ESTIMATION OF MOTION FROM A SEQUENCE OF IMAGES USING SPHERICAL PROJECTIVE GEOMETRY

Madasu Hanmandlu
FOE, Multimedia University
Jalan Multimedia, 63100
Cyberjaya, Selangor D.E.
Malaysia
masasu@mmu.edu.my

Shantaram Vasikarla
Information Technology Dept.
American InterCon. University
12655 W. Jefferson Blvd.
Los Angeles, CA 90066, U.S.A.
svasikarla@la.aiuniv.edu

Vamsi Krishna Madasu
School of IT and EE
University of Queensland
Brisbane, QLD 4072
Australia
s40451381@student.uq.edu.au

Abstract: Motion is an important cue for many applications. Here we propose a solution for estimating motion from a sequence of images using three algorithms, viz., Batch, Recursive and Bootstrap methods. The motion derived using spherical projection relates the image motion to the object motion. This equation is reformulated into a dynamical space state model, for which Kalman filter can be easily applied to yield the estimate of depth. We also propose a new approach for establishing correspondences using local planar invariants and hierarchical groupings. The proposed algorithm provides a simple yet robust method having lower time complexity and less ambiguity in matching than its predecessors.

Keywords: Edge rings, Correspondence, Spherical Projection, Estimation, Kalman Filter.

1. Introduction

An object in motion can provide a vivid description of its structure. From a sequence of images, it is possible to track the object in motion and estimate the motion and structure parameters. In general, a motion analysis system has three tasks, namely, feature extraction, feature matching and motion/structure computation. While analyzing images, reliable tokens such as points, lines, corners and curves are detected. Feature extraction, thus, is the first step in motion analysis. For the detection of corners and edges, we use SUSAN, which stands for ‘Smallest Unvalue Segment Assimilating Nucleus’ [1]. It is a new approach for edge detection (1-D feature), corner detection (2-D feature) and structure preserving noise reduction. Most researchers use the Canny edge detector [2] and Plessey corner detector for feature extraction. SUSAN has been proven to be much faster than these standard approaches. The matching algorithm proposed by us uses local planar invariants [3] and incorporates the characteristics of hierarchical groupings for reducing time complexity. The key idea of the algorithm is to assume that apparent motion between two images can be approximated by planar geometric transformations like similarity or affine transforms. In order to simplify the formulation, rigid body motion and spherical formulation have been assumed. The image motion equation resulting from the differentiation of spherical projection equation is the starting step for the formulation of motion equation. An optimal motion estimation technique is presented by Weng et al. [4]. The paper discusses a type of motion for which algorithms based on the epipolar constraints are very sensitive to noise. Ali Azerbayejani and Pentland [5] have presented feature based recursive estimator that uses Extended Kalman Filter (EKF) for recovery of motion, pointwise structure and focal length from arbitrarily long sequences. This formulation applies to general perspective projection, including the special case of orthographic projection. In this paper, we incorporate the use of EKF for spherical projection. The idea of using spherical projection has been inspired by the work of Yeng and Huang [6], who proposed spherical projection as a fundamental tool in determining of 3-D motion and structure of a rigid body from image sequences. Depth estimated using two frames is bound to be sensitive to noise, that is why the Kalman filter is increasingly being used for the estimation of depth from a sequence of images [18]. The organization of this paper is as follows: Section 2 discusses briefly the algorithm proposed by us for establishing correspondences. In section 3, we discuss the spherical projection and go on with the mathematical formulation of motion equations. In section 4 the three techniques for estimation of motion, namely, Batch Processing, Recursive Estimation and Bootstrap approach are discussed. The results and future scope of our work are discussed in sections 5 and 6 respectively.

2. Correspondence of features

The method adapted by us uses local planar invariants [3]. The key idea of the algorithm is to assume that apparent motion between two images can be approximated by planar geometric
transformations like similarity or affine transforms. Under such an assumption local planar invariants related to the kind of transformations used as approximations should have same value in both the images. A global constraint is added to ensure a global coherency between all possible matches. All the local matches must define approximately the same geometric transformation between the two images. If there are more than two images, the matching is done pairwise and in the second step global matches are deduced. The algorithm in [10] uses geometric hashing technique, which suffers from a high level of complexity for large, images (or images having a large number of features). We have attempted to resolve this problem by grouping features hierarchically. However, we do not implement hierarchical matching but incorporate characteristics of hierarchical groupings described in [13]. Moreover, the constraints used by us are different. The Quasi-invariants used by us work well for noisy images and for uncalibrated camera unlike the Epipolar constraints of [14].

2.1 Matching Algorithm

We define two types of basic features: V-sections and Y-sections, which consist of two and three lines respectively, having a common end point. At the next level of hierarchy, edge rings, which consist of V and Y sections, are defined. These features usually do not have the same pixel coordinates in the two images. The coordinate difference is called Apparent Motion. Apparent motion is not a 2-D geometric transformation. However, for generic views of nearly coplanar objects, it is well approximated by 2-D transformation. The steps in the algorithm are:

- Matching starts at the edge ring level. The edge rings are said to be matched if the number of Y or V sections is at least 75% of each other.
- Once root node matches are initialized by the above method, either similarity or affine approximation is chosen. Local invariants are calculated for salient feature configurations: angle and length ratios for similarities and affine coordinates for affine transformations.
- These invariants of two images are matched according to the thresholds derived experimentally (from the noise level).
- When invariants of two configurations match, affine transformation between the configurations is computed.
- Transformations are represented as points in parameter space: four points for similarities (two translations, one rotation and one scaling factor) and six parameters for affine transformations (four parameters for linear part and two for translation).
- Correct matches define transformations close to the best approximation to the apparent motion.
- Invariant matches give rise to configuration matches and feature matches are deduced from these. In case of ambiguities, only the most probable matches are considered correct.

The main limitations of the algorithm come from the approximation of apparent motion.

3. Spherical Projection

Spherical projection is proposed as a fundamental tool in determining 3-D motion and structure of a rigid body from an image sequence. Points on the image plane are represented by their central projections on a unit sphere. The central projection of a world point \( P \) on the unit sphere \( P_c \) given the projection \( P_t \) and world point \( P \) on the image plane, \( P_c \) is found from equation below [14]:

\[
\begin{align*}
\frac{P_c}{||P_t||} = P_t'
\end{align*}
\]

where,

\[
\begin{bmatrix}
 x \\
 y \\
 z \\
 F
\end{bmatrix} = \begin{bmatrix}
 X_t \\
 Y_t \\
 F
\end{bmatrix}, \quad \begin{bmatrix}
 x_c \\
 y_c \\
 z_c
\end{bmatrix}
\]

\((x, y, z), (X_t, Y_t, F)\) and \((x_c, y_c, z_c)\) are the coordinates of the point in world pace, image plane and on the unit sphere respectively. Therefore, from time \( t \) to \( t' \), \( P \) moves to its corresponding point \( P' \) under the rigid motion. The pair \((P_c, P_t')\) on the unit sphere corresponds to the pair \((P_t, P_t')\) on the image plane.

3.1 Mathematical Formulation

Let \( P \) be a world point with position vector \( r(s, t) \) in an absolute reference frame. The observer (or camera) is assumed to be placed at the center of the unit sphere, i.e., observer is at \( v(t) \). \( P_c \) is the projection of the point \( P \) on the unit sphere, \( q(s, t) \) is the unit vector along \( P_cP \), \( \lambda \) is the depth of the point \( P \) to \( P_c = P_cP \). Then \( r(s, t) \) is defined by the equation:

\[
r(s, t) = v(t) + \lambda Q(s, t)
\]

The motion equation derived in [14] is used here.

\[
q_t = S^2(q)(u/\lambda) + S(q)\Omega
\]
Descretizing this with $\Delta t = 1$, 

\[ q_{k+1} = q_k + S^2(q)u/\lambda + S(q)\Omega \] 

(3.4)

\[ \Delta q_k = q_{k+1} - q_k = S^2(q)u/\lambda + S(q)\Omega \] 

(3.5)

where, $q_k$ is the value of $q$ at $k^{th}$ instant.

4. Estimation of motion

Now, we propose three methods to solve this equation in the following section.

4.1 Batch Processing

From equation (3.5),

\[ \Delta q_k = \begin{bmatrix} S^2(q) & S(q) \end{bmatrix} \begin{bmatrix} u/\lambda \\ \Omega \end{bmatrix} \] 

(4.1)

Since, the rightmost matrix is a vector of order 6 X 1 (three components of each $u/\lambda$ and $\Omega$) and hence 6 unknowns, we need six equations. Therefore, we consider six points from each of two frames occurring at the $k^{th}$ and $(k+1)^{th}$ instants.

\[
\begin{bmatrix}
\Delta q_1 \\
\Delta q_2 \\
\Delta q_3 \\
\Delta q_4 \\
\Delta q_5 \\
\Delta q_6
\end{bmatrix} =
\begin{bmatrix}
S^2(q^1) & S(q^1) \\
S^2(q^2) & S(q^2) \\
S^2(q^3) & S(q^3) \\
S^2(q^4) & S(q^4) \\
S^2(q^5) & S(q^5) \\
S^2(q^6) & S(q^6)
\end{bmatrix}
\begin{bmatrix}
u/\lambda \\
\Omega
\end{bmatrix}
\]

(4.2)

The superscript denotes the different corner points. The above can be written as: $B = AX$

\[ X = (A'A)^{-1}A'\] 

(4.3)

Thus, knowing the depth $\lambda$, the motion parameters $u_1, u_2, u_3, \Omega_1, \Omega_2$ and $\Omega_3$ can be calculated.

4.2 Recursive Estimation

From equation (3.5),

\[ \Delta q_k = \begin{bmatrix} S^2(q) & S(q) \end{bmatrix} \begin{bmatrix} u/\lambda \\ \Omega \end{bmatrix} \] 

(4.4)

i.e., $z(k) = H(k)X(k) + n(k)$

(4.5)

Here, $z(k) = \Delta q_k$ is a 3 X 1 observation vector,

\[ X(k) = \begin{bmatrix} u/\lambda \\ \Omega \end{bmatrix} \] 

is a 6 X 1 state vector,

\[ H(k) = \begin{bmatrix} S^2(q) & S(q) \end{bmatrix} \] 

is a 3 X 6 linear map from state space to observation space and is known as the measurement matrix.

$n(k)$ is zero mean Gaussian noise sequence with covariance, $\bar{E}[n(k) n(l)'] = R(k)\delta_{kl}$

where, $\delta_{kl}$ is Kronecker delta function and $R(k)$ is known measurement error covariance matrix (for example camera distortion).

Hence, noise associated with different match points in a single frame is correlated, but the noise between image frames is assumed to be uncorrelated. Since, the body (or camera) is assumed to have undergone uniform motion, we can write:

\[ \begin{bmatrix} u/\lambda \\ \Omega \end{bmatrix}_{k+1} = \begin{bmatrix} u/\lambda \\ \Omega \end{bmatrix}_k + w(k) \] 

(4.6)

\[ X(k+1) = X(k) + w(k) = \phi \cdot X(k) + w(k) \] 

(4.7)

where, $\phi = I_{6x6}$ = the state evolution matrix and $w(k)$ = modeling error at the $k^{th}$ instant.

4.3 Bootstrap Technique

Using the recursive motion estimation, we have computed $u/\lambda$ rather than $u$. Since our interest is in computing $u$ alone, we formulate an alternative approach. Consider the state vector:

\[ X = \begin{bmatrix} u \\ \Omega \end{bmatrix} \] 

(4.8)

The dynamic equation is: $X_t = 0$. The corresponding discrete equation is:

\[ X(k+1) = X(k) + w(k) \] 

(4.9)

where, $w(k)$ stands for the modeling error. The measurement equation can be written as:

\[ \Delta q = \begin{bmatrix} D' & S(q) \end{bmatrix} \begin{bmatrix} u \\ \Omega \end{bmatrix} + v(k) \] 

(4.10)

where, $D' = S^2(q)d$ and $d = 1/\lambda$.

The measurement matrix: $H = \begin{bmatrix} D' & S(q) \end{bmatrix}$ contains the unknown term $d$, so we go in for the estimation of $d$ using another Kalman Filter. The state vector is taken to be:

\[ X' = \begin{bmatrix} q' \\ d \end{bmatrix} \] 

(4.11)

The dynamic equation can be written as:

\[ X'_{k+1} = \begin{bmatrix} q' \\ d \end{bmatrix} = f(X') \] 

(4.12)

This is a non-linear state equation, which has to be linearized with respect to the state $X'$. Using Taylor series, the perturbation can be written as:

\[ \Delta X'_{k+1} = \frac{\partial f}{\partial X'} X'_{k+1} + h.o.t \] 

(4.13)
Neglecting the h.o.t (higher order terms), we get,
\[ \Delta X'(k) = G(X'(k))\Delta X'(k) + w(k) \]  
(4.14)
where, \( w(k) \) stands for the modeling error at the \( k^{th} \) instant.
\( G(X'(k)) \) is the matrix of the partial derivatives of \( f(X'(k)) \) with respect to \( X' \).

As \( q'_{k} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \begin{bmatrix} q_{k}' \\ d_{k} \end{bmatrix} \)
i.e., \( z(k) = H'X'(k) + n(k) \)  
(4.15)
where, \( Z_{k} = q_{k}' \) is a 3 X 1 observation matrix
\( H' = \begin{bmatrix} I & 0 \end{bmatrix} \) is a 3 X 4 linear map from the state space to observation space (observation matrix). The Extended Kalman filter can be now applied on this model. The estimates of \( (q, d) \) and \( (u, \Omega) \) can be put in a bootstrap form.

5. Results of Implementation

Now we deal with the results of correspondence and motion. For this reason we had used a toy car and toy rail bogie to capture a sequence of images by fixing the camera position and translating the toy car. Afterwards SUSAN method was applied for feature extraction. In each frame of the sequence we have chosen a set of points, then we applied the matching algorithm. By assuming a focal length of unity, the image co-ordinates are converted into spherical co-ordinates using the equations:-
\[ q_{1} = X_{i}/M, q_{2} = Y_{i}/M, q_{3} = F/M, \]
\[ M = \sqrt{X_{i}^{2} + Y_{i}^{2} + F^{2}} \]

Having got the spherical data, we are now in a position to apply Batch, Recursive and Bootstrap techniques. We take eight points as shown in the frames which has corresponded in all eight frames and their \( x \) and \( y \) coordinates are given for first four frames. In the following table we give the \( x \) and \( y \) coordinates of points selected starting from frame 1 to frame 8.

<table>
<thead>
<tr>
<th>Frame No. 1</th>
<th>Frame No. 8</th>
</tr>
</thead>
<tbody>
<tr>
<td>254-103</td>
<td>232-102</td>
</tr>
<tr>
<td>184-129</td>
<td>166-129</td>
</tr>
<tr>
<td>178-134</td>
<td>160-134</td>
</tr>
<tr>
<td>196-135</td>
<td>179-134</td>
</tr>
<tr>
<td>208-139</td>
<td>186-142</td>
</tr>
<tr>
<td>239-145</td>
<td>214-145</td>
</tr>
<tr>
<td>220-162</td>
<td>201-162</td>
</tr>
<tr>
<td>218-167</td>
<td>206-167</td>
</tr>
</tbody>
</table>

Fig 1: Frames of the first image sequence

5.1 Results of Batch Processing

Since there is a depth velocity ambiguity, the translational velocity can only be estimated to a factor of depth. There are now six motion parameters \( u_{1}/\lambda, u_{2}/\lambda, u_{3}/\lambda, \Omega_{1}, \Omega_{2} \) and \( \Omega_{3} \). Hence we require at least six pairs. So we use the equation (4.3) for computing the estimates of motion between every two consecutive frames. The estimates of motion for the various pairs of frames for the translation case are as follows: The dimensions of parameters \( u/\lambda \) and \( \Omega \) are in sec\(^{-1}\) and rad/sec respectively.

State vector for frame 1 to frame 2 is
\[ (-.3533 , .0003 , .0053 , .0008 , .0095 , -.0373) \]
and for frame 7 to frame 8
\[ (-.4145 , .0040 , -.0023 , -.0252 , .0058 , -.0608) \]

5.2 Results of Recursive Technique

Since, \( u \) is in the same plane as defined by \( (q, X, q_{i}) \), we can write \( u(q_{i}, X, q_{i}) = 0 \), which leads to three equations : (i) \( u(q_{i}, X, q_{i}) = 0 \), (ii) \( u(q_{i}, X, q_{i}) = 0 \) and (iii) \( u(q_{i}, X, q_{i}) = 0 \) to solve for \( u \), where \( q_{1}, q_{2}, q_{3} \) are from the same frame and \( q_{11}, q_{12}, q_{13} \) are the derivatives of \( q_{1}, q_{2}, q_{3} \) respectively. Since the right hand side of these equations is zero, we can obtain \( u \) only within a scale factor. The value of \( u \) obtained above can be used as an initial value, which is then updated by the Kalman filter.

5.3 Results of Bootstrap Technique

Since the results of above two techniques do not give the values of translational velocity directly, we employ the Bootstrap technique of the previous chapter. In this approach the motion estimation is divided into two parts. The first deals with the estimation of depth, which is required in the second
part for the estimation of motion. In each iteration, two sets of Kalman filtering equations are used. The final values are obtained as under:

No. of frames in sequence = 8
Depth Estimated, $\hat{\lambda} = 88.8660$ cm
State vector, $X = (-34.2037\ -1.5927\ -6.1879)$

The dimensions of $u$ in cm/s and $\Omega$ are in rad/sec.

Actual depth obtained by measurement is 87.5 cm and car is having a actual velocity of 35.17 cm/s in negative X-direction.

6 Conclusions

We have made an attempt to formulate the methods for the estimation of motion using spherical projection, which allows the motion equation to be represented in matrix form. Using this form, we have proposed three approaches. The depth obtained from Bootstrap technique is found to be reasonably accurate (of the order 5% error for the cases studied) under the constraints of uncalibrated camera while taking the sequence of images. We have considered a limited number of frames lest the environmental effects would have a bearing on the results. It is found that SUSAN detector requires that the frames should have good contrast for the extraction of features. Though our correspondence algorithm gives matches V and Y sections in different frames, we have used only the corner correspondences in the estimation of motion thus dealing with the restricted number of features.

References

13. Zhengou Zhang, Richard Deriche, Oliver Faugeras, Quang Tuan Loung, A robust technique for matching two uncalibrated images through recovery of unknown epipolar geometry, Artificial Intelligence 78, 1995.