Iterative Computation of 3D Plane Parameters

Haris Baltzakis and Panos Trahanias

Institute of Computer Science
Foundation for Research and Technology-Hellas
PO Box 1385, 71110, Heraklion, Crete, Greece
and
Department of Computer Science
University of Crete
PO Box 2208, 71409, Heraklion, Crete, Greece

{xmpalt,trahania}@ics.forth.gr

Abstract

Knowledge of the position and orientation of 3D planes that exist in a scene is very important for many machine vision-based tasks, such as navigation, self-localization and docking. In this paper we present an iterative method for robustly computing the plane parameters out of given image sequences. The proposed method does not assume knowledge of point correspondences across the image frames, relying only on raw image data. Its application is, therefore, independent of the ability to reliably compute point correspondences. The competence to determine unknown parameters of the 3D-camera motion is also inherent to the method.

1 Introduction

Most vision-based navigation tasks usually rely on certain scene (environmental) features, which provide adequate information for the task at hand. Planar regions are important aspects in indoor workspaces and are frequently employed in machine vision problems. They constitute very common structures in man made environments and play a crucial role when dealing with certain tasks, such as indoor navigation, self-localization and docking. Applications aiming to derive and use plane parameters in order to achieve such tasks usually involve two distinct phases: (a) scene segmentation to extract planar areas and, (b) estimation of plane parameters for the areas of interest. Projective invariants [5,6] provide a means to tackle the former phase, while, for the latter phase, depending on the actual task at hands, various approaches have been proposed to obtain analytical solutions of the plane unknowns [2,3,4].

BMVC 2000 doi:10.5244/C.14.48

The estimation of 3D plane parameters is a specific case of the reconstruction problem. This is due to the fact that knowledge of the plane position and orientation provides enough information to determine the 3D coordinates of each point on it. On the other hand, a solution to the reconstruction problem (estimation of 3D point coordinates) provides also the exact plane parameters through the application of best-fit algorithms.

The pursuit of any task regarding recovery of 3D information usually involves, as a first, unavoidable step, the establishment of point correspondences. Employment of optic flow [7,8,9] for establishing large sets of low confidence correspondences is a common practice. However, errors are usually introduced in the computation of point correspondences, which are propagated in the next stages affecting the accuracy of computations. Additionally, for utilizing established point correspondences in order to compute 3D space coordinates, camera motion must also be known. The problem of extracting camera motion has often been addressed in the past [10,11,12,13,14] either as a stand-alone problem, or, as an integrated step in 3D reconstruction approaches.

In this paper we present a novel, iterative method for the estimation of 3D plane parameters, based on the minimization of a cost function. The proposed method, unlikely most known approaches, does not rely on given point correspondences; rather, it starts from either assumed or estimated motion and plane parameters, which are used to analytically derive point correspondences. The inferred correspondences are evaluated by mapping them on the original images and the resulting error percentage constitutes the error function that is utilized by an iterative algorithm to refine the initial estimates. The iterative nature of the proposed method permits the derivation of the camera motion in addition to the plane parameters. This is facilitated by including in the error (cost) function two terms: the cost due to false estimation of the plane parameters and the cost due to false estimation of the camera motion.

While relying on known equations for the computation of 3D coordinates and camera motion parameters, the innovative formulation of the proposed method as an iterative approach furnishes it with enhanced performance and robustness, even in cases of limited a-priori information. Moreover, this formulation completely eliminates the need for a-priori establishing point correspondences across the image frames.

An inherent constraint in methods trying to resolve both camera motion and plane parameters is that translation magnitude and plane distance cannot be simultaneously determined [1]. This, stems from the fact that optical stimuli received from planes lying far from a fast moving observer (camera) are equivalent to stimuli received from planes laying closer to a slower moving observer. A-priori knowledge about either the plane distance or the camera translation magnitude is necessary in order to overcome this ambiguity and obtain unique solutions. In practical applications such knowledge may be inferred from other sources, such as the motion encoders of a mobile platform or given 3D measurements of the workspace. When the magnitude of the translational motion is not available, plane distance is only inferred relatively to this unknown magnitude.

A significant advantage is the ability to utilize more than one pair of frames for the sake of robustness. This is extremely meaningful in cases that the additional frames introduce no extra, unknown parameters (e.g. when the camera motion is constant). The proposed method was tested extensively and yielded very promising results in the cases of real and artificial indoor sequences. In order to evaluate the results, a comparative test against the well-known method of the flow matrix [1] is also presented.

2 3D Plane Estimation

2.1 Mathematical Preliminaries

Let c$_0$ be a camera and let $O^{t_o}x^{t_o}y^{t_o}z^{t_o}$ be its orthogonal coordinate system defined by the optical center $O^{t_0}=(O_x^{t_0},O_y^{t_0},O_z^{t_0})$ and the unit vectors $\bar{x}^{t_0}$, $\bar{y}^{t_0}$, $\bar{z}^{t_0}$. Let c$_1$ be a second camera with optical center and unit vectors, defined with respect to the coordinate system of the first camera, $O^{t_1}=(O_x^{t_1},O_y^{t_1},O_z^{t_1})$ and $\bar{x}^{t_1}$, $\bar{y}^{t_1}$, $\bar{z}^{t_1}$, respectively. Let also P be a space point lying on a plane, with $(X^{t_0},Y^{t_0},Z^{t_0})$, (X^{t},Y^{t},Z^{t}) its coordinates with respect to the two cameras, c$_0$ and c$_1$.

If $R^{t_0\rightarrow t_1}$ is the rotation matrix (the matrix having as columns the unit vectors $\bar{x}^{t_1}$, $\bar{y}^{t_1}$, $\bar{z}^{t_1}$), it can be shown [1] that:

$$\begin{bmatrix} X^{t_1} \\ Y^{t_1} \\ Z^{t_1} \end{bmatrix} = -O^{t_1} + R^{t_0\rightarrow t_1}\begin{bmatrix} X^{t_0} \\ Y^{t_0} \\ Z^{t_0} \end{bmatrix} \tag{1}$$

which expands to,

$$\begin{bmatrix} X^{t_1} \\ Y^{t_1} \\ Z^{t_1} \end{bmatrix} = -\begin{bmatrix} O_x^{t_1} \\ O_y^{t_1} \\ O_z^{t_1} \end{bmatrix} + \begin{bmatrix} \bar{x}_x^{t_1} & \bar{y}_x^{t_1} & \bar{z}_x^{t_1} \\ \bar{x}_y^{t_1} & \bar{y}_y^{t_1} & \bar{z}_y^{t_1} \\ \bar{x}_z^{t_1} & \bar{y}_z^{t_1} & \bar{z}_z^{t_1} \end{bmatrix}\begin{bmatrix} X^{t_0} \\ Y^{t_0} \\ Z^{t_0} \end{bmatrix} \tag{2}$$

Eq. (2) associates the 3D coordinates of point P with respect to c$_1$ with the coordinates of the same point as seen from c$_0$.

The projections of point P on the image planes of the two cameras are given as

$$(x^{t_o},y^{t_0}) = (f\frac{X^{t_0}}{Z^{t_0}},f\frac{Y^{t_0}}{Z^{t_0}}) \tag{3}$$

$$(x^{t_1},y^{t_1}) = (f\frac{X^{t_1}}{Z^{t_1}},f\frac{Y^{t_1}}{Z^{t_1}}) \tag{4}$$

where f is the camera focal length.

2.2 Method Formulation

Each point P laying on a plane, will satisfy a plane equation, that, with respect to the coordinate system of c$_0$, has the general form of:

$$aX^{t_0} + bY^{t_0} + cZ^{t_0} + d = 0, \quad \|a,b,c\| = 1 \tag{5}$$

where, (a, b, c) denotes the plane normal and d the distance from the origin (optical center of c$_0$). By substituting on the above plane equation 3D coordinates with corresponding 2D projection coordinates from eq. (3), we obtain:

$$ax^{t_0}\frac{Z^{t_0}}{f}+bx^{t_0}\frac{Z^{t_0}}{f}+cZ^{t_0}+d=0 \tag{6}$$

By solving eq. (6) for Z^{t_0}, we obtain

$$Z^{t_0}=\frac{df}{ax^{t_0}+bx^{t_0}+cf} \tag{7}$$

The above value for Z^{t_0} can be substituted in eq. (1) resulting in :

$$\begin{bmatrix} X^{t_1} \\ Y^{t_1} \\ Z^{t_1} \end{bmatrix} = -\begin{bmatrix} O_x^{t_1} \\ O_y^{t_1} \\ O_z^{t_1} \end{bmatrix} - \frac{1}{ax^{t_0}+by^{t_0}+cf} R^{t_0 \to t_1} \begin{bmatrix} x^{t_0} \\ y^{t_0} \\ d \end{bmatrix} \tag{8}$$

Moreover, expanding the rotation matrix in elements (eq. 2) and combining the left part of eq. (8) with the equations for the projective transformation that takes place in c$_1$ (eq. 4), we obtain:

$$x^{t_1}=\frac{O_x^{t_1}+\dfrac{x^{t_0}\overline{x}_x^{t_1}+y^{t_0}\overline{y}_x^{t_1}+d\overline{z}_x^{t_1}}{ax^{t_0}+by^{t_0}+cf}}{O_z^{t_1}+\dfrac{x^{t_0}\overline{x}_z^{t_1}+y^{t_0}\overline{y}_z^{t_1}+d\overline{z}_z^{t_1}}{ax^{t_0}+by^{t_0}+cf}} \tag{9}$$

and

$$y^{t_1}=\frac{O_y^{t_1}+\dfrac{x^{t_0}\overline{x}_y^{t_1}+y^{t_0}\overline{y}_y^{t_1}+d\overline{z}_y^{t_1}}{ax^{t_0}+by^{t_0}+cf}}{O_z^{t_1}+\dfrac{x^{t_0}\overline{x}_z^{t_1}+y^{t_0}\overline{y}_z^{t_1}+d\overline{z}_z^{t_1}}{ax^{t_0}+by^{t_0}+cf}} \tag{10}$$

Equations (9) and (10) indicate that knowledge of the relative positions of the two cameras and the plane parameters provides enough information to analytically compute point correspondences for each plane point projected on the image planes.

It is a straightforward task to test the accuracy of calculated point correspondences, just by mapping each couple of implied image coordinates on the original images and checking whether their respective intensity values match. Since the accuracy of the calculated image points is heavily depended on the accuracy of the assumptions made about the camera motion and the plane equation, the above technique provides a means for easily evaluating the correctness of all parameters involved. Assumptions far away from the correct ones give rise to erroneous point correspondences and, hence, lead to very small number of verified point matches. As the assumptions become better, the number of correctly matched points will eventually increase.

This inference forms the basis for devising an iterative algorithm for computing the plane parameters and the camera motion:

Iterative Algorithm

Computation of 3D plane parameters and camera motion parameters.

```
Step i.   Initialisation of the (unknown) plane parameters $\mathbf{a},\mathbf{b},\mathbf{c},\mathbf{d}$, and the camera
          motion parameters. Initialisation of epoch number to zero.
Step ii.  Increase epoch number.
Step iii. For each of the unknown parameters repeat:
          • Calculate point correspondences using eqs. (9) and (10).
          • Evaluate the point correspondences by backprojecting them on the image plane
            and matching them with the original image data. Extract the error as the
            percentage of unmatched points.
          • Change the current parameter towards the direction that minimizes the error.
Step iv.  If no parameters have changed in step iii then exit. Otherwise, goto step ii.
```

Point matching, needed in step iii of the algorithm, is performed by a correlation procedure in a small neighbourhood. For implementing the iterative algorithm a simulated annealing like scheme was chosen. While eliminating the need for computation of error-space gradients, its probabilistic nature ensures convergence even in cases that local error minima may be reached during the iterative process. This has also been verified through extensive experimentation; in all cases the algorithm was able to rapidly converge to the correct set of parameters (see also next section).

When more than two image frames are available, slight modifications to the above algorithm can be made in order to utilize the extra information. This is extremely meaningful in cases that the additional frames introduce no extra, unknown parameters (e.g. when the camera motion is constant).

3 Experimental Results

The proposed method was tested extensively with artificial and real image sequences. Sample results from these experiments are presented here that demonstrate the effectiveness of the proposed method. As mentioned earlier, there is always an inherent ambiguity between the camera translation magnitude and the distance of the plane. To cope with that, available ground truth for the camera translation magnitude was employed in our experiments.

3.1 Artificial Data

Figure 1 shows two consecutive frames of an artificial sequence of a textured wall that has been employed in our experiments; ground truth is available for this sequence and hence it can be used for experimental verification. The camera is initially placed at a distance of 250 units from the wall and faces it with an angle of 45^o. The camera motion consists of a translation of 10 units towards the direction of its optical axis (z-axis) and a rotation of 1^o around its vertical axis (y-axis).

Figure 2 shows the estimation of the plane and motion parameters during the iterative process. No a-priori knowledge was used to initialise the algorithm (the initial estimates were chosen randomly). As can be observed, the algorithm succeeded in reaching the correct values in less than twenty iterations.

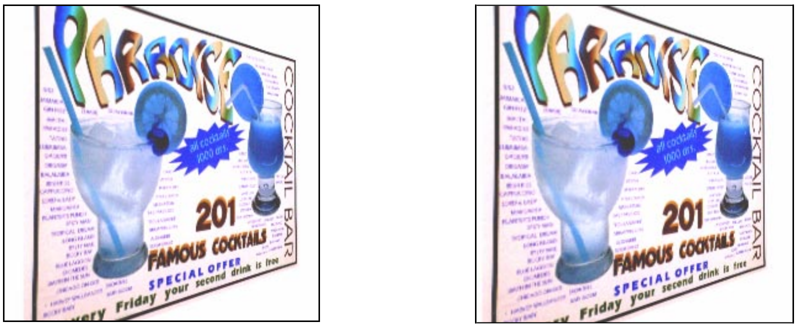


Figure 1. Two frames of an artificial sequence used to demonstrate the operation of the proposed method.

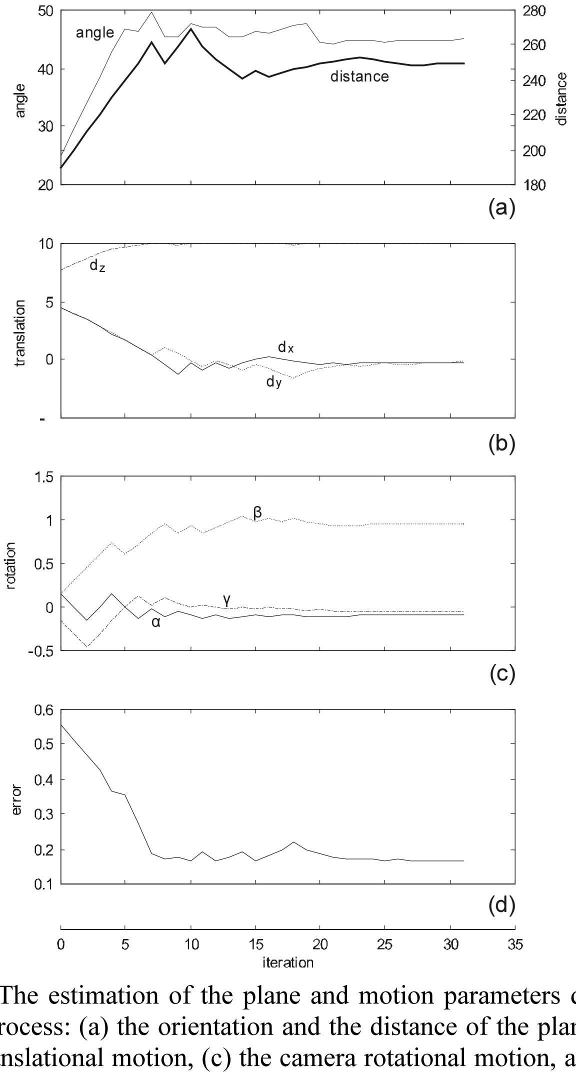
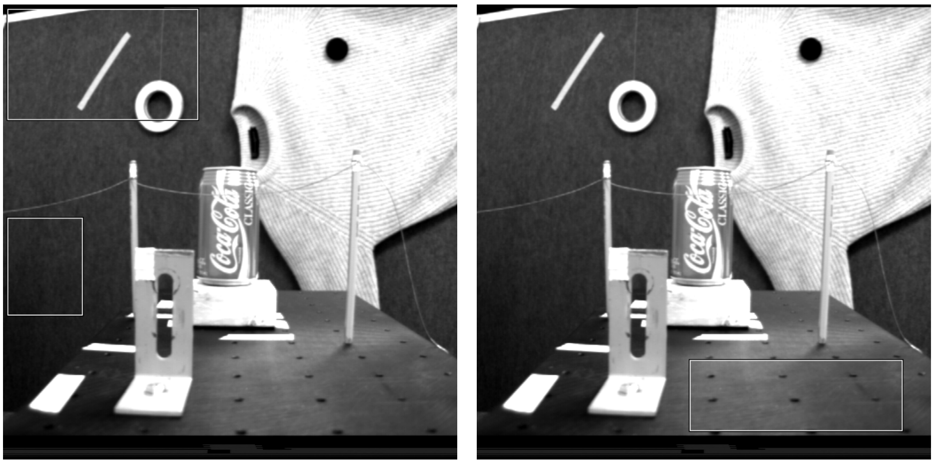


Figure 2. The estimation of the plane and motion parameters during the iterative process: (a) the orientation and the distance of the plane, (b) the camera translational motion, (c) the camera rotational motion, and (d) the error function. Distances are expressed in world units. Angles and Orientations are expressed in degrees.

3.2 Real Data

For evaluating the performance of the proposed method with real data, we used four frames of the well-known coca-cola sequence (Figure 3). In this sequence the camera is translating with a constant velocity in the direction of its optical axis. Ground truth concerning both the motion and the camera parameters is available, but not for the imaged scene (depth of image points).

Two different planes appearing in the sequence where chosen: (a) the plane defined by the background wall, and (b) the plane defined by the table. These planes were modeled in our experiments by constraining the algorithms to rectangular areas, manually placed on the imaged scene. The plane of the background wall was modeled by two rectangles, as shown in the left image (case a) of Figure 3. The plane of the table was modeled by one rectangle, as shown in the right image (case b) of Figure 3. Starting from random initial estimates, the proposed method quickly converged to values consistent with the ground truth provided (in case of the camera motion) and our visual perception (in case of the plane parameters). The results are summarized in Table 1.



Case a. The area of the background wall was modeled using two rectangular areas in the left side of the imaged scene.

Case b. The surface of the table was modeled using a rectangular area in the bottom-right of the imaged scene.

Figure 3. The search areas for applying both methods in the coca-cola sequence.

Table 1. Results for the coca-cola sequence.

	case a	case b
Plane orientation	perpendicular to the camera z-axis	Perpendicular to the camera y-axis
Plane distance	≈796 pixels	≈46 pixels

In addition to the above sequence, results from a real sequence (Figure 4), acquired in a corridor outside our laboratory, are provided. In this experiment the camera axis was forming an angle of 45^o with the wall while translating in a direction parallel to it. Despite the lack of accurate knowledge of the camera parameters, the results were quite accurate and are illustrated in Table 2

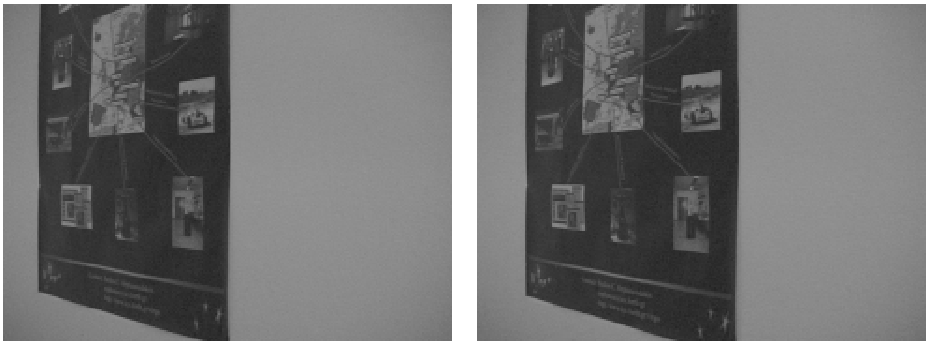


Figure 4. Two frames of a real scene depicting a wall, while the camera is translating in a direction parallel to the wall.

Table 2. Results for the wall sequence.

Plane orientation	≈ 45^o
Plane distance	≈ 85 cm / 110 pixels per cm
Camera Motion	≈ 10 cm translation parallel to the wall

3.3 Comparative test

Finally, a comparative test with the well-known method of the flow matrix [1] was performed. The framesets used were artificial, depicting the wall shown in Figure 1 from various positions and directions and with varying camera motion. The results are summarized in Table 3.

Table 3. Comparative results for plane detection methods.

Real plane parameters	iterative method	Flow matrix method
a=0.71	a=0.72	a=0.73
b=0.00	b=0.03	b=0.01
c=0.71	c=0.69	c=0.68
d=250	d=253	d=246
translation : (0,0,3)		
rotation : none		
a=0.71	a=0.70	a=0.71
b=0.00	b=0.10	b=0.10
c=0.71	c=0.71	c=0.71
d=250	d=242	d=235
translation : (0,0,3)		
rotation : (0°,2°,0°)		
a=0.71	a=0.71	a=0.75
b=0.00	b=0.10	b=0.12
c=0.71	c=0.71	c=0.65
d=250	d=249	d=137
translation : (0,0,3)		
rotation : (0°,10°,0°)		
a=0.71	a=0.73	a=0.81
b=0.00	b=0.16	b=0.10
c=0.71	c=0.68	c=0.57
d=250	d=252	d=52
translation : (0,0,10)		
rotation : (0°,20°,0°)		

The results in Table 3 are only indicative of the ability of our method to converge to correct parameters. The accuracy of the iterative algorithm can, in most cases, be tuned to be more accurate by choosing appropriate thresholds, or, just by letting the algorithm perform more iterations. As can be observed from the above table, both methods performed well in the cases where the magnitude of the camera motion was relatively small. As the camera motion, and especially rotation, was growing larger, the flow matrix method failed to give accurate results. This failure of the flow matrix method may be explained by the fact that it employs the simplified equations giving instant motion estimations, instead of the full rotation matrix expressions employed by the proposed method. Moreover, the iterative nature of our method facilitates converge to the actual parameter values, even in cases where the initial estimates are rather inaccurate.

4 Discussion

In this paper a new, iterative method for plane parameter estimation has been proposed. The method starts from known relations describing the camera motion and induced motion on the image plane and presents a novel formulation as an iterative procedure that results in an effective computational implementation. The proposed method was tested extensively and yielded very good results for both artificial and real image sequences.

According to the proposed method, point correspondences are analytically computed from estimated plane and motion parameters. This approach deviates from existing ones, in that, the problem is assumed completely reversed, starting from what, usually, is what we seek for. Hence, no a-priori point correspondences are required and the method can be applied directly to raw image data.

The iterative - simulated annealing like - nature of the method furnishes it with the ability to achieve convergence even if the initial assumptions are far from the correct values, and, at the same time, to provide robustness not easily seen in straightforward, equation-reversion methods. The experimental results obtained have demonstrated that iterative approaches are applicable for plane parameter estimation. In expense of simplicity, more sophisticated iterative algorithms could be designed in order to achieve even faster convergence.

References

1. Kanatani, K. Geometric Computation for Machine Vision, Oxford University Press. Yew York, 1995.
2. Dev, A., Krose, B., Groen, F. Navigation of a mobile robot on the temporal development of the optic flow. Proceedings IROS'97, IEEE , pp. 558-563. 1997.
3. Santos-Victor, J., Sandini, G., Gurotto, F., Garibaldi, S. Divergent stereo in autonomous navigation. From bees to robots. Int. J. Comp.. Vision, 14, 159-177, 1995.
4. Sipola, R., Buxton, H. Surface orientation and time to contact from image divergence and deformation. Proc. 2nd Europ. Conf. Computer. Vision., Ligure, Italy, pp. 187-202. 1992.
5. Sinclair D., Blake, A. Quantitive planar region detection. International Journal of Computer Vision 18 (1), 77-91. 1996.
6. Kanatani, K. Computational cross ratio for computer vision. CVGIP - Image Understunding vol. 60, no. 3, 371-381. 1994.
7. Baron, J. L., Fleet, D. J., Beauchemin, S. S. Performance of optical flow techniques. International Journal of Computer Vision, 12:1, 43-77. 1994.
8. Buxton, B., Blake, A. Computation of optic flow from the motion of edge features in image sequences. Image Vis. Comput. 2:59-74.1984.
9. Simoncelli, E.P., Adelson, E.B., Heeger, D.J. Probability distributions of optical flow. IEEE Conf. Computer Vision and Pattern Recognition, Mauii, Hawaii, 1991.
10. Longuet-Higgins H.C. A computer algorithm for reconstructing a scene from two projections. Nature 293 (10), pp 133-135. 1981.
11. Brouce, A.R., Horn, B.K., Passive Navigation. Computer Graphics and Image Processing, 21, pp. 3-20. 1983.
12. Negahdaripour, S., Lee, S. Motion recovery from image sequences using only first order optic flow information. Int. J. Computer Vision, 9 (3), 163-184. 1992.
13. Huang, T.S., Netravali, A. Motion and Structure from Feature Correspondences: A review. Proceedings of the IEEE, vol. 82, no. 2. February 1994.
14. Wend, J., Huang, T.S. Nerendra Ahuja. Motion and Structure from Two Perspective Views: Algorithm, Error Analysis, and Error Estimation. IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 11, no. 5. May 1989