From Depth and Optical Flow to Rigid Body Motion

Xinhua Zhuang
Robert M. Haralick
Yunxin Zhao

Department of Electrical Engineering, FT-10
University of Washington
Seattle, WA 98195

ABSTRACT

The problem 'rigid body motion from depth and optical flow' was investigated in Ballard and Kimball (1983) and by Haralick and Zhuang (1986). Given four non-coplanar spatial points each having associated with them optical flow and depth information, a theoretical solution was uniquely determined in Haralick and Zhuang (1986) where 'depth information' involves not only depths of selected spatial points but also the derivatives of depths taken with respect to the time variable and the image coordinates.

In this article, we will develop an algorithm to uniquely determine the rigid body motion from optical flow and depth, where the depth, however, does not involve any derivative information. Thus, the original assumptions made by Ballard and Kimball (1983), Haralick and Zhuang (1986) are relaxed. The proposed algorithm is appealing in contrast to the existing linear optical flow-motion algorithms; see Zhuang and Haralick (1984), and Zhuang et al. (1987). It requires only three instead of eight optical flow image points required by the algorithms referenced above.

1. Introduction

Determining the relative motion between an observer and his environment is a major problem in Computer Vision. Its applications include mobile robot navigation and the monitoring of dynamic industrial processes.

The topic 'rigid body motion from depth and optical flow' was addressed in Ballard and Kimball (1983), and in Haralick and Zhuang (1986). A theoretical solution was given in Haralick and Zhuang (1986), using four non-coplanar initial spatial points each associated with optical flow and depth information, where 'depth information' involves both depths of the points and derivatives of depths taken with respect to the time variable and the image coordinates.

This article presents an algorithm to uniquely solve the rigid body motion given optical flow image points and depth information, where depth information does not involve the derivatives. Thus the original assumptions used by Ballard and Kimball (1983) and by Haralick and Zhuang (1986) are relaxed.

The proposed algorithm is appealing in contrast to the existing linear optical flow-motion algorithms; see Zhuang and Haralick (1984) and Zhuang et al. (1987) for several supporting reasons:

1. It needs fewer optical flow image points: Three instead of eight points! This could be important in some situations when, without further cues, the reliable optical flow image points could be obtained only at some prominent image feature points such as corners.

2. Noise effects are taken into consideration at the very beginning of the algorithm development. This should tend to make the results better behaved under noise.

3. It returns complete information about the rigid motion, i.e., the instantaneous rotation \( \omega \) and the instantaneous translation \( k \). The previously mentioned linear optical flow-motion algorithms also could uniquely determine the rotation. However, they could only determine the translation direction. They could not determine the translation itself.

4. The range data sensor using laser technology has become commercially available. Thus, requiring the depths in the new algorithm does not constitute a serious restriction.

The article is organized as follows: In Section II, we show how six unknowns, three for \( \omega \) and three for \( k \), obey a system of linear equations given the optical flow image points and depths. We also show how to apply the least-squares approach to the system of linear equations for solving \( \omega \) and \( k \). In Section III, we explicitly show the least-squares solution to \( \omega, k \). In Section IV, we summarize the least-squares solution procedure and in Section V we give a few simulation results. The final section is a summary.

2. Problem Formulation

Suppose a rigid body is in motion in the half space, \( z < 0 \). (See Fig. 1). Let \( p(t) \) be the position vector of an object point at the time \( t \), \( p(t) = (x(t), y(t), z(t)) \). Let
\((X(t),Y(t))\) denote the central projective coordinates of \(p(t)\) onto the image plane. Without loss of generality we take \(z = 1\):

\[
\begin{align*}
X(t) &= x(t)/z(t), \\
Y(t) &= y(t)/z(t), \\
p(t) &= z(t)(X(t),Y(t),1)'.
\end{align*}
\] (1)

Let \((u(t),v(t))\) denote the instantaneous velocity of the moving image point \(X(t),Y(t)\), i.e., \(u(t) = X(t),v(t) = Y(t)\).

We shall call each element \([(X(t),Y(t)),(u(t),v(t))]\) an optical flow image point. Usually, we may not be able to observe all optical flow image points. Denote a finite number of observed optical flow image points by \(q_i = [(X_i(t),Y_i(t)),(u_i(t),v_i(t))]\), \(i = 1,\ldots,n\). In addition, the respective 3D depths \(z_i, i = 1,\ldots,n\), are also observed.

The instantaneous representation of the rigid motion is described by

\[
\dot{p}(t) = \omega(t) \times p(t) + k(t),
\] (2)

where \(\omega(t) = (\omega_1(t),\omega_2(t),\omega_3(t))'\) represents the instantaneous rotational angular velocity and \(k(t) = (k_1(t),k_2(t),k_3(t))'\) the instantaneous translation velocity.

The problem we solve is: Given \(q_i, z_i, i = 1,\ldots,n\) determine \(\omega, k\). Using (1), we could work out \(\dot{p}\) in (2) as follows:

\[
\dot{p} = \dot{z}(X,Y,1)' + z(u,v,o)',
\] (3)

where for simplicity the time variable ‘\(t\)’ has been omitted. Combining (2) and (3) together, it follows

\[
\dot{z}(X,Y,1)' + z(u,v,o)' = z\omega \times (X, Y, 1)' + k,
\] (4)

which can be split into two equations as follows:

\[
\dot{z}(X,Y)' + z(u,v)' = z(\omega_3 - \omega_2 Y, \omega_1 X - \omega_2, \omega_1 X - \omega_3)' + (k_1, k_2, k_3)',
\] (5)

\[
z = z(\omega_1 Y - \omega_2 X) + k_3.
\] (6)

Substituting (6) for \(z\) in (5), it follows that

\[
z(\omega_3 - \omega_2 Y, \omega_1 X - \omega_2, \omega_1 X - \omega_3)' - [z(\omega_1 Y - \omega_2 X, k_3)](X, Y)' + (k_1, k_2, k_3)' - z(u,v)' = 0
\]

or after rearrangement

\[
- z(XY,1+Y^2)'\omega_1 + z[1+X^2,XY]'\omega_2 + z(-X,Y)'\omega_3 \\
+ (1,0)'k_1 + (0,1)'k_2 - (X,Y)'k_3 - z(u,v)' = 0
\] (7)

Plugging optical flow image points \(q_i\) and depths \(z_i, i = 1,\ldots,n\), into (7) and letting

\[
g_i(\omega, k) = z_i(X,Y,1+Y^2)'\omega_1 + z_i[1+X^2,XY]'\omega_2 \\
+ z_i(-X,Y)'\omega_3 + (1,0)'k_1 + (0,1)'k_2 - (X,Y)'k_3,
\] (8)

we obtain \(2n\) linear equations with six unknowns \(\omega_1, \omega_2, \omega_3, k_1, k_2, k_3\):

\[
g_i(\omega, k) - z_i(u_i,v_i)' = 0, \quad i = 1,\ldots, n,
\] (9)

where each index \(i\) implies two linear equations.

Letting

\[
r_k = -z_i(X,Y,1+Y^2)'\omega_1 + z_i[1+X^2,XY]'\omega_2 \\
+ z_i(-X,Y)'\omega_3 + (1,0)'k_1 + (0,1)'k_2 - (X,Y)'k_3,
\]

then

\[
g_i(\omega, k) = (r_k, z_i, r_i, s_i, s_i, s_i) \cdot (\omega_1, \omega_2, \omega_3, k_1, k_2, k_3)' \tag{10}
\]

and (9) can be written as

\[
(r_k, z_i, r_i, s_i, s_i, s_i) \cdot (\omega_1, \omega_2, \omega_3, k_1, k_2, k_3)' = z_i(u_i, v_i)', \quad i = 1,\ldots, n. \tag{11}
\]

Without noise in the measurements of optical flow image points \(q_i\) and depths \(z_i, i = 1,\ldots,n\), the true rotation \(\omega\) and translation \(k\) always satisfy (11). Conversely, if the coefficient matrix of (11) has a rank six, then the unique solution to (11) gives the true rotation \(\omega\) and translation \(k\). Having a rank six requires \(n \geq 3\), i.e., the least necessary number of optical flow image point-depth pairs is three. In practice, more pairs are preferable to increase the probability that the rank of the coefficient matrix of (11) will be equal to six and to smooth out noise effects.

When noise is present in the optical flow image points and depths, Eq. (11) is no longer valid. A reasonable thing to do is to find a least-squares solution, i.e., to minimize the residual function \(e^2(\omega, k)\) which is defined as

\[
e^2(\omega, k) = \sum_{i=1}^{n} \|g_i(\omega, k) - z_i(u_i, v_i)\|^2 \tag{12}
\]

The next section shows how to solve \(\min_{\omega, k} e^2(\omega, k)\).

3. Solving \(\min_{\omega, k} e^2(\omega, k)\)

The minimization of \(e^2(\omega, k)\) with respect to \(\omega, k\) is an unconstrained optimization problem. The necessary condition for a minimal point is given by

\[
\frac{\partial e^2}{\partial \omega_j} = 0, \quad \frac{\partial e^2}{\partial k_j} = 0, \quad j = 1, 2, 3.
\]

Taking partial derivative of \(e^2\) with respect to \(\omega_j\), there results

\[
\frac{\partial e^2}{\partial \omega_j} = 2 \sum_{i=1}^{n} [g_i(\omega, k) - z_i(u_i, v_i)'] \cdot r_i, \quad j = 1, 2, 3. \tag{13}
\]
Taking partial derivative of $e^j$ with respect to $k_j$, there results

$$\frac{\partial e^j}{\partial k_j} = 2 \sum_{i=1}^{n} g_i(\omega, k) - z_i(u_i, v_i)' \cdot s_j', \quad j = 1, 2, 3. \quad (14)$$

Setting these partials to zero results in

$$\begin{align*}
\sum_{i} g_i(\omega, k) \cdot r_j' &= \sum_{i} z_i(u_i, v_i) \cdot r_j', \quad j = 1, 2, 3, \\
\sum_{i} g_i(\omega, k) \cdot s_j' &= \sum_{i} z_i(u_i, v_i) \cdot s_j', \quad j = 1, 2, 3,
\end{align*} \quad (15)$$

which can be rewritten as:

$$\begin{pmatrix}
(\omega_1, \omega_2, \omega_3, k_1, k_2, k_3) \\
\sum_{i} r_i (r_i, r_i, s_i, s_i)' (r_i, r_i, r_i, s_i, s_i, s_i) \\
\sum_{i} s_i (r_i, r_i, r_i, s_i, s_i, s_i) (r_i, r_i, r_i, s_i, s_i, s_i)
\end{pmatrix} = \sum_{i} z_i(u_i, v_i)' (r_i, r_i, r_i, s_i, s_i) (r_i, r_i, r_i, s_i, s_i, s_i),$$

By rearranging we obtain

$$W(\omega_1, \omega_2, \omega_3, k_1, k_2, k_3)' = b, \quad (17)$$

where

$$W = \sum_{i} (r_i, r_i, r_i, s_i, s_i)' (r_i, r_i, r_i, s_i, s_i), \quad b = \sum_{i} z_i(r_i, r_i, r_i, s_i, s_i)' (u_i, v_i)' \cdot \quad (18)$$

It is clear that the $6 \times 6$ symmetric matrix $W$ is a summation of $n$ non-negative Gram matrices $W_i's$, where

$$W = (r_i, r_i, r_i, s_i, s_i)' (r_i, r_i, r_i, s_i, s_i),$$

$$b = \sum_{i} z_i(r_i, r_i, r_i, s_i, s_i)' (u_i, v_i)' \cdot \quad (18)$$

$$\langle \cdot, \cdot \rangle$$ represents the inner product,

$$\begin{align*}
\langle r_i \rangle^2 &= z_i(X_i Y_i Y_i + Y_i + 2 Y_i^2 + 1), \\
\langle r_i \rangle^2 &= z_i(X_i Y_i Y_i + X_i + 2 X_i Y_i + 1), \\
\langle s_i \rangle^2 &= z_i(X_i Y_i + X_i Y_i^2 + 2 X_i Y_i), \\
\langle r_i, r_i \rangle &= -\langle X_i, X_i \rangle, \\
\langle r_i, s_i \rangle &= -\langle X_i, Y_i \rangle, \\
\langle s_i \rangle^2 &= \langle X_i Y_i + Y_i^2, \\
\langle s_i \rangle^2 &= \langle X_i + Y_i^2, \\
\langle s_i, s_i \rangle &= \langle X_i - Y_i, \rangle, \\
\langle s_i, s_i \rangle &= \langle X_i - Y_i, \rangle, \\
\langle s_i, s_i \rangle &= \langle X_i - Y_i, \rangle.
\end{align*}$$

Since $W$ is symmetric, it admits a decomposition of form

$$W = U'DU,$$ \quad (20)

where $U$ is orthonormal and $D$ is diagonal. If the matrix $W$ is positive, i.e., the rank of $W$ equals six, then the least-squares solution to (11) is given by:

$$(\omega_1, \omega_2, \omega_3, k_1, k_2, k_3)' = U'D^{-1}Ub. \quad (21)$$

In the following, we always assume that $n \geq 3$ and the rank of the matrix $W$ equals six.

In the next section, we describe the least-squares procedure described above.

4. Algorithm Summary

Now we are ready to summarize the algorithm as follows:

**Step 1.** Form $W, b$:

$$W = \sum_{i=1}^{n} (r_i, r_i, r_i, s_i, s_i)' (r_i, r_i, r_i, s_i, s_i), \quad b = \sum_{i=1}^{n} z_i(r_i, r_i, r_i, s_i, s_i)' (u_i, v_i)'.$$

**Step 2.** Decompose the positive matrix $W$:

$$W = U'UD.$$

**Step 3.** Compute the least-squares estimation of $\omega, k$:

$$(\omega_1, \omega_2, \omega_3, k_1, k_2, k_3)' = U'D^{-1}Ub.$$

5. Simulation Experiments

The experiments needed to verify the above algorithm should tell us: (1) what is the minimum number of points in order to compute motion and surface structure from accurate or noisy optical flow and depth measurements in practice; (2) what is the likelihood we come across a set of optical flow image points and associated depths that violate the rank assumption, i.e., $\text{Rank}(W) = 6$.

One way of testing the validity of the rank assumption is to generate many sets of data randomly, and to compute the eigenvalues of $W$. If $W$ has a zero eigenvalue, then the rank

395
assumption is violated. This method, however, suffers from the defect that, since most eigenvalue computing routines return inexact values even with accurate given data, we need a threshold around zero below which we would like to say the eigenvalues are zero or nearly zero. This obviously is a problem, since the rank assumption is not violated until one eigenvalue is strictly zero.

The approach used here has therefore been different. We have actually implemented the algorithm and tested it with hundreds sets of randomly chosen data. Implicitly, this tells us the likelihood of the occurrence of sets of optical flow image points and associated depths that violate the rank assumption, since any time we encounter such a set of data, the algorithm will break down. It also answers questions 1 and 3.

Two representative simulation results for different noise cases are given as follows.

5.1 Simulation 1

The true motion parameters for this simulation are: 
\[ w = (-1.59, 1.34, -4.41), \quad k = (0.74, 0.58, -0.34) \].

Original data consisting of nine accurate optical flow image points and depths is tabulated in Table 1. Its noisy version formed by adding Gaussian noise with zero mean and 0.01 (0.001) variance to the optical flow velocities (depths) is tabulated in Table 2. The results obtained by using the data in rows 1-3, 4-6, 7-9 or rows 1-4, 6-9 or rows 1-5, 5-9 are tabulated in Tables 3, 4 and 5, respectively. As seen clearly, the data in rows 7-9 does not give a good result. However, the data in rows 6-9 gives a good result and the data in rows 5-9 gives an even better result.

5.2 Simulation 2

The purpose of this simulation is to test the algorithm with a substantial set of data at different noise levels. A total of two hundred forty data were done. The true motion parameters were the same as in Simulation 1. We generated 100 points of optical flow and depth data. Uniformly distributed noise with zero mean were added to the generated data. Data set sizes of 3, 4, and 5 optic flow points, were generated by selecting sets of 3, 4, or 5 points from the originally generated 100 optic flow points. For each data set size and noise level, 40 such selections were made. Experimental results are tabulated in terms of the average relative estimation error and standard deviation for each motion parameter. The average relative estimation errors are defined as follows,

\[ e_{\omega_i} = \frac{1}{N} \sum_{j=1}^{N} \left| \frac{\omega_i^{(j)} - \omega_i}{\omega_i} \right| \quad i = 1, 2, 3 \]

\[ e_{k_i} = \frac{1}{N} \sum_{j=1}^{N} \left| \frac{k_i^{(j)} - k_i}{k_i} \right| \quad i = 1, 2, 3 \]

The corresponding standard deviations are defined as follows:

\[ S_{\omega_i} = \sqrt{\frac{1}{N} \sum_{j=1}^{N} (\omega_i^{(j)} - \bar{\omega}_i)^2} \quad i = 1, 2, 3 \]

\[ S_{k_i} = \sqrt{\frac{1}{N} \sum_{j=1}^{N} (k_i^{(j)} - \bar{k}_i)^2} \quad i = 1, 2, 3 \]

where \( N \) is the total number of estimates.

Two noise levels of uniform noise were used. The range of the uniform distribution was \( \pm 1 \% \) or \( \pm 5 \% \) of the data range. These results are summarized in Table 6 and 7 which correspond to the uniform noise level image being \( \pm 1 \% \) and \( \pm 5 \% \), respectively. These two tables demonstrate that to keep the relative estimation errors to be the same size as the relative range on the uniform noise, the data set size must have at least 5 points each. Several observations from our experiments are as follows.

In spite of testing the algorithm with hundreds sets of randomly generated optical flow image points and associated depths, we have not met a set of optical flow image points and associated depths that violate the rank assumption.

For the case where there is no noise in the optical flow velocities and depths, the algorithm is extremely accurate. Even when there is noise in the optical velocities and depths, the algorithm works well.

6. Summary

Given three optical flow image point-depth pairs, \((r_i, z_i), i = 1, 2, 3\), the instantaneous rigid motion can be uniquely determined whenever the 6 \( \times \) 6 matrix

\[
\begin{bmatrix}
  r_1^1 & r_1^2 & r_1^3 & s_1^1 & s_1^2 & s_1^3 \\
  r_2^1 & r_2^2 & r_2^3 & s_2^1 & s_2^2 & s_2^3 \\
  r_3^1 & r_3^2 & r_3^3 & s_3^1 & s_3^2 & s_3^3 \\
\end{bmatrix}
\]

has a rank six. When noise is present, there exists an algorithm to estimate the rigid motion as a least-squares solution given by \((\omega_1, \omega_2, \omega_3, k_1, k_2, k_3) = U'D^{-1}Ub\), where

\[ U'DU = W = \sum_i(r_i^1, r_i^2, r_i^3, s_i^1, s_i^2, s_i^3)(r_i^1, r_i^2, r_i^3, s_i^1, s_i^2, s_i^3)' \]

\[ b = \sum_i z_i(r_i^1, r_i^2, r_i^3, s_i^1, s_i^2, s_i^3)'(u_i, v_i)' \]

All \( r_i^j, s_i^j \) are \( 2 \times 1 \) vector and defined in Section 2.
References


![Fig.1 Imaging Geometry](image)

Table 1. Original Noiseless Data

<table>
<thead>
<tr>
<th>N</th>
<th>X</th>
<th>Y</th>
<th>U</th>
<th>V</th>
<th>Z</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.78</td>
<td>-0.97</td>
<td>16.0909</td>
<td>-4.7487</td>
<td>-0.0760</td>
</tr>
<tr>
<td>2</td>
<td>0.077</td>
<td>-0.35</td>
<td>5.87327</td>
<td>-1.0996</td>
<td>-0.3460</td>
</tr>
<tr>
<td>3</td>
<td>-0.008</td>
<td>-0.009</td>
<td>-29.1639</td>
<td>-20.0030</td>
<td>-0.0246</td>
</tr>
<tr>
<td>4</td>
<td>1.1</td>
<td>1.6</td>
<td>-11.0584</td>
<td>-20.0324</td>
<td>-0.0456</td>
</tr>
<tr>
<td>5</td>
<td>0.28</td>
<td>0.025</td>
<td>-4.0075</td>
<td>-4.0278</td>
<td>-0.0225</td>
</tr>
<tr>
<td>6</td>
<td>0.28</td>
<td>-0.01</td>
<td>-5.0459</td>
<td>-2.0249</td>
<td>-0.0300</td>
</tr>
<tr>
<td>7</td>
<td>-0.27</td>
<td>0.026</td>
<td>-4.0004</td>
<td>-3.0106</td>
<td>-0.0040</td>
</tr>
<tr>
<td>8</td>
<td>-3.4</td>
<td>-0.7</td>
<td>-32.0030</td>
<td>-50.0730</td>
<td>-0.0235</td>
</tr>
<tr>
<td>9</td>
<td>0.12</td>
<td>-0.52</td>
<td>-6.0012</td>
<td>-1.5146</td>
<td>-0.0380</td>
</tr>
</tbody>
</table>

Table 2. Noisy data obtained by adding zero mean Gaussian noise having variance .01 to the optic flow velocities and variance .001 to the optic flow depths.

<table>
<thead>
<tr>
<th>N</th>
<th>X</th>
<th>Y</th>
<th>U</th>
<th>V</th>
<th>Z</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.78</td>
<td>-0.97</td>
<td>16.0909</td>
<td>-4.7323</td>
<td>-0.0755</td>
</tr>
<tr>
<td>2</td>
<td>0.077</td>
<td>-0.35</td>
<td>5.87327</td>
<td>-1.0996</td>
<td>-0.3460</td>
</tr>
<tr>
<td>3</td>
<td>-0.008</td>
<td>-0.009</td>
<td>-29.1639</td>
<td>-20.0030</td>
<td>-0.0246</td>
</tr>
<tr>
<td>4</td>
<td>1.1</td>
<td>1.6</td>
<td>-11.0584</td>
<td>-20.0324</td>
<td>-0.0456</td>
</tr>
<tr>
<td>5</td>
<td>0.28</td>
<td>0.025</td>
<td>-4.0075</td>
<td>-4.0278</td>
<td>-0.0225</td>
</tr>
<tr>
<td>6</td>
<td>0.28</td>
<td>-0.01</td>
<td>-5.0459</td>
<td>-2.0249</td>
<td>-0.0300</td>
</tr>
<tr>
<td>7</td>
<td>-0.27</td>
<td>0.026</td>
<td>-4.0004</td>
<td>-3.0106</td>
<td>-0.0040</td>
</tr>
<tr>
<td>8</td>
<td>-3.4</td>
<td>-0.7</td>
<td>-32.0030</td>
<td>-50.0730</td>
<td>-0.0235</td>
</tr>
<tr>
<td>9</td>
<td>0.12</td>
<td>-0.52</td>
<td>-6.0012</td>
<td>-1.5146</td>
<td>-0.0380</td>
</tr>
</tbody>
</table>

Table 3 shows the estimated rotation, translation, and their error using noisy data points from rows 1, 2, 3, noisy data points from rows 4, 5, 6, and noisy data points from rows 7, 8, 9 of Table 2.

<table>
<thead>
<tr>
<th>Data Points Used</th>
<th>Estimated Rotation</th>
<th>Estimated Translation</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>( u_1, u_2, u_3 )</td>
<td>( \theta_1, \theta_2, \theta_3 )</td>
<td>( k_1, k_2, k_3 )</td>
<td>( 10^{-6} )</td>
</tr>
<tr>
<td>1-3</td>
<td>-1.55, 1.41, -4.67</td>
<td>0.74, 0.58, -0.26</td>
<td>0.08</td>
</tr>
<tr>
<td>4-6</td>
<td>-1.76, 1.61, -4.28</td>
<td>0.78, 0.51, -0.34</td>
<td>0.12</td>
</tr>
<tr>
<td>7-9</td>
<td>-0.26, -0.80, -5.1</td>
<td>0.55, 0.41, -0.59</td>
<td>1.42</td>
</tr>
</tbody>
</table>

Table 4 shows the estimated rotation, translation and their error using noisy data points from rows 1, 2, 3, and noisy data points from rows 6, 7, 8, 9 of Table 2.

<table>
<thead>
<tr>
<th>Data Points Used</th>
<th>Estimated Rotation</th>
<th>Estimated Translation</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>( u_1, u_2, u_3 )</td>
<td>( \theta_1, \theta_2, \theta_3 )</td>
<td>( k_1, k_2, k_3 )</td>
<td>( 10^{-6} )</td>
</tr>
<tr>
<td>1-4</td>
<td>-1.63, 1.31, -4.38</td>
<td>0.75, 0.58, -0.33</td>
<td>0.003</td>
</tr>
<tr>
<td>5-9</td>
<td>-1.34, 0.87, -4.35</td>
<td>0.68, 0.55, -0.33</td>
<td>0.001</td>
</tr>
</tbody>
</table>

Table 5 shows the estimated rotation, translation and their error using noisy data points from rows 1, 2, 3, and noisy data points from rows 5, 6, 7, 8, 9 of Table 2.

<table>
<thead>
<tr>
<th>Data Points Used</th>
<th>Estimated Rotation</th>
<th>Estimated Translation</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>( u_1, u_2, u_3 )</td>
<td>( \theta_1, \theta_2, \theta_3 )</td>
<td>( k_1, k_2, k_3 )</td>
<td>( 10^{-6} )</td>
</tr>
<tr>
<td>1-5</td>
<td>-1.61, 1.36, -4.41</td>
<td>0.75, 0.58, -0.34</td>
<td>0.0001</td>
</tr>
<tr>
<td>5-9</td>
<td>-1.34, 0.87, -4.35</td>
<td>0.68, 0.55, -0.33</td>
<td>0.001</td>
</tr>
</tbody>
</table>

Table 6 shows results when the range of the uniform noise level is 1% of the data range. Each result is obtained from 40 simulations.

<table>
<thead>
<tr>
<th>Data Set Size</th>
<th>Used in Estimation</th>
<th>Estimated Error Mean &amp; Standard Deviation</th>
<th>Rotation</th>
<th>Motion Parameters</th>
<th>Translation</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \epsilon )</td>
<td>( \theta_1, \theta_2, \theta_3 )</td>
<td>( k_1, k_2, k_3 )</td>
<td>( \epsilon_1 )</td>
<td>( \epsilon_2 )</td>
<td>( \epsilon_3 )</td>
</tr>
<tr>
<td>3</td>
<td>0.0170</td>
<td>0.0114</td>
<td>0.0097</td>
<td>0.0152</td>
<td>0.0184</td>
</tr>
<tr>
<td>4</td>
<td>0.0066</td>
<td>0.0038</td>
<td>0.0020</td>
<td>0.0160</td>
<td>0.0207</td>
</tr>
<tr>
<td>5</td>
<td>0.0064</td>
<td>0.0039</td>
<td>0.0030</td>
<td>0.0160</td>
<td>0.0207</td>
</tr>
<tr>
<td>6</td>
<td>0.0064</td>
<td>0.0039</td>
<td>0.0030</td>
<td>0.0160</td>
<td>0.0207</td>
</tr>
</tbody>
</table>

Table 7 shows results when the range of uniform noise level is 5% of the data range. Each result is obtained for 40 simulations.

<table>
<thead>
<tr>
<th>Data Set Size</th>
<th>Used in Estimation</th>
<th>Estimated Error Mean &amp; Standard Deviation</th>
<th>Rotation</th>
<th>Motion Parameters</th>
<th>Translation</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \epsilon )</td>
<td>( \theta_1, \theta_2, \theta_3 )</td>
<td>( k_1, k_2, k_3 )</td>
<td>( \epsilon_1 )</td>
<td>( \epsilon_2 )</td>
<td>( \epsilon_3 )</td>
</tr>
<tr>
<td>3</td>
<td>0.3315</td>
<td>0.3051</td>
<td>0.1239</td>
<td>0.3680</td>
<td>0.3166</td>
</tr>
<tr>
<td>4</td>
<td>0.4344</td>
<td>0.4069</td>
<td>0.1351</td>
<td>0.6480</td>
<td>0.7692</td>
</tr>
<tr>
<td>5</td>
<td>0.4435</td>
<td>0.4092</td>
<td>0.1351</td>
<td>0.6480</td>
<td>0.7692</td>
</tr>
<tr>
<td>6</td>
<td>0.4435</td>
<td>0.4092</td>
<td>0.1351</td>
<td>0.6480</td>
<td>0.7692</td>
</tr>
<tr>
<td>7</td>
<td>0.4435</td>
<td>0.4092</td>
<td>0.1351</td>
<td>0.6480</td>
<td>0.7692</td>
</tr>
<tr>
<td>8</td>
<td>0.4435</td>
<td>0.4092</td>
<td>0.1351</td>
<td>0.6480</td>
<td>0.7692</td>
</tr>
<tr>
<td>9</td>
<td>0.4435</td>
<td>0.4092</td>
<td>0.1351</td>
<td>0.6480</td>
<td>0.7692</td>
</tr>
</tbody>
</table>