense that it does not correspond to maximum likelihood estimate. The contribution of the paper is that we study different error metrics, evaluate the metrics, and propose to use two normalized error metrics under dependent and independent noise model, respectively. They are demonstrated to be optimal in the sense of maximum likelihood. In addition, based on the bilinear nature of the objective functions, we propose to use some specific optimization algorithms to achieve efficient and accurate convergence. Robust estimation problem is also addressed to handle outliers caused by independent motions. Promising results have been obtained in experiments. The estimated motion parameters can be used to detect various independently moving objects on the road.

1. Introduction

As a camera moves with respect to a scene, the image changes over time. The goal of ego-motion computation is to estimate 3D camera motion from a sequence of images. Techniques for computing ego-motion from image sequences can be categorized either as discrete-time methods or as instantaneous-time methods, depending on whether the input is image displacement or image velocity [10]. In this paper, we concentrate on instantaneous-time algorithms. A reasonably good optical flow estimate for each frame is assumed already available as the input for the system. The ego-motion problem is to estimate the 3D translation and rotation from a collection of flow vectors sampled at some image positions.

Much of parameter estimation is about choosing the best function to minimize, choosing the best method to optimize, and choosing the mathematical model to describe the running system, including a model of the error process [12]. Accordingly, different ego-motion estimation methods derive from the basic equations [2, 6, 10] different variations as the objective to minimize, and choose different optimization algorithms to find the optimal parameters. In the case of planar surfaces under perspective projection, eight-parameter quadratic flow field is an exact model and can be solved by several approaches [9]. When the depth of each object point varies arbitrarily, Heeger and Jepson [5] proposed a two-step least squares solution for the motion parameters and it can be solved by sampling the parameter space and seeking the minimum. At the same time, they also proposed a subspace algorithm [6] to recover 3D motion and depth information by successively solving three linear problems. But it brings bias to the estimates. In [2], some algebraic manipulations is applied on the basic equations (1) to remove depth and a bilinear constraint on translational and rotational velocity is obtained for each pixel. The authors [10] did an extensive experimental study of comparing the bilinear constraint [2] and other work such as [6] and [7]. It is shown that among several popular algorithms, the bilinear constraint gives the best result in terms of bias and sensitivity to noise.

In this paper, we will show that the bilinear constraint is sub-optimal in the sense of maximum likelihood. We present two normalized error metrics that are optimal under the assumption of dependent and independent Gaussian noise model, respectively. Since noise is unavoidable in flow estimate, discussion of noise model is important for ego-motion estimation using optical flow. The paper starts with a mathematical formulation of the problem in section 2. Maximum likelihood estimate (MLE) of the parameters is derived for different noise model in section 3. Section 4 provides further interpretation of the MLE solution. Our optimization and robust estimation scheme are described in section 5 and 6, respectively. Section 7 briefly discusses the application of the 3D motion estimation to independent motion detection. Experimental results, both simulation and real applications are demonstrated in section 8. Section 9 concludes the paper.
2. Problem formulation

Camera ego-motion estimation is to solve a parameter estimation problem. The basic equations are well-known:

\[ \bar{v}(\tilde{x}) = A(\tilde{x}) \left( \frac{1}{Z(\tilde{x})} \tilde{T} + \frac{1}{f} \bar{\omega} \times \tilde{x} \right) \]  

(1)

\[ \bar{A}(\tilde{x}) = \begin{bmatrix} f & 0 & -x \\ 0 & f & -y \end{bmatrix} \]

where \( \tilde{x} = (x, y, f)^T \) is the image plane coordinate corresponding to a 3D point \( \tilde{X} = (X, Y, Z)^T \), with Z axis pointing along the optical axis of the camera, and X and Y axis are parallel to x and y axis of image coordinates respectively, \( f \) is camera focal length, \( \tilde{T} = (T_X, T_Y, T_Z)^T \) and \( \bar{\omega} = (\omega_X, \omega_Y, \omega_Z)^T \) are translational and rotational motion of the background with respect to the camera, and \( \bar{v}(\tilde{x}) = (v_x(\tilde{x}), v_y(\tilde{x})) \) are the motion flow estimate. From equation (1), \( \tilde{T}, \bar{\omega} \) and \( Z(\tilde{x}) \) can be estimated, with an input of a motion flow field. Translation and depth parameters can be estimated only up to a scale factor due to the scale ambiguity in the perspective projection. Theoretically, a unique solution was proven to exist \[9\]. But in practice the attempts to solve realistic problems were frustrated by the sensitivity to noise, estimate bias, and instabilities of the algorithms. 3D motion estimation is still an intriguing problem today.

A flow field can be thought as the superimposition of two components – a translational component and a rotational one. In the absence of outliers and noise, we would expect the translational component of each pixel to intersect at a common point, the focus of expansion (FOE). With FOE involved, eq. (1) can be written as

\[ \bar{v}(\tilde{x}) = \frac{T_Z}{Z(\tilde{x})}(\tilde{x} - \tilde{x}_0) + \bar{B}(\tilde{x}) \cdot \bar{\omega} \]  

(2)

\[ \bar{B}(\tilde{x}) = \begin{bmatrix} -xy / f & f + x^2 / f & f - y \\ f - y^2 / f & -xy / f & -x \\ 0 & 0 & 1 \end{bmatrix} \]

where \( \tilde{x}_0 = (x_0, y_0, f) = (fT_X/T_Z, fT_Y/T_Z) \) is the FOE coordinate. We aim to estimate \( \Theta = (\tilde{x}_0, \bar{\omega}) \) from the input flow field. Assuming \( \bar{v}(\tilde{x}) \) to be the ground truth of the flow vector at pixel location \( \tilde{x} \), and using \( \bar{v}(\tilde{x}) \) to denote the measurement, the difference \( (\tilde{v}(\tilde{x}) - \bar{v}(\tilde{x})) \) is the noise in the measure of flow. It is assumed to be isotropic zero-mean Gaussian noise with covariance matrix \( \Sigma \). Hence, \( \Sigma \) is a diagonal matrix with two identical entries \( \sigma(\tilde{x}) \).

Due to the Gaussian assumption of noise, and the assumption that the noise at different pixel locations is independent from each other, we have \( \bar{v}(\tilde{x}) \sim N(\bar{v}(\tilde{x}), \Sigma) \), and the maximum likelihood estimate (MLE) of the parameter is

\[ \Theta = \arg \max_{\Theta, Z(\tilde{x})} \prod \left\{ \frac{1}{Z(\tilde{x})} \tilde{T} + \frac{1}{f} \bar{\omega} \times \tilde{x} \right\} \]

(3)

Two different noise models will be considered:
1) independent model: the Gaussian noise independent of the flow vector, i.e. the noise has constant variance;
2) proportional model: the Gaussian noise with standard deviation proportional to the magnitude of the flow vector. Due to the linear dependency on the angular velocity \( \bar{\omega} \) in (2), the error estimate in motion and depth does not depend on the rotational parameters \[4\]. We will assume the noise proportional to the translational component.

3. Error Metrics

In this section, we assume there is no outlier due to independently moving objects. Outlier issue will be addressed in section 6.

If we assume an independent noise model, then the noise is homogenous, meaning the standard deviation \( \sigma(\tilde{x}) \) does not depend on \( \tilde{x} \). Hence, the MLE problem in (3) becomes

\[ \Theta = \arg \min_{\Theta, Z(\tilde{x})} \left\{ \sum \left\| \bar{v}(\tilde{x}) - \bar{A}(\tilde{x}) \cdot \left( \frac{1}{Z(\tilde{x})} \tilde{T} + \frac{1}{f} \bar{\omega} \times \tilde{x} \right) \right\|^2 \right\} \]

(4)

Using the shorthand notation of vand \( \bar{B}, \) and denoting \( p(\tilde{x}) = 1/Z(\tilde{x}) \), the objective (4) can be expressed as

\[ \Theta = \arg \min_{\Theta, p(\tilde{x})} \sum \left\| \bar{v}(\tilde{x}) - p(\tilde{x}) \cdot \bar{A}(\tilde{x}) \cdot \tilde{T} - \bar{B}(\tilde{x}) \cdot \bar{\omega} \right\|^2 \]

(5)

Denote \( f = \sum \left\| \bar{v}(\tilde{x}) - p(\tilde{x}) \cdot \bar{A}(\tilde{x}) \cdot \tilde{T} - \bar{B}(\tilde{x}) \cdot \bar{\omega} \right\|^2 \) , take partials w.r.t. \( p(\tilde{x}) \) at each pixel location \( \tilde{x} \), and set each \( \partial f / \partial p(\tilde{x}) = 0 \). We get for every \( \tilde{x} \)

\[ p(\tilde{x}) = \frac{(\bar{A}(\tilde{x}) \cdot \tilde{T} - \bar{B}(\tilde{x}) \cdot \bar{\omega})^T (\bar{A}(\tilde{x}) \cdot \tilde{T})}{(\bar{A}(\tilde{x}) \cdot \tilde{T})^T (\bar{A}(\tilde{x}) \cdot \tilde{T})} \]

(6)

Substituting the above expression of \( p(\tilde{x}) \) into equation (5) to eliminate the depth, the error metric becomes

\[ f = \sum \left\| (\bar{A}(\tilde{x}) \cdot \tilde{T}) - (\bar{A}(\tilde{x}) \cdot \tilde{T} - \bar{B}(\tilde{x}) \cdot \bar{\omega}) \right\| \]

(7)
Substituting the expression of $\hat{A}$ and $\hat{B}$ and FOE coordinates, the above error metric becomes

$$f_h = \sum_i h_i$$

$$= \sum_i \left( \frac{(v_{xi} - B_{xi}\theta)(y_i - y_0) - (v_{yi} - B_{yi}\theta)(x_i - x_0)}{\sqrt{(x_i - x_0)^2 + (y_i - y_0)^2}} \right)^2$$

where $B_{xi}$ and $B_{yi}$ are the first and second row of $\hat{B}(\bar{x}_i)$, respectively. For notation simplicity, the dependence on pixel location is omitted. This metric was also observed in different forms by others [1, 7]. Compared to the minimization problem in (4), the minimization in (8) does not include the unknown depth of $Z(\bar{x})$ that is location dependent. Hence the problem in (8) is more tractable to estimate the parameters $\Theta = (\bar{x}_0, \theta)$.

If a proportional noise model is assumed, the noise in flow is heterogeneous, meaning the standard deviation depends on pixel location $\bar{x}$. The standard deviation of noise can be expressed as

$$\sigma(\bar{x}) = \sigma \cdot \sqrt{(v_{xi} - B_{xi}\theta)^2 + (v_{yi} - B_{yi}\theta)^2}$$

which leads to error metric (9) by eliminating depth in (3)

$$f_d = \sum_i d_i^2 = \sum_i \left( \frac{(v_{xi} - B_{xi}\theta)(y_i - y_0) - (v_{yi} - B_{yi}\theta)(x_i - x_0)}{\sqrt{(v_{xi} - B_{xi}\theta)^2 + (v_{yi} - B_{yi}\theta)^2}} \right)^2$$

Note that if the bilinear constraint on $\hat{T}$ and $\hat{\omega}$ [2]

$$\hat{T}^T(\bar{x} \times \hat{v}(\bar{x})) = (\hat{T}^Tx \times \hat{\omega}) = 0$$

or equivalently, the instantaneous epipolar constraint [4]

$$(\hat{T}^T x \times \hat{v}(\bar{x}), 0)^T - \hat{\omega} \times \bar{x} = 0$$

is used, the error metric can be written in the form of

$$f_d = \sum_i a_i$$

$$= \sum_i \left( (v_{xi} - B_{xi}\theta)(y_i - y_0) - (v_{yi} - B_{yi}\theta)(x_i - x_0) \right)^2$$

Obviously, $a_i$ in (12) is the numerator of $h_i$ and $d_i$ in (8) and (9), so (12) does not correspond to MLE of $\Theta$. And $h_i$ and $d_i$ are the renormalized metrics from $a_i$.

4. Alternative Interpretation

The MLE of 3D motion parameters $\Theta$ (8) and (9) can also be interpreted more visually and intuitively. In absence of outliers and noise, we expect the translational component of the motion field to intersect at a single point of FOE. Or equivalently, the translational component of flow should lie on the line connecting FOE and the corresponding pixel location. Therefore, it is natural to minimize the deviation from the rotation-compensated flow vector to the line connecting FOE and the pixel.

With independent noise model, the distance from the end point of the rotation-compensated flow vector to the line connecting FOE and the pixel represents the noise level. From a point to a line, there are two distance metrics used frequently, algebraic distance (perpendicular to the horizontal axis) and geometric distance (orthogonal to the line). Here the algebraic distance is

$$a_i = (v_{xi} - B_{xi}\theta)(y_i - y_0) - (v_{yi} - B_{yi}\theta)(x_i - x_0)$$

and the geometric distance is

$$h_i = \frac{(v_{xi} - B_{xi}\theta)(y_i - y_0) - (v_{yi} - B_{yi}\theta)(x_i - x_0)}{\sqrt{(x_i - x_0)^2 + (y_i - y_0)^2}}$$

Note the $a_i$ is the same as in (12), and $h_i$ is what is in (8).

The difference between these two distance metrics can be illustrated as in Fig. 1. Red solid arrows indicate ground truth of the flow vectors, while blue dashed arrows represent the measurements. Assume the same flow vector corrupted by same amount of noise but located at three different pixels. These three pixel locations are chosen to be with same horizontal distance away from FOE so that the three green dashed lines represent the equally scaled algebraic distances. It is obvious that three algebraic distances are location dependent and will contribute differently to the objective function (12). All the geometric distances, shown as three purple solid lines, are of same length, which means they contribute equally to objective (8). We want to make equal use of all the data points contaminated by the same amount of noise, no matter where it is located. So geometric distance is more meaningful and it represents the noise level. We name (8) the geometric metric.

However, with proportional noise model, a more meaningful measure that represents the noise level is $\theta$, the angle between the rotation-compensated flow vector

![Fig.1. Comparison of algebraic distance (green dashed lines) and geometric distance (purple solid lines). Algebraic distance is location dependent while geometric distance is not.](image-url)
and the line connecting FOE and the pixel. In fact, if we assume isotropic Gaussian perturbation for flow vectors, i.e. i.i.d. Gaussian for x and y component of the flow, it is easy to prove that \( \sin(\theta) \) has the same Gaussian distribution. Therefore, \( \sin(\theta) \) is used to indicate the proportional noise level.

\[
d_i = \sin(\theta_i) = h_i / \sqrt{(v_{ix} - B_1^i \omega)^2 + (v_{iy} - B_2^i \omega)^2}
\]

(15)

\[
d_i = (v_{ix} - B_1^i \omega)(y_i - y_0) - (v_{iy} - B_2^i \omega)(x_i - x_0)
\]

\[
\sqrt{(v_{ix} - B_1^i \omega)^2 + (v_{iy} - B_2^i \omega)^2 \sqrt{(x_i - x_0)^2 + (y_i - y_0)^2}}
\]

The \( d_i \) in (15) is the same as that in (9). And it is the renormalized geometric distance. We call error metric (9) the normalized metric.

One alternative way to look at the distance (15) is its relationship with the correlation between two unit vectors lying in the direction of rotation-compensated flow vector and the line connecting FOE and the pixel, and solving the related linear problem.

\[
c_i = \frac{(v_{ix} - B_1^i \omega)(x_i - x_0) + (v_{iy} - B_2^i \omega)(y_i - y_0)}{\sqrt{(v_{ix} - B_1^i \omega)^2 + (v_{iy} - B_2^i \omega)^2 \sqrt{(x_i - x_0)^2 + (y_i - y_0)^2}}}
\]

(16)

It is noticed that \( d_i^2 + c_i^2 = 1 \). It is not surprising since the correlation coefficient \( c_i \) is the cosine of the angle between two vectors, i.e. \( c_i = \cos(\theta_i) \), and we know that \( d_i = \sin(\theta_i) \). Therefore, minimizing sum of squared normalized distance, in fact, is equivalent to maximizing sum of squared correlation between two unit vectors.

In real applications, noise in flow field is unlikely to precisely follow either model. Instead, it is likely to depend on local SNR. So it is advisable to add as another weight factor some function of the standard deviation of the flow estimate. In addition, due to the fact that most flow estimation methods rely on first order approximation of local smoothness of image intensity, usually small motion is more accurately estimated. Therefore, if the depth varies significantly throughout the scene, the proportional noise model will have advantage over the independent noise model.

5. Optimization

In terms of optimization, obviously the algebraic metric (12) is the easiest and the closed-form solution is given by SVD in [7]. Compared to the normalized metric (9), the geometric metric (8) is not as difficult. Noticing that the numerator of (8) is bilinear, we can easily write a least square solution of \( \omega \) in terms of \( \hat{T} \). Then a nonlinear search method can be used to solve for \( \hat{T} \). Many general nonlinear techniques are readily available, such as Gaussian-Newton algorithm, Levenberg-Marquardt procedure, and so on. The convergence performance depends on the cost function surface. It has been proved [4, 10] that several local minima exist on the cost surface. Therefore, the global minimum cannot be easily obtained. For optimization of (9), it is even more complicated. Our experiments demonstrate that direct use of LM method does not work out for the application. We try to exploit the bilinear structure in the objectives.

A general unconstrained bilinear problem is

\[
\min \ Q(\hat{x}, \hat{y}) = c^T \hat{x} + \hat{x}^T D \hat{y} + \hat{d}^T \hat{y}
\]

(17)

where \( c \) is an n vector, \( d \) is an m vector, and \( D \) is an \( n \times m \) matrix. It is a concave programming problem. The problem is linear when either of the vectors \( \hat{x} \) and \( \hat{y} \) is fixed. That is why such problems are termed bilinear programming. A very natural approach to solving such problems, first proposed by Konno (1976), is to iterate between two (sets of) parameters. For solving (17), the method consists in starting with an arbitrary fixed \( \hat{x} \in X \), and solving the related linear problem

\[
\min \ Q(\hat{x}, \hat{y}) = c^T \hat{x} + \hat{x}^T D \hat{y} + \hat{d}^T \hat{y}
\]

The solution of this problem, \( \hat{y} \), is then used to solve another linear problem

\[
\min \ Q(\hat{x}, \hat{y}) = c^T \hat{x} + \hat{x}^T D \hat{y} + \hat{d}^T \hat{y}
\]

(19)

This yields a new value for \( \hat{x} \). This procedure is repeated until a pair of values \( (\hat{x}^*, \hat{y}^*) \) is found that solves both these linear programs. Such a point can be shown to be a KKT point of (17). Although Konno could not theoretically guarantee convergence for the algorithm, no counterexample has been presented to date to show that the method fails.

Considering (9), if we denote

\[
w_i = \left( (v_{ix} - B_1^i \omega_i)^2 + (v_{iy} - B_2^i \omega_i)^2 \sqrt{(x_i - x_0)^2 + (y_i - y_0)^2} \right)^{-1}
\]

(20)

where \( \omega_i \) and \( \hat{x}_i \) mean the estimate of parameters from last iteration, then in current iteration, \( w_i \) can be regarded as constants. The optimization of (9) becomes bilinear least squares. However, due to the structure of weight, a good initialization is required. Considering the continuity of camera motion, it is reasonable to assume that the parameters estimated from the previous frame in a sequence are a good initial value for the next frame. In our implementation, we start the optimization from a set of seed locations around the previous estimated FOE to improve the convergence performance.

6. Robust estimation

As in many applications, the output from motion estimation is not only noisy, but also contains outliers, e.g. flow vectors corresponding to independent motion. The employment of robust estimation methods is essential. Robust estimation methods help to recover
meaningful descriptions of a statistical population even when the data contain outlying elements belonging to a different population. They are also less sensitive to the violation of underlying assumptions, such as the noise model.

Various methods have been proposed to achieve robust estimation in computer vision area including M-estimator, case deletion diagnostics, and RANSAC algorithm. Each of them has certain advantages and disadvantages. It is shown [11] that M-estimators are vulnerable to poor starting conditions and to high percentage of gross outliers. Case deletion is superior to M-estimators in terms of convergence, and the solution was typically more accurate. The disadvantage is that they require a fairly good estimate of error model. RANSAC and its variant LMS have comparatively better performance than the other two categories, especially when there are a big amount of outliers. But the computation is heavy when outlier percentage is high.

In our implementation, case deletion method is employed. It requires the knowledge of the standard deviation \( \sigma \) of the error, as outliers are typically discriminated from inliers using

\[
i \in \begin{cases} 
\text{inliers} & r_i \leq t \sigma \\ 
\text{outliers} & \text{else} 
\end{cases}
\]

(21)

where \( r_i \) is standard for residuals, e.g. \( d_i \) defined in (15), and \( t \) is a threshold. A well-known robust empirical estimate of the standard deviation for Gaussian residual [8] is

\[
\sigma = 1.4826 \left( 1 + 5/(n - p) \right) \sqrt{\text{med} | r_i |}
\]

(22)

where \( n \) is the number of data, and \( p \) is the number of parameters. \text{Med} stands for median.

It should be noted that the outliers are referred to those 2D flow vectors \((v_{ix}, v_{iy})\) who are not consistent with the 3D motion parameters estimated from the majority of the flow vectors. If only \( d_i \) is used to detect outliers, then only one dimension of information is used and that is not enough to detect 2D outliers. We may incorporate the correlation coefficient defined in (16). However, it is not convenient to model the distribution of the correlation (cosine) while the normalized distance (sine) can be reasonably modeled as a Gaussian distribution. We combine these two metrics as an outlier detector.

7. Application to Obstacle Detection

Detecting obstacles on the road is an important task in building vision systems for driver assistance. Various obstacles such as vehicles and pedestrians would induce different image motion over the flow field. Obstacle detection is interested in finding independently moving objects and / or those objects that do not fit the surface model of the driving way.

Independent motion detection is a natural extension of outlier detection. By thresholding certain error metric, it is straightforward to detect most of the independent motion. However, if the independent moving object happens to generate motion vectors in the same direction as if there is not such an object, then it cannot be detected by solely using the proposed method.

Detecting objects that are not in the surface of the driving way can be achieved by plane fitting. However, the main difficult that rises in a traffic surveillance scenario is that most part of the road has no meaningful texture and there is no reliable input of motion vector. Therefore, in most real traffic applications, it tends to be an ill-posed problem.

8. Experimental Result

8.1 Simulation

To compare the three objectives, algebraic metric (12), geometric metric (8), and normalized metric (9) in this paper, we synthesized an optical flow field with white additive Gaussian noise. As mentioned previously, we will use two different noise models. The noise in two directions is independent. In addition, authors claim [4, 6] that by using algebraic error metric (12) the bias of the estimated FOE from the ground truth is not severe when there is rich depth in the scene. To compare the bias under different depth structure, we simulated scenes both with single depth and with rich depth. In the simulation, the focal length is set to be 100 units, and the frame size is 256x128 in the unit of pixel. Assuming the actual pixel size is 1 unit per pixel, the FOV (field of view) is about 104 degree. Different levels of noise are added to the flow field. For dependent noise, 0%, 5%, 10% and 15% of the flow magnitude are used. For independent noise, levels of 0.05, 0.1, 0.15, 0.2, and 0.25 pixel are added to both x and y direction. In addition, different portions of flow data were used, i.e. all the data, 1/2, and 3/4 of the data on the left / right. This step is used to check how sensitive the error metric is to different FOV, and to check how sensitive the error metric is to different location of FOE. This is not artificial because in real applications, not all the pixel locations will produce equally good flow estimate. For the scene with rich depth, we use a random cloud of points placed in the simulated 3D space in front of the camera. The depth ranges from 200 to 2000. For the scene with single depth, the depth is set to 1200. In the results shown below, the bias is measured in degree. It is the angle between the 3D vector of ground truth of FOE and the 3D vector of FOE obtained from experiments. The bias of rotation estimation is not reported due to the limited space. Usually, we repeat the same experiment for 10 times, and get the mean as the experimental result.
8.2 Cost surface – pure translation

First, we simulated the optical flow field without rotation. The synthesized coordinate of FOE is (100, 50). One example of the synthesized flow field free of noise is shown in Fig. 2(a). The noise is dependent on the flow vector. The depth varies across the scene. We aim to compare algebraic and normalized metric. Similar results are observed for independent noise case to compare algebraic and geometric metric. Variation of depth does not lead to any difference from single depth case. We make the flow field asymmetric about the center, which presents more difficult to recover 3D motion. To remove the possible influence from optimization method, the minimum was found by exhaustive search over the sampled solution space with sampling interval of 1 in both x and y directions. The bias is shown in Fig. 3. The horizontal axis represents different parameter settings. Noise levels are labeled. Five cases in each noise level are: using all data, ¼ data at left, ¼ data at right, ½ data at left, and ½ data at right, respectively. We conclude that under pure translation, noise and FOV has significant effect on algebraic metric, no matter the scene contains rich depth information or not. Geometric metric is very robust to both noise and change of FOV. Another point is the result of using the right portion of the data is better than that of using the left portion. Recall that the FOE is located at the right. If the FOE is set at (20,10), the biases from all experimental settings are close to 0. Generally, for all metrics, the more lateral the translation and the smaller the FOV, the more severe are the bias effects.

8.3 Cost surface – general motion

Second, the optical flow field with rotation is simulated. Again, the FOE is set to be (100, 50). We add different level of noise and use different portion of data as before. The rotational components are \( \omega_x = 0.0025, \omega_y = -0.0025, \omega_z = 0 \). One example of the synthesized flow field free of noise is shown in Fig. 2(b). These motion parameters do not correspond to camera fixating. The minimum was found by search over a 5D space, with sampling interval of 1 in both x and y directions, and 0.0025 in all the rotational speed directions. As the search involves huge amount of computation, we only searched locally centered around the true parameters with \( x_0 \) ranging from 80 to 120, and \( y_0 \) from 30 to 70. All the \( \omega \)'s are from -0.01 to 0.01. So the possible value of bias is actually confined within a small range. This experiment is only repeated twice. For the independent noise case, the advantage of geometric metric over algebraic one is significant. Fig. 4 shows the example where there is rich depth throughout the scene. If there is only the single depth, the bias of algebraic metric is larger while the geometric metric results in zero biases. For the dependent noise case, if we compare algebraic and normalized metrics, no matter what the depth structure is, we saw similar results to Fig. 4. But the biases from normalized metric become larger at high noise level.

It is noticed that normalized metric is always better than algebraic one under dependent noise, and geometric metric is always better than algebraic one under independent noise. When the noise becomes heavier and the FOV becomes smaller and more skewed, the advantage is more significant. We also try to compare normalized metric and geometric one. With independent noise present, the normalized metric behaves badly for the obvious reason that it over-emphasizes the noise in small flow vectors. In proportional noise case, the normalized metric is systematically better than the geometric one. If we search minimum locally as previously, a similar result to fig. 5 can be obtained. At each noise level, ten
different numbers of random samples are chosen, i.e. 10, 30, 50, 100, 200, 400, 600, 800, 1000, and 1500 random flow vectors are chosen as the input data. On average normalized metric generates smaller bias than the geometric one does. But in an individual experiment with a specific data set, it is not always the case. We think the reason is that the normalized metric amplifies the round-off noise on small flow vectors. The geometric metric just gives more weights to large flow vectors and ignores in some sense the small vectors. If there are a big number of flow vectors available, and most of the flow vectors tend to have similar magnitude, then it really does not matter to replace the normalized metric with geometric one. On the contrary, if the magnitude of flow vector varies significantly and especially if there are only few large vectors, then geometric metric tend to emphasize only those large vectors and generate big biases. In real application, especially in traffic surveillance scenario, due to the fast motion of the camera, the wide range of depth, and the first order approximation of local smoothness of image intensity, normalized metric tends to have advantage.

Another issue is motion ambiguity, i.e. the existence of local minima on cost surface makes it difficult to solve the 3D motion problem with noise and limited FOV. For example, we still set the FOE at (100, 50). The rotational velocity is chosen so that the camera is fixating at the center of the frame, i.e. \( \omega_x = -0.0125, \omega_y = 0.025, \omega_z = 0 \). No noise is added to the motion field. It turns out that both geometric and normalized metrics at (0, 0, 0, 0, 0) are 0 and smaller than those at (100, 50, -0.0125, 0.025, 0). The discrepancy is possibly due to the limited resolution. Based on the objective value, (0, 0, 0, 0, 0) is a better solution than the true one. But both solutions are local minima on the objective function surface. If the starting point is close to the true solution, a regular optimization algorithm will have a convergence at the vicinity of true solution.

8.4 Optimization and convergence

Using the synthesized flow field without rotation, the minimum is always guaranteed to be obtained in a very small number of iterations.

However, with rotational velocity, the cost function manifold in 5D space is far more complicated. As introduced early, we choose to start from a group of seed locations and then iteratively optimize rotation and translation. We will not report the optimization result of algebraic metric here because it is shown to have severe bias effect. To compare the convergence performance of algorithms with geometric and normalized metrics, we use same motion parameters on the scene with rich depth. Starting points are chosen at a grid with the interval of 20. The starting points are chosen to be as far away as possible from the ground truth of the parameters. The bias computed from two different metrics are shown in Fig. 5. The noise levels are marked. At each noise level, same ten numbers of random samples are chosen as the input for optimization. Again, we find that the normalized metric is systematically better than the geometric one. In real applications, considering the comparative ease of optimizing the geometric metric, we find it advisable to use geometric metric if the noise is not overwhelming, and there are good amount of data available, and the magnitude of flow vectors do not vary dramatically. Otherwise, normalized metric is suggested.
We tested the proposed algorithm on some real traffic video sequences, with independent moving objects. Biased Least Squares (BLS) and Variable Bandwidth Data Fusion (VBDF) are the techniques used for motion estimation [3]. Motion flow estimate for each pixel has an independent Gaussian distribution with covariance matrix proportional to the variance of the image noise, based on the property of BLS solution. Magnitudes of the flow vectors vary a lot. Normalized metric is used to locate the FOE and to estimate rotational motion of the camera. The result is visually satisfactory as long as the estimate of motion vectors is reasonably good and there are more inliers than outliers. In the selected frames shown below (Fig. 6), the transparent red labels independently moving objects, yellow dot shows estimated FOE, and blue lines with red-dot-end indicate estimated optical flows. No post-processing steps are adopted.

8.5 Real image sequence

9. Summary

This paper compared different error metrics for camera ego-motion estimation, including algebraic metric, geometric metric, and normalized metric. We analyzed noise model and proposed proper (in the sense of maximum likelihood) error metric for certain noise assumption. The performance of the metrics were demonstrated and verified by the experiments. Combined with certain optimization technique and robust estimation scheme, the normalized metric is shown to be promising in the application to some real traffic image sequences.