Incorporating Motion Error in Multi-Frame Structure from Motion

J. Oliensis and J. Inigo Thomas
Computer Science Department
University of Massachusetts at Amherst
Amherst, MA 01003

Abstract

Recovering structure from motion even using information from multiple image frames is difficult, in part because motion error can introduce large, correlated errors in the structure estimate. We propose a method for recursively recovering structure from motion that can deal with this problem. Encouraging results on real images and synthetic data are reported.

1 Introduction

Two frame structure from motion is known to be inaccurate. The reasons for this are familiar: for two frames, each structure value is determined by a single measurement which is unreliable for distant points, and which is strongly affected by small errors in the motion. The natural remedy to this problem is to refine the structure estimation by combining measurements from many frames [1] [2] [11] [6] [13] [12] [15]. In this paper, we propose an implementation of this procedure which differs from previous approaches, in that the motion error is explicitly taken into account.

The difficulty of multi-frame structure from motion has several sources. First, even assuming the motion is known, determining the structure of a point from two frames is a non-linear problem, and biased in the depth estimate. We have studied this problem experimentally in [9]. Our results indicate that in fact the non-linearity of the pure structure measurement is not a serious problem—it can be compensated for by a multiplicity of measurements from different camera positions.

Another, probably more serious difficulty is the motion error. It is known that small errors in the rotation can introduce relatively large errors in depth measurements [3]. Moreover, motion error introduces strong correlations in the structure errors. For instance, if the translation is identified incorrectly, then the measured positions of all points will be displaced away from their actual ones in approximately the same direction: their position errors will be correlated. An analogous result holds for rotation error. Conversely, the correlations in the structure errors are the record of the motion error. In order to include the potentially large effects of motion error, and to compensate for them, a record of these correlations must be maintained in the current error estimate. This information can then be used by a recursive algorithm for updating the old structure estimate based on new image information. This explicit inclusion of motion error is the main new contribution of our paper.

2 Overview of the Algorithm

The camera is assumed to be navigating in an unknown, fixed environment, consisting of isolated 3D points. The correspondences of these points are assumed given. The goal is to determine the locations of the 3D points in the current camera frame.

If no assumption about the motion is made (such as smoothness), the complete information on the camera position for an image is just contained in that image; therefore, it can be determined only with a limited accuracy. On the other hand, each image can be considered a new, independent measurement of the structure—or rather the shape [16]. Thus, it is possible in principle to refine the shape estimate to arbitrary accuracy using a large number of frames.

We adopt here a recursive procedure for determining shape. An obvious strategy is, for each new image, to combine the old shape estimate with the information contained in the new image, weighted by their respective uncertainties. This strategy, which we call shape from pose, is not used in this paper, since each image only partially constrains the shape. Instead, we use Horn's relative orientation algorithm [7] for image frame pairs as our "measurement". This algorithm is relatively robust; and, because of the simplicity of its objective function, an approximate error analysis is relatively tractable. However, other 2 (or 3) frame motion algorithms could be used equally well.

At each time step, Horn's algorithm is used to recover the structure and motion. Also, the structure error of this algorithm is estimated, consisting of the complete covariance matrix, including cross-correlations between different 3D points. These cross-correlations represent the effects of motion error, as stated above.

Then the output of Horn's algorithm is fused with the previous shape estimate to derive an improved estimate. We always maintain the shape estimate in the coordinate system of the moving camera. Thus, to compute the fused estimate, both the old shape estimate and its estimated error must be transformed into the current system. This transformation is not exactly known and induces additional error. Finally,
the old estimate and new measurement are fused by
the standard Kalman filter method.

3 Estimating the Motion Error

Horn's algorithm computes the motion between the
previous (left) and most recent (right) camera frames
by minimizing the objective function:

\[ E = \sum_i (b \cdot (\hat{F}_i \times \hat{r}_i))^2 \equiv \sum_i E_i^2. \] (1)

(We have adopted the notation of \([7]\).) We compute
the error in this motion estimate by linearizing, as is
standard. Thus, we need to calculate the derivatives
of the motion parameters with respect to the image co-
ordinates. We use \( W \) to represent the five motion pa-
rameters: two for the translation direction, and three
for the rotation. The translation magnitude is omit-
ated since it cannot be recovered due to the well-known
scale ambiguity under arbitrary motion. Also, the
image coordinates of the two images are represented by
a vector \( V \) of length \( 4m \), where \( m \) is the number of
3D points.

The recovered values of the motion parameters cor-
respond to a minimum of the objective function \( E \),
and therefore \( \partial E / \partial W = 0 \). After a perturbation in
the image coordinates, the perturbed motion parameters
recovered by Horn's algorithm minimize the new
\( E \). Thus the derivative above is again zero when eval-
uated at the new values for \( V \) and \( W \): it has not
changed in value. This implies that:

\[ \frac{\partial^2 E}{\partial W \partial V} dV + \frac{\partial^2 E}{\partial W \partial W} dW = 0 \] (2)

for the small perturbations \( dV \) and \( dW \). Defining

\[ N = \frac{\partial^2 E}{\partial W \partial V}, \quad M = \frac{\partial^2 E}{\partial W \partial W} \] (3)

and assuming \( M \) has an inverse, equation 2 can be
rewritten as

\[ \frac{\partial W}{\partial V} = -M^{-1} N \] (4)

The motion error covariance can be written in terms
of the \( 5 \times 4m \) matrix \( \partial W / \partial V \):

\[ \text{Cov}(W) \equiv E\{dW dW^T\} \simeq \frac{\partial W}{\partial V} E\{dV dV^T\} \frac{\partial W^T}{\partial V}. \] (5)

Assuming that the image noise at each pixel is in-
dependent and has the same standard deviation \( \sigma \), the
covariance of \( dV \) is proportional to the identity matrix,
and the linearized estimate of the motion error is
the \( 5 \times 5 \) matrix:

\[ \sigma^2 \frac{\partial W}{\partial V} \frac{\partial W^T}{\partial V}. \] (6)

3.1 Determining the \( M \) matrix

First, the motion parameters incorporated in \( W \)
must be specified. Let \( \hat{b} \) be the unit translation vector,
and let \( \hat{b}_e \) be the translation actually recovered by
Horn's algorithm. Then an arbitrary \( \hat{b} \) near \( \hat{b}_e \) can be represented as:

\[ \hat{b} = b_T + \sqrt{1 - |b_T|^2} \hat{b}_e. \] (7)

where \( b_T \) is in the plane perpendicular to \( \hat{b}_e \). This
representation is adequate because for this linearized
analysis we are assuming the motion error to be small.
\( b_T \) is represented explicitly by its projection
on two perpendicular axes \( \hat{e}_1 \) and \( \hat{e}_2 \) in the plane normal
to \( \hat{b}_e \).

The rotation \( R \) is represented using quaternions,
but in a slightly unusual notation. Let the axis of
rotation be \( \hat{n} \), and the rotation angle be \( \theta \). Then the
unit quaternion is:

\[ \hat{R} = (\cos \frac{\theta}{2}, \hat{n} \sin \frac{\theta}{2}) \equiv (R_0, \hat{R}), \] (8)

with \( R_0^2 + \hat{R}^2 = 1 \). The representation is given by
the three-dimensional vector \( \hat{R} \). The rotation matrix is
given by:

\[ \hat{R}_i = R(\hat{R}_i) = (1 - 2\hat{R}^2)\hat{R}_i + 2\hat{R}\hat{R}_i \hat{R}. \] (9)

The vector \( \hat{R} \) is represented in terms of its projections
along three orthogonal 3D directions \( \hat{n}_1 \), \( \hat{n}_2 \) and \( \hat{n}_3 \).

The first partial derivatives are:

\[ \frac{\partial E}{\partial (b_T \cdot \hat{e})} = 2 \sum_i E_i(\hat{e} \cdot \frac{\hat{e} \cdot b_T}{\sqrt{1 - |b_T|^2}} \hat{b}_e) \cdot (\hat{F}_i \times \hat{r}_i), \]

\[ \frac{\partial E}{\partial (\hat{R} \cdot \hat{n})} = -2 \sum_i E_i\hat{b} \cdot \hat{r}_i \times \hat{C}_{\hat{R} \cdot \hat{n}}(\hat{F}_i), \] (10)

where \( \hat{C}_{\hat{R} \cdot \hat{n}}(\hat{F}_i) \) is obtained by differentiating eq. 9.

Evaluating the second partial derivatives at the un-
perturbed solution, where \( \hat{b} = \hat{b}_e \) and \( b_T = 0 \), we get:

\[ \frac{\partial^2 E}{\partial (b_T \cdot \hat{e}) \partial (b_T \cdot \hat{e})} = 2 \sum_i (\hat{e} \cdot (\hat{F}_i \times \hat{r}_i))\hat{n} \cdot (\hat{F}_i \times \hat{r}_i), \]

\[ \frac{\partial^2 E}{\partial (b_T \cdot \hat{e}) \partial (\hat{R} \cdot \hat{n})} = -\hat{e} \cdot \hat{n} (\hat{b}_e \cdot (\hat{F}_i \times \hat{r}_i))^2, \] (11)

\[ \frac{\partial^2 E}{\partial (b_T \cdot \hat{e}) \partial (\hat{R} \cdot \hat{n})} = -2 \sum_i (\hat{e} \cdot (\hat{F}_i \times \hat{r}_i))(\hat{b}_e \cdot (\hat{F}_i \times \hat{C}_{\hat{R} \cdot \hat{n}}(\hat{F}_i))) \]
These equations determine the elements of $M$. $M$ could also be calculated in the spirit of Horn's paper \[7\], by summing first over all image points prior to calculating the derivatives. The computation of the $A'$ matrix is omitted due to lack of space \[10\].

### 4 Determining the Structure Error

To calculate the estimated structure error due to Horn's algorithm, we calculate the matrix of partial derivatives of the structure with respect to the image coordinates, just as for the motion. The equation for the location of the $i$-th 3D point, $p_i$, is:

$$
p_i = \frac{b(\hat{R} \times \hat{r}_i) \cdot (R \times \hat{r}_i \times \hat{r}_i)}{|R \times \hat{r}_i \times \hat{r}_i|^2} \cdot \hat{r}_i,
$$

where $\hat{r}_i$ is a 3D vector of the (right) image coordinates and focal length. $p_i$ depends on the image coordinates partly through the motion parameters $W$, which are themselves functions of these coordinates. The partial derivatives of $p_i$ can therefore be computed by the chain rule. We represent the collection of 3D structure estimates by the state vector $P_n = (p_1, \ldots, p_m)$, which has length $3m$. Then:

$$
dP = \frac{\partial P_n}{\partial W} dW + \frac{\partial P_n}{\partial V} dV
$$

Since the matrix $(\partial W/\partial V)$ has already been computed above, to calculate $dP/dV$ we need only to compute $\partial P_n/\partial W$ and $\partial P_n/\partial V$. This computation is omitted, again due to lack of space \[10\]. Finally, the estimated covariance of the structure is:

$$
Cov(P_n) = \frac{dP_n}{dV} Cov(V) \frac{dP_n}{dV} = \sigma^2 dP_n dP_n^T
$$

### 5 Fusing the New and Old Structure Estimates

Our aim is to combine the current estimate of the structure $P_n$ with the previous one, $P_{n-1}$, in order to produce a new estimate in the current camera frame. Also, the combined estimate of structure error should be updated to this frame. This combination requires that the past estimate be moved to the current coordinate frame using the calculated motion parameters. The moved structure estimate can be written as:

$$
P_n(-) = C_n(P_{n-1}) - B_n,
$$

where $C_n$, a $3m \times 3m$ matrix, is defined by $C_n = \text{diag}(R_n, R_{n-1}, R_{n-2})$, with $R_n$ the estimated rotation between the coordinate frames. $B_n$, a $3m \times 1$ matrix, is defined similarly in terms of the translation. Note that the transition matrices $C_n$ and $B_n$ are noisy.

The expected error of $P_n(-)$ in the new coordinate system is computed by linearising:

$$
dP_n(-) = \frac{\partial P_n(-)}{\partial P_{n-1}} dP_{n-1} + \frac{\partial P_n(-)}{\partial W} dW + \frac{\partial P_n(-)}{\partial V} dV
$$

Define the $3m \times 5$ matrix $A_n$ by:

$$
A_n = \begin{bmatrix}
\frac{\partial P_n(-)}{\partial W} & \frac{\partial W_n}{\partial V}
\end{bmatrix}
$$

Then the covariance is:

$$
Cov(dP_n(-)dP_n(-)^T) = C_n Cov(dP_{n-1}dP_{n-1}^T)C_n^T + A_n Cov(dV_n dV_n) A_n^T
$$

Here we have assumed that $W_n$ and $P_n(-)$ are statistically independent; probably, after a few frames, the correlation between the left image and the previous combined estimate can be neglected safely.

Lastly, the structure from Horn's algorithm and $P_n(-)$ must be combined to give the new estimate $P_n(+)$, which is straightforward except for the overall scale ambiguity. Ideally, the scale should be removed from the state vector; currently, we fix the scale of the translation step to be its exact value. The overall drift induced by an incorrect treatment of scale is probably insignificant. Then, the standard Kalman filter result \[5\] for the fused estimate is:

$$
P_n(+) = Cov(P_n(+)) Cov(P_n(-))^{-1} P_n(-) + Cov(P_n(+) - P_n(-))
$$

where $Cov(P_n(+))$ is the new covariance of the combined estimate, given by the usual expression \[5\]. All ingredients have been assembled for computing the fused structure.
6 Experiments

We have carried out a series of experiments using the algorithm described above for one synthetic and two real image sequences. Recall that it is the shape rather than the structure that we expect to determine with good accuracy. Accordingly, we use Horn's absolute orientation algorithm [8] to determine the rotation, translation, and scale which bring the reconstructed positions of all 3D points into the best registration with their true positions. The average distance between the true and estimated positions of the two real image sequences. Recall that it is the shape of the object, not the absolute orientation that we are interested in.

The average distance between the 3D points after registration is our measure of the error in the shape determination.

The parameters of our experimental sequences are shown in Table 1. For all sequences, the focal length was 16 mm. In the first experiment, a synthetic image sequence was used. The camera was translated along its current z direction with a translation varying in size from 0.44 to 0.6, while rotating by 1.5° about the y (vertical) direction. The sign of the rotation was flipped after the first step, and then after every two steps. The image noise was Gaussian, with \( \sigma = 0.3 \) pixels, modelling a flow error \( \sigma = 0.42 \).

The average 3D error for our recursive algorithm, displayed in Table 2.1, decreases swiftly and monotonically, contrasting with the 2 frame error, and with the results reported in [2], for a recursive algorithm which does not include the effects of motion error. These results were achieved even though the translational motion was largely into z, with a very small baseline to triangulate with. If a single outlier point is included, the average structure error for our algorithm improves significantly; after the last iteration, it is only 1.7 units. Table 2.2 displays the unregistered depths recovered after the last frame for all points. The sizes of the transformations computed by absolute orientation are shown in Table 2.2. Compared to the two-frame results, a much smaller transformation is required to register the recursive structure estimate with the ground truth.

The second experiment employs an image sequence previously discussed in [14]; the first image is displayed in Figure 1. The motion consisted of a box rotating around its approximately vertical body axis in steps of about 3.6°, with a stationary camera. The average 3D error for our recursive algorithm decreases dramatically compared to the two-frame result after just 2-3 frames (Table 3.1). This fast decrease can be attributed to our correctly combining successive measurements to obtain an effectively wider baseline. The slight increase in the error after iteration 4 is probably due to the fortuitously low errors in iterations 3 and 4. Even in the final iteration, the 1.7 mm error is comparable to the accuracy of the ground truth, which is about 1.5 mm. This accuracy is high considering the large depths (~ 600 mm). In Table 3.2, the depths after registration are displayed for the last frame. There are no outliers. Finally, the sizes of the transformations for absolute orientation are shown in Table 3.3.

For the third experiment, a mobile robot was moved down a hallway [17]; the first image is shown in Figure 2. Only approximate ground truth was available. A few additional points were tracked over some but not all frames, or had no ground truth. These points when present were used in computing the two-frame motion. The robot motion was approximately a translation in the z direction of 1.95 feet, with unknown small rotations. Several of the tracked points lay on the occluding boundaries of cones, and accordingly were not well-defined 3D points.

Also, because of the small number of points, absolute orientation could not be used. In Table 4.1, we display the average 3D distance error for 6 of the 8 tracked points. (2 were excluded because they are spatially distant and close to the FOE in every image. Thus their depths are poorly determined both by the 2 frame and recursive algorithms.) The recursive algorithm does much better than the two-frame algorithm. The depths computed in each frame for the tracked points are displayed in Table 4.2. It seems that because of the consistency of the motion, and nearly constant small baselines, the recursive algorithm acts as if it is simply combining independent measurements with similar uncertainties. The results are reasonably good, considering the intrinsic uncertainties in the 3D point locations on the cones.

Acknowledgements

We thank Prof. V. Aurich, J. Weule, and the University of Düsseldorf for providing computational resources and technical help. We also thank H. Sawhney, T. Kumar, R. Mannathu, B. Collins, B. Draper, R. Beveridge, and R. Heller for their assistance.

References


8. Table 1. Parameters for the three motion sequences used.

<table>
<thead>
<tr>
<th>Sequence</th>
<th># Frames</th>
<th># pts.</th>
<th>Depth Range</th>
<th>FOV</th>
<th>Image</th>
</tr>
</thead>
<tbody>
<tr>
<td>Synthetic</td>
<td>13</td>
<td>20</td>
<td>43-65</td>
<td>45°</td>
<td>512 x 512</td>
</tr>
<tr>
<td>Box</td>
<td>8</td>
<td>12</td>
<td>585-707 mm</td>
<td>(23.4°, 22.4°)</td>
<td>256 x 242</td>
</tr>
<tr>
<td>Cones</td>
<td>6</td>
<td>8</td>
<td>34-87 ft.</td>
<td>24.03°</td>
<td>256 x 256</td>
</tr>
</tbody>
</table>

Table 2.1. Average 3D error after absolute orientation for synthetic sequence. For the recursive algorithm, most of the error in 9-11 is due to a single outlier point.

<table>
<thead>
<tr>
<th>Rot. (dg)</th>
<th>Trans.</th>
<th>Scale</th>
</tr>
</thead>
<tbody>
<tr>
<td>P</td>
<td>13.7</td>
<td>0.87</td>
</tr>
<tr>
<td>R</td>
<td>-6</td>
<td>-0.6</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Iter.</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frame</td>
<td>22.2</td>
<td>12.0</td>
<td>15.2</td>
<td>25.9</td>
<td>10.2</td>
<td>17.9</td>
<td>14.5</td>
<td>24.0</td>
<td>10.3</td>
<td>14.8</td>
<td>11.9</td>
</tr>
<tr>
<td>Recur.</td>
<td>11.5</td>
<td>11.9</td>
<td>9.9</td>
<td>9.3</td>
<td>8.6</td>
<td>7.6</td>
<td>5.9</td>
<td>5.0</td>
<td>4.5</td>
<td>3.6</td>
<td>3.4</td>
</tr>
</tbody>
</table>

Table 2.2. Unregistered depths in last frame for 20 tracked points.

<table>
<thead>
<tr>
<th>2 Rot. (dg)</th>
<th>-10</th>
<th>-8</th>
<th>-22</th>
<th>-4</th>
<th>-23</th>
<th>-19</th>
<th>-19</th>
<th>-9</th>
<th>-6</th>
<th>-13</th>
</tr>
</thead>
<tbody>
<tr>
<td>F Trans.</td>
<td>13.7</td>
<td>18.6</td>
<td>42.3</td>
<td>6.7</td>
<td>27.9</td>
<td>18.3</td>
<td>35.8</td>
<td>8.9</td>
<td>17.9</td>
<td>13.9</td>
</tr>
<tr>
<td>R Scale</td>
<td>0.87</td>
<td>0.73</td>
<td>0.32</td>
<td>0.85</td>
<td>0.46</td>
<td>0.85</td>
<td>0.42</td>
<td>0.99</td>
<td>0.77</td>
<td>0.81</td>
</tr>
<tr>
<td>E Trans.</td>
<td>-6</td>
<td>-1</td>
<td>-0.2</td>
<td>-0.5</td>
<td>-0.8</td>
<td>-0.5</td>
<td>-0.9</td>
<td>-0.8</td>
<td>-1.2</td>
<td>-1.0</td>
</tr>
<tr>
<td>R Scale</td>
<td>11.8</td>
<td>8.7</td>
<td>8.0</td>
<td>6.7</td>
<td>5.4</td>
<td>3.8</td>
<td>3.0</td>
<td>2.4</td>
<td>1.9</td>
<td>1.8</td>
</tr>
</tbody>
</table>

Table 2.3. Magnitudes of the rotation, translation, scale used for registration in the last 10 iterations.

<table>
<thead>
<tr>
<th>Iteration</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 Frame (mm)</td>
<td>4.66</td>
<td>4.88</td>
<td>13.97</td>
<td>3.01</td>
<td>5.55</td>
<td>6.37</td>
<td>5.87</td>
</tr>
<tr>
<td>Recursive (mm)</td>
<td>4.67</td>
<td>10.9</td>
<td>3.02</td>
<td>1.15</td>
<td>0.88</td>
<td>1.51</td>
<td>1.7</td>
</tr>
</tbody>
</table>

Table 2.1. Average 3D error after absolute orientation for box sequence.
Table 3.2. Depths (mm) after registration in last frame for 12 tracked points.

<table>
<thead>
<tr>
<th></th>
<th>True</th>
<th>598.7</th>
<th>611.9</th>
<th>589.1</th>
<th>600.6</th>
<th>654.9</th>
<th>629.9</th>
<th>644.8</th>
<th>665.8</th>
<th>669.7</th>
<th>687.8</th>
<th>629.7</th>
<th>616.2</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 Fr.</td>
<td></td>
<td>607.0</td>
<td>613.9</td>
<td>595.4</td>
<td>596.2</td>
<td>649.0</td>
<td>653.2</td>
<td>648.3</td>
<td>668.6</td>
<td>657.8</td>
<td>685.2</td>
<td>631.2</td>
<td>611.3</td>
</tr>
<tr>
<td>Rec.</td>
<td></td>
<td>600.3</td>
<td>611.7</td>
<td>591.0</td>
<td>600.5</td>
<td>652.8</td>
<td>650.4</td>
<td>644.0</td>
<td>665.0</td>
<td>667.9</td>
<td>687.1</td>
<td>632.6</td>
<td>615.8</td>
</tr>
</tbody>
</table>

Table 3.3. Magnitudes of the registration transformations.

<table>
<thead>
<tr>
<th></th>
<th>2 Fr. Rot. (dg)</th>
<th>-2.0</th>
<th>-3.9</th>
<th>-12.4</th>
<th>-2.2</th>
<th>-2.2</th>
<th>-3.2</th>
<th>-2.1</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Recur. Rot. (deg)</td>
<td>-2.0</td>
<td>-12.3</td>
<td>-1.9</td>
<td>-0.2</td>
<td>-0.3</td>
<td>-1.1</td>
<td>-2.1</td>
</tr>
<tr>
<td></td>
<td>2 Fr. Trans. (mm)</td>
<td>21.79</td>
<td>46.76</td>
<td>160.20</td>
<td>25.08</td>
<td>24.07</td>
<td>44.27</td>
<td>22.89</td>
</tr>
<tr>
<td></td>
<td>Recur. Trans. (mm)</td>
<td>21.98</td>
<td>152.97</td>
<td>26.64</td>
<td>3.64</td>
<td>3.55</td>
<td>14.09</td>
<td>24.48</td>
</tr>
<tr>
<td></td>
<td>2 Fr. Scale</td>
<td>1.07</td>
<td>1.15</td>
<td>0.54</td>
<td>1.10</td>
<td>0.94</td>
<td>0.84</td>
<td>1.14</td>
</tr>
<tr>
<td></td>
<td>Recur. Scale</td>
<td>1.08</td>
<td>1.60</td>
<td>0.94</td>
<td>1.05</td>
<td>1.06</td>
<td>0.98</td>
<td>1.10</td>
</tr>
</tbody>
</table>

Table 4.1. Average 3D error for cone sequence (excluding two points, see text).

<table>
<thead>
<tr>
<th>Iteration</th>
<th>Frame (ft)</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>7.7</td>
<td>6.0</td>
<td>3.5</td>
<td>48.2</td>
<td>45.3</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>44.8</td>
<td>49.7</td>
<td>44.7</td>
<td>31.1</td>
<td>40.9</td>
<td>36.8</td>
</tr>
<tr>
<td>2</td>
<td>4.9</td>
<td>4.4</td>
<td>3.8</td>
<td>29.2</td>
<td>28.8</td>
<td>32.4</td>
</tr>
<tr>
<td>3</td>
<td>4.9</td>
<td>5.0</td>
<td>4.9</td>
<td>27.2</td>
<td>43.0</td>
<td>31.6</td>
</tr>
<tr>
<td>4</td>
<td>36.0</td>
<td>33.5</td>
<td>43.5</td>
<td>25.3</td>
<td>133.3</td>
<td>30.2</td>
</tr>
</tbody>
</table>

Table 4.2. Recovered depths and ground truth in feet for 6 tracked points.

<table>
<thead>
<tr>
<th>Iteration</th>
<th>Frame (ft)</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>38.1</td>
<td>37.0</td>
<td>37.0</td>
<td>38.1</td>
<td>38.4</td>
<td>38.4</td>
</tr>
<tr>
<td>1</td>
<td>36.1</td>
<td>38.0</td>
<td>37.6</td>
<td>36.1</td>
<td>37.5</td>
<td>37.9</td>
</tr>
<tr>
<td>2</td>
<td>34.2</td>
<td>34.8</td>
<td>34.4</td>
<td>34.2</td>
<td>32.0</td>
<td>33.5</td>
</tr>
<tr>
<td>3</td>
<td>32.2</td>
<td>105.0</td>
<td>35.3</td>
<td>32.2</td>
<td>76.2</td>
<td>34.3</td>
</tr>
<tr>
<td>4</td>
<td>30.3</td>
<td>32.3</td>
<td>35.0</td>
<td>30.3</td>
<td>31.3</td>
<td>33.8</td>
</tr>
</tbody>
</table>

Table 4.3. Detected and matched feature points.

Fig. 1. Box sequence

Fig. 2. Cone sequence