

Handling Small Motions without Differential Approximation

Siying Liu and Karianto Leman
Institute for Infocomm Research, Singapore
E-mail: {liusy1, karianto}@i2r.a-star.edu.sg

Abstract—Let us consider the scenario of a moving camera capturing a video sequence. If we were to implement a purely discrete algorithm to recover structure, then adjacent frames are considered to contain no useful information and are often used solely for the purpose of facilitating point tracking. On the other hand, there are many differential algorithms that are meant to deal with small motion between adjacent frames but fail when the motion is too large. In this paper, we observe that the difference between these two classes of algorithms is often artificial. Proper normalization of the data can enable the discrete algorithms to handle differential data without the noise associated with the differential approximation. We term this method “Time Normalization” (TN). We show how TN can be used to overcome degeneracies (or instabilities) under existing vision algorithms. We also provide a geometrical understanding of the problem.

I. INTRODUCTION

The traditional discrete Structure from motion (SfM) algorithms are designed to handle only a few frames (two for the essential matrix, three for the trifocal tensor). As such, there have been numerous attempts to integrate additional frames into this discrete framework. The purpose of these formulations vary from formulating matching constraints as demonstrated by Dornaika et al. [1] to increasing the robustness of camera pose recovery as proposed by Zhang et al. [12]. However, these algorithms have a common theme in that they can be classified as being either differential or discrete. Seminal works on differential formulation such as those by Triggs et al. [8], Åström et al. [4] and Heyden et al. [2] require the motion to be small and the additional frame to be close to the original set of frames. On the other hand, the discrete algorithms [12], [1] require a large motion and become degenerate if the motion is small. However, since a large motion can be viewed as a result of a series progressive small motions, it thus raises a question on how to bridge the gap between these two types of formulations.

In this paper, we seek to address the negative aspects of differential algorithms mentioned above, while preserving their advantages. In the first half of the paper, we focus on understanding the performance of discrete algorithms when faced with small motion. As such, we revisit the claim made in [8] which states that theoretically, nothing is gained by a differential formulation. It argues that since in practical vision problems, all motions are discrete, the traditional discrete algorithms can be used to handle the motion information. Using the case of the essential matrix [3], we show that while [8] is correct in pointing out that the motion information

is encapsulated in the discrete algorithms, the said information is improperly weighted. Hence, it is often drowned out by the discrete information. To remedy the problem, we develop a smooth transition from discrete to differential, termed “time normalization” (TN) that allows the motion information to be extracted from the existing discrete information and weighted properly. It serves as a generic means to extend the discrete algorithms so that they can seamlessly incorporate small motion information. Such a formulation would have the benefit of avoiding small motion approximation and thus evading the noise that comes with it. On the more practical side, our algorithm allows us to make use of closely spaced still images rather than continuous video streams. This means great savings in memory, higher shutter speed and better resolution. The proposed formulation will allow us to adapt [12], [1] so that they can handle infinitesimally small motion. Conversely, we can develop formulations which are similar to those proposed by [8], [4] but do not need to incorporate the noise caused by approximating discrete motion as differential. Other related papers are [5], [9], [10]. Perhaps most importantly, we also manage to preserve the linearity and discrete nature of the problem, alleviating the need for differential approximation.

II. STEREO RIG VIEWING A MOVING SCENE

Let us consider a stereo rig that is viewing a moving scene. Define P and P' be a 3D feature point in the first and second cameras' coordinate frames,

$$P' = R(P - T), \quad (1)$$

where R and T are the rotation and translation relating the two views.

As time evolves, the same feature point moves by ΔP and $\Delta P'$ in the respective coordinate frames. They are related by:

$$P' + \Delta P' = R(P + \Delta P - T). \quad (2)$$

Putting the points in pixel coordinates, the essential matrix constraint is given by:

$$p^T E p' = 0, \quad (3)$$

$$(p + \Delta p)^T E (p' + \Delta p') = 0. \quad (4)$$

where p and p' represents the pixel coordinates of point P in the two cameras. (Note: $E = \hat{T}R^T$, where \hat{T} denotes the skew symmetric matrix associated with T .)

If Δp and $\Delta p'$ are small, then both equations (3) and (4) are very similar. If the motion is such that it is infinitesimally

small, e.g., optical flow, then the traditional essential matrix constraint will indicate that the motion does not introduce any constraint at all. We know intuitively that the above statement is incorrect. Given two cameras viewing an object in motion, the direction of motion of the object in each camera will give us an indication of the camera's pose relative to each other. As such, the question now becomes one of how this information can be included into the representation of camera pose. A simple way to incorporate the motion information into the essential matrix representation is to differentiate equation (3). This will result in

$$\dot{p}^T E p' + p^T E \dot{p}' = 0. \quad (5)$$

However, this equation is highly unsatisfactory since it has no geometric meaning. Further, it is purely differential in nature and as such is susceptible to large amounts of noise should the motion be large. We address both of these issues in the next two subsections.

A. Geometric derivation of the optical flow constraint

We know that the optical flow of a point places a constraint on its 3-D velocity $[V_x \ V_y \ V_z]^T$ such that

$$V = \lambda v + \dot{\lambda} \hat{p}. \quad (6)$$

where $v = \dot{p}/|p|$, $\hat{p} = p/|p|$. λ is a scalar in terms of f/Z in a calibrated camera¹. The $\hat{\cdot}$ sign is used to denote a unit vector and the $\dot{\cdot}$ represents derivative.

We know that the component of the 3D velocity perpendicular to the normal of the epipolar plane \hat{n} must be the same in both views. Therefore,

$$V^T \hat{n} = (V')^T \hat{n}' \Rightarrow \lambda v^T \hat{n} = \lambda' (v')^T \hat{n}'. \quad (7)$$

Observe that

$$E \hat{p}' = -\hat{n} |T| \sin(a2), \quad (8)$$

$$\hat{p}^T E = \hat{n}' |T| \sin(a1). \quad (9)$$

Substituting equations (8) and (9) into (7) to remove \hat{n} and \hat{n}' we can obtain

$$-v^T E \hat{p}' = \hat{p}^T E v' \frac{\lambda' \sin(a2)}{\lambda \sin(a1)}. \quad (10)$$

Using the sine rule and making some rearrangement in the terms, we can obtain

$$\hat{p}^T E v' + v^T E \hat{p}' = 0. \quad (11)$$

Note that (11) differs from equation (5) only by a scale factor $|p||p'|$. Therefore, we can conclude that equation (5) is a constraint that equates the feature points' motion component that is parallel to the normal of the epipolar plane. This is intuitively correct, since we cannot equate the motion components within the epipolar plane as the motion seen in one view places no constraint on the motion seen in the second view. (Note: Equation (11) does not incorporate the rigid motion constraint. i.e. Every point can move independently.)

¹Let $V = P_2 - P_1$, we have $V = \lambda_2 \hat{p}_2 - \lambda_1 \hat{p}_1 = \lambda_2 (\hat{p}_1 + v) - \lambda_1 \hat{p}_1 = \lambda_2 v + (\lambda_2 - \lambda_1) \hat{p} \Rightarrow v = \lambda v + \dot{\lambda} \hat{p}$

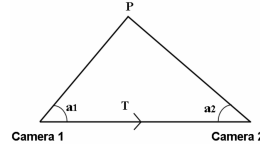


Fig. 1: Epipolar geometry of two views.

To summarize our findings thus far, we have shown intuitively that the motion information is poorly represented (in the case of pure differential motion it would be totally unrepresented) by the traditional discrete essential matrix equations given in (3) and (4). Despite the fact that equations (3) and (4) indicate otherwise, there is geometrically meaningful information present in the differential motion. This is in contrast to [8] that apart from the advantages of linearity, there is no difference between the differential and discrete forms of SfM equations.

We have shown that the differential equation in (5) is a meaningful constraint on the essential matrix, it remains to obtain a smooth transition from the discrete to differential domain. In this regard, we note that if we take the subtraction of (3) and (4) we will obtain the equation

$$\Delta p^T E p' + p^T E \Delta p' + \Delta p^T E \Delta p' = 0. \quad (12)$$

When we have $\lim_{\Delta p' \rightarrow 0}$ and $\lim_{\Delta p \rightarrow 0}$, equation (12) will reduce to (5). (12) has the advantage of properly representing the differential information in the limiting case when the motion is infinitesimally small. However, due to the extra $\Delta p^T E \Delta p'$ term, it is not dependent on the motion being infinitesimally small.

B. Time normalization

Equations (12) and (3) represent the constraints imposed on the essential matrix by the point matches p and p' and the motion made by these points. These constraints are also present in the traditional essential matrix equations given by (3) and (4) (with the exception of the pure differential case). The crucial difference is that in the traditional essential matrix equations, the information bearing equation (12) is embedded in (4) in the form (12)+(3), where the terms in (3) are much greater than those in (12). Hence, nearly all the motion information in (12) is drowned out by being added to the already known constraint (3). By separating out the information bearing constraint and weighting it appropriately, we can finesse the problems posed by the traditional discrete formulation while preserving both the discrete and linear properties of the original equations.

C. Weighting

To compute the essential matrix E linearly, we can use (3) and (12) to create a $2N \times 9$ matrix, A , such that $A(E^T)^s = 0$. Where $(E^T)^s$ is the stack of E^T . We now address the issue of how different rows of A should be weighted.

There are essentially two kinds of rows in A . Those derived from (3), forming a $N \times 9$ submatrix A_1 will leave residuals of the form

$$r = xx'E_1 + xy'E_2 + xE_3 + yx'E_4 + yy'E_5 + yE_6 \\ + x'E_7 + y'E_8 + E_9, \quad (13)$$

while those derived from (12), A_2 , will leave a residue of the form (note: since we are only doing weighting, we use the approximate equation (5) for simplicity)

$$e = (xu' + ux')E_1 + (uy' + xv')E_2 + uE_3 + (vx' + yu')E_4 \\ + (vy' + yv')E_5 + vE_6 + u'E_7 + v'E_8. \quad (14)$$

If we have some notion as to the correct essential matrix, then according to [11], [7], rows of submatrix A_1 should be multiplied by the factor w_1 , where $w_1 = 1/(\nabla r)$, $\nabla r = (r_x^2 + r_y^2 + r_{x'}^2 + r_{y'}^2)^{\frac{1}{2}}\sigma_{pp'}$ and $r_x = (x'E_1 + y'E_2 + E_3)$. (Note: We are assuming that $\sigma_x = \sigma_{x'} = \sigma_y = \sigma_{y'} = \sigma_{pp'}$).

Similarly, for rows of A_2 , the appropriate weight is $w_2 = 1/(\nabla e)$. If we assume that

$$(x\sigma_{\dot{p}\dot{p}'}^2, (y\sigma_{\dot{p}\dot{p}'}^2, (x'\sigma_{\dot{p}\dot{p}'}^2, (y'\sigma_{\dot{p}\dot{p}'}^2) \\ \gg (u\sigma_{pp'}^2, (v\sigma_{pp'}^2, (u'\sigma_{pp'}^2, (v'\sigma_{pp'}^2),$$

Then $\nabla e \approx (e_u^2 + e_v^2 + e_{u'}^2 + e_{v'}^2)^{\frac{1}{2}}\sigma_{\dot{p}\dot{p}'}$. We note that $e_u = r_x$, $e_{u'} = r_{x'}$, $e_v = r_y$, $e_{v'} = r_{y'}$.

Since the ratios of w_1 and w_2 are not dependent on E , we can compute the relative weights of rows A_1 and A_2 even though E is unknown.

$$w = \frac{w_2}{w_1} = \frac{\sigma_{pp'}}{\sigma_{\dot{p}\dot{p}'}} \quad (15)$$

If we assume a matching error with standard deviation 1 pixel, a flow with magnitude 1 pixel and standard deviation 0.1 pixel (10 percent error in the flow), then $w = 10$.

III. IMPLICATIONS FOR A MOVING STEREO RIG

For general scenes, TN does not make any significant difference to the stability of camera pose recovery. This is because even after weighting, the rows of A_2 , are still 7 to 10 times smaller than the rows of A_1 (This is because on the average, the flow magnitude is 100 times smaller than the magnitude of correspondence points. Hence, even after re-weighting with TN, the flow is still considerably smaller). This is because the noise level present in the optical flow is significantly higher than that in point matches. If however, the point motion can be accurately recovered, then the gains obtained by TN can be significant. The other circumstance occurs when the information present from the motion is crucial for disambiguating the scene.

A. The moving stereo rig viewing a planar scene

It is a well established fact that a pair of stereo images of a plane will result in a degenerate configuration from which it is impossible to recover the essential matrix. However, if one has a moving stereo rig, then this need not be so.

The intuitive reasoning behind this phenomenon is as follows. If the stereo rig takes images of a plane, then moves forward and takes a second set of images of the same plane,

the combined set of matched points can be considered as coming from two planes, one in front of the other. Therefore, in theory a planar configuration is not degenerate when viewed by a moving stereo rig. A small motion will mean that the 'two planes' are very close to each other and the scene is effectively still planar. We show how TN can give us a better understanding of the problem.

In a planar scene, there exists a homography H such that for the matched points p and p' , $p = Hp'$. During essential matrix recovery, a planar scene is a degenerate case because SH is always a possible solution to E , S being any skew symmetric matrix. However, in general, SH will not satisfy equation (11). As such, in general, the calibration of a moving stereo rig is not degenerate when faced with a planar scene.

This phenomenon can also be explained in the discrete domain. If the stereo rig were to make a small motion, the essential matrix given by SH when applied to equation (4) will leave a residue given by

$$r_p = \Delta p^T SH p' + p^T SH \Delta p' + \Delta p^T SH \Delta p'. \quad (16)$$

Since Δp and $\Delta p'$ are small, the difference between the residue left by SH and that left by E will be small. In a noisy situation, we might conclude that the planar degeneracy remains. If however we make use of TN and substitute (4) with a scaled up equation (12), the residue that results from replacing E with SH will be significantly increased. This means that TN can allow us to effectively disambiguate a scene that is traditionally considered ambiguous.

Planar or near planar scenes abound in our everyday environment. The degeneracy of a self calibrating stereo rig to such scenes is a major limitation. This is especially so since even if one knows that the scene in view is planar, there is no way to recover camera pose from the said scene [6]. The algorithm described above means that if we use TN, we can potentially build a moving, self-calibrating stereo rig to whom planar scenes are no longer degenerate (it can treat planar scenes just like any other general scene).

IV. EXPERIMENTS

a) *Simulation Results:* The results are shown in Figure 2 to Figure 6. Unless otherwise stated, the simulation conditions are as follows.

- Stereo rig of baseline 1 unit was moving forward while viewing a scene that consists of a plane with normal $[0 \ 0 \ 1]^T$. On average, points were 10 units away. (Performance varies little with plane orientation.)
- The cameras had field of view of 45 degrees and saw 50 matched points that were an average of 10 units away.
- Isotropic noise was added to both images. The standard deviation of the noise in the point matches is 0.5 pixel standard deviation in a 400×400 image. Flow noise was at 7% of flow magnitude.
- Error refers to the average error from 80 iterations.

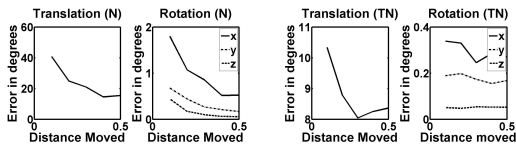


Fig. 2: Without (TN), stability to planar scenes is highly dependent on the distance moved. Errors in rotation are shown in its x:y:z components

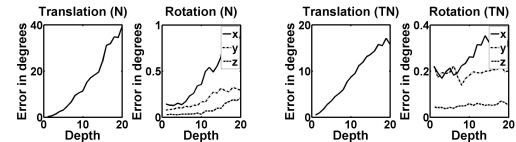


Fig. 3: Performance is highly depth-dependent. This is because both algorithms are degenerate to a plane at infinity. Amount moved by the stereo rig is set at the very large value of 0.5 units to allow better results with the non time normalized data.

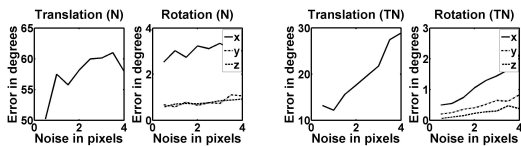


Fig. 4: Stability of the algorithm with increasing noise. Amount moved by rig is 0.1 units.

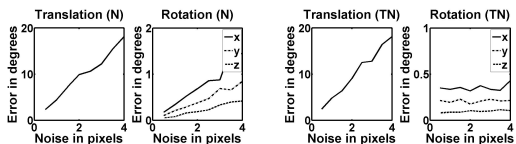


Fig. 5: Performance when the scene is not planar.

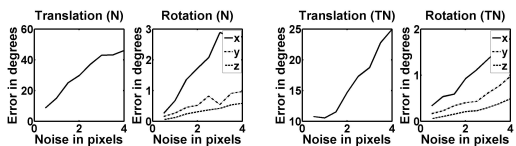


Fig. 6: Depth variation is 20% of average depth. This shows that (TN) also benefits near planar scenes

b) Real Results: The results for Figures 7(a) – (e) (labeled from left to right) are presented in Table 1.

- For the scene in Fig 7(b), the translational separation of the cameras is 1m in the Y-direction and there is no relative rotation. For all other scenes, the translational separation of the cameras is 40cm in the Y-direction and the relative rotation is 14 degrees about the X-axis.
- SIFT was used to obtain flow and matched points.
- The error for each scene is the average error from 3 successive trials (as RANSAC produces slightly different results each run). "Mean Rotation Error" is the mean rotation errors in each of the 3 directions.
- DB is the ratio of the average distance to the baseline.

From both simulation and real results, TN considerably improves performance. Without TN, it can disambiguate a planar scene only if the plane is very close to the stereo system.

Error in degrees	(a)	(b)	(c)	(d)	(e)
Mean Rotation Error (TN)	2.8	2.8	2.8	2.7	2.4
Translation Error (TN)	3.1	7.9	3.7	6.7	5.3
Mean Rotation Error (N)	2.1	2.2	3.0	2.6	2.5
Translation Error (N)	20.3	18.6	20.1	7.5	9.8

TABLE I: Results for each scene after applying the eight point algorithm to both TN and non-TN data.

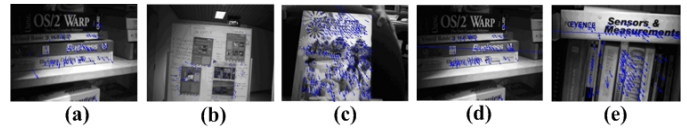


Fig. 7: (a) Resolution 240×320 , $DB \approx 2$, Planar; (b) Resolution 480×640 , $DB \approx 5$, Planar; (c) Resolution 240×320 , $DB \approx 6$, Planar; (d) Resolution 480×640 , $DB \approx 2$, Planar; (e) Resolution 480×640 , $DB \approx 2$, Non-Planar.

V. CONCLUSION

We have presented a generic means of extending discrete algorithms to the differential domain. To demonstrate the value of this concept, we show how this formulation can be used to solve non-trivial traditional problems in computer vision. Other than its generality, our formulation has the additional advantage of not requiring the differential approximation that is the bane of so many differential algorithms. This enables it to handle both large and small motions. We have shown how TN allows us to see and utilise information that was poorly represented in the discrete formulation, thus making a previously intractable problem reasonably well-posed.

REFERENCES

- [1] F. Dornaika and R. Chung. Stereo correspondence from motion correspondence. *Proc. of Computer Vision and Pattern Recognition*, 1999.
- [2] A. Heyden. Differential-algebraic multiview constraints. *IEEE Int. Conf. Pattern Recognition*, 2006.
- [3] H. Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. *Nature*, pages 133–135, 1981.
- [4] K. Åström and A. Heyden. Multilinear constraints in the infinitesimal-time case. *Proc. of Computer Vision and Pattern Recognition*, 1996.
- [5] K. Åström and A. Heyden. Continuous time matching constraints for image streams. *Int'l Journal of Computer Vision*, pages 85–86, 1998.
- [6] P. Torr, A. Fitzgibbon, and A. Zisserman. The problem of degeneracy in structure and motion recovery from uncalibrated image sequences. *Int'l Journal of Computer Vision*, 32(1):27–44, 1999.
- [7] P. Torr and D. Murray. The development and comparison of robust methods for estimating the fundamental matrix. *Int'l Journal of Computer Vision*, 24(3):271–300, 1997.
- [8] B. Triggs. Differential matching constraints. *Int'l Journal of Computer Vision*, pages 370–376, 1999.
- [9] T. Vieville and O. Faugeras. Motion analysis with a camera with unknown and possibly varying intrinsic parameters. *Int'l Journal of Computer Vision*, pages 750–756, 1995.
- [10] T. Vieville and O. Faugeras. The first order expansion of motion equations in the uncalibrated case. *Computer Vision and Image Understanding*, 64(1):128–146, 1996.
- [11] J. Wolberg. Prediction analysis. *D. Van Nostrand Company*, pages 133–135, 1967.
- [12] Z. Zhang. Motion and structure of four points from one motion of a stereo rig with unknown extrinsic parameters. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 17(12):1222–1227, 1995.