The Fusion of Image and Range Flow

J.L. Barron and H. Spies

1 Dept. of Computer Science
University of Western Ontario
London, Ontario, Canada, N6A 5B7
barron@csd.uwo.ca

2 Interdisciplinary Center for Scientific Computing
University of Heidelberg
INF 368, 69121 Heidelberg, Germany
Hagen.Spies@ifw.uni-heidelberg.de

Abstract We present quantitative results for computing local least squares and global regularized range flow using both image and range data. We first review the computation of local least squares range flow and then show how its computation can be cast in a global Horn and Schunck like regularization framework [15]. These computations are done using both range data only and using a combination of image and range data [14]. We present quantitative results for these two least squares range flow algorithms and for the two regularization range flow algorithms for one synthetic range sequence and one real range sequence, where the correct 3D motions are known a priori. We show that using both image and range data produces more accurate and more dense range flow than the use of range flow alone.

Keywords: Range Flow, Image Flow, Regularization, Range Image Sequences, Quantitative Error.

1 Introduction

We can use image sequences to compute optical flow in a local least squares calculation, for example, Lucas and Kanade [8], or in a global iterative regularization, for example, Horn and Schunck [5]. In addition to the use of image intensity data, it is possible to use densely sampled range sequences [17] to compute range flow. Range data (for example from a Biris range sensor [3]) consists of 2D arrays of the 3D coordinates (in millimeters) of a scene, i.e. the 3D X, Y, and Z values, plus the grayscale intensity at those same points. Since our range sensor acquires images under orthographic projection we can only compute image flow (orthographic optical flow) rather than perspective optical flow, although the same algorithms can be used in both cases. Just as optical/image flow can be computed from time-varying image data [1], range flow can be computed from time varying range data [17]. The Biris range sensor is based on active triangulation using a laser beam and on a dual aperture mask. It has a reported depth accuracy of about 0.1mm for objects at a distance of 250mm [3]. This paper investigates the computation of range flow on one synthetic range sequence and on one real range sequence made with a Biris
range sensor using regularization on both range and/or intensity derivatives. We also show how local and global optical flow computations can be extended into 3D, allowing the calculation of dense accurate range flow fields, often when the number of individual range velocities is sparse.

Although the work reported here was performed with Biris range sensor data, there is no reason why our algorithms could not be used with other sources of time-varying depth information, such as depth maps from stereo [16] or motion and structure [6] algorithms. Here we assume locally rigid objects (although both of our sequences have globally rigid objects). Instead of computing camera motion parameters and overall scene motion, we are interested in computing the range flow, i.e. the 3D velocity, at each point the depth data is sampled at. Towards this end, we start with the range constraint equation of [6, 16, 17].

2D optical flow methods have recently been generalized into the 3D domain. Chaudhury et al. [4] formulated at 3D optical flow constraint, using $I_x$, $I_y$, $I_z$ and $I_t$ derivatives. Thus they have a time-varying volume of intensity where all 4 derivatives can be computed. A lot of this work has been medically motivated, for example, to compute 3D flow for CT, MRI and PET datasets [11, 12, 18, 7]. Since range flow is computed with respect to a moving 3D surface, derivative data in the Z dimension is not available, resulting in different constraint equations for range data and for 3D optical flow.

The basic algorithms used in this paper have been reported in more detail elsewhere:

1. Quantitative flow analysis using the Lucas and Kanade least squares calculation and the Horn and Schunck regularization were reported in [1].
2. The computation of full range flow (and its two types of normal flow) in a total least squares framework was reported in [13]. Here, the range flow calculation is reformulated in a least squares framework [2].
3. The direct regularization was presented in [15] for a number of different sequences, including a real sequence made from the 3D motion of a growing caster oil bean leaf using a Biris range sensor.
4. The computation of range flow from both intensity and range data in both a total least squares framework (as opposed to a least squares framework used here) and a regularization framework is reported in [14].

We examine the quantitative performance of these algorithms on two intensity/range sequences: a synthetic sequence where the range and intensity data was error-free, yielding good flows and a real sequence where both the range and intensity data are poor. In the later case, we also know the true 3D velocity and are thus able to quantitatively analyze the flow. The results are quite good when the combined intensity and range data are taken into account, especially when one considers that the range structure is very poor at most locations (the surfaces are planar).
2 2D Image Flow

The well known motion constraint equation:

\[ I_x u + I_y v + I_t = 0 \]  \hspace{1cm} (1)

forms the basis of most optical flow algorithms. \( I_x, I_y \) and \( I_t \) in equation (1) are the \( x, y \) and \( t \) intensity derivatives while \( \mathbf{v} = (u, v) \) is the image velocity (or optical flow) at pixel \((x, y)\), which is an approximation of the local image motion. Equation (1) is 1 equation in 2 unknowns and manifests the aperture problem. Raw normal velocity (the component of image velocity normal to the local intensity structure) can be totally expressed in terms of derivative information:

\[ v_{tn} = \frac{-I_t(I_x I_y)}{I_x^2 + I_y^2} \]  \hspace{1cm} (2)

while tangential velocity, \( v_t \) cannot, in general, be recovered.

To solve for \( \mathbf{v} \) we need to impose an additional constraint. An example of a local constraint is to assume that locally all image velocities are the same. For example, Lucas and Kanade [8] use a least squares computation to integrate local neighbourhoods of normal image velocities into full image velocities. For a \( n \times n \) neighbourhood, they solve a \( n \times 2 \) linear system of equations \( A_{n \times 2} \mathbf{v} = B_{n \times 1} \) as

\[ \mathbf{v} = (A^T A)^{-1} A^T B, \] \hspace{1cm} (3)

where \( A \) has entries \( I_{xi} \) and \( I_{yi} \) in the \( i \)th row and \( B \) has entries \(-I_{ti}\) in the \( i \)th row. We perform eigenvector/eigenvalue analysis on \( A^T A \) using routines in [9].

Eigenvalue \( \lambda_0 \leq \lambda_1 \) and corresponding eigenvector \((\mathbf{e}_0 \text{ and } \mathbf{e}_1)\) decomposition of the symmetric matrix \( A^T A \) yields least squares full image velocity, if both \( \lambda_0 \geq \tau_{D1} \) and \( \lambda_1 \geq \tau_{D1} \), or least squares normal image velocity, \( v_{tn} = \mathbf{v} \cdot \mathbf{e}_1 \), if \( \lambda_1 \geq \tau_{D1} \) but \( \lambda_0 \leq \tau_{D1} \). On the other hand, Horn and Schunck [3] impose a global smoothness constraint on the optical flow field and minimize:

\[ \int \int (I_x u + I_y v + I_t)^2 + \alpha^2 (u_x^2 + u_y^2 + v_x^2 + v_y^2) \partial x \partial y. \] \hspace{1cm} (4)

We can minimize this functional using Euler-Lagrange equations (with \( \nabla^2 u \) and \( \nabla^2 v \) approximated as \( \bar{u} = u \) and \( \bar{v} = v \) respectively) as:

\[
\begin{bmatrix}
\alpha^2 + I_x^2 & I_x I_y \\
I_x I_y & \alpha^2 + I_y^2 \\
\end{bmatrix}
\begin{bmatrix}
u \\
\end{bmatrix}
= \begin{bmatrix}
\alpha^2 \bar{u} - I_x I_t \\
\alpha^2 \bar{v} - I_y I_t \\
\end{bmatrix},
\] \hspace{1cm} (5)

yielding the Gauss-Seidel iterative equations:

\[
\begin{bmatrix}
u^{n+1} \\
v^{n+1} \\
\end{bmatrix}
= A^{-1}
\begin{bmatrix}
\alpha^2 \overline{u}^{n} - I_x I_t \\
\alpha^2 \overline{v}^{n} - I_y I_t \\
\end{bmatrix},
\] \hspace{1cm} (6)
3 3D Range Flow

Biris range data consists not only of 3D coordinate \((X,Y,Z)\) data of an environmental scene but also intensity data for each of those environmental points. The motion constraint equation can easily be extended into the range constraint equation [17] in 3D:

\[
Z_X U + Z_Y V + W + Z_t = 0, \tag{7}
\]

where \(V = (U,V,W)\) is the 3D range velocity and \(Z_X, Z_Y\) and \(Z_t\) are spatio-temporal derivatives of the depth coordinate \(Z\). Raw normal velocity can also be computed directly from \(Z\) derivatives as

\[
V_{nn} = \frac{-Z_t(Z_X, Z_Y, 1)}{Z_X^2 + Z_Y^2 + 1}. \tag{8}
\]

For a \(n \times n\) neighbourhood, we can solve a \(n \times 3\) linear system of equations \(A_{n \times 3} V = B_{n \times 1}\) as

\[
V = (A^T A)^{-1} A^T B, \tag{9}
\]

where \(A\) has entries \(Z_{Xi}, Z_{Yi}\) and 1 in the \(i^{th}\) row and \(B\) has entries \(-Z_{ti}\) in the \(i^{th}\) row. Alternatively to this least squares computation a total least squares approach may be used [13]. The eigenvalues \((\lambda_0 \leq \lambda_1 \leq \lambda_2)\) and their corresponding eigenvectors \((\hat{e}_0, \hat{e}_1\) and \(\hat{e}_2)\) can be computed from the \(3 \times 3\) symmetric matrix \(A^T A\) and then used to compute least squares full range velocity, \(V_F\), when \(\lambda_0, \lambda_1, \lambda_2 > \tau_{D2}\), an estimate of least squares line normal velocity, \(V_L\), when \(\lambda_1, \lambda_2 > \tau_{D2}\), \(\lambda_0 \leq \tau_{D2}\) and an estimate of the least squares plane normal velocity, \(V_P\), when \(\lambda_2 > \tau_{D2}\), \(\lambda_0, \lambda_1 \leq \tau_{D2}\). The terms line and plane normal velocity are motivated by the fact that these types of normal velocity always occur on lines or planes on the 3D surface. That is:

\[
V_F = (V \cdot \hat{e}_0) \hat{e}_0 + (V \cdot \hat{e}_1) \hat{e}_1 + (V \cdot \hat{e}_2) \hat{e}_2 \tag{10}
\]

\[
V_L = (V \cdot \hat{e}_1) \hat{e}_1 + (V \cdot \hat{e}_2) \hat{e}_2 \tag{11}
\]

\[
V_P = (V \cdot \hat{e}_2) \hat{e}_2. \tag{12}
\]

Of course \(V\) is \(V_F\). This computational scheme breaks down if \(A^T A\) cannot be reliably inverted as then we cannot compute \(V\) as required in equations (10) to (12). Below we outline how to compute line and planar normal flow when \(A^T A\) is nearly singular. We can rewrite the eigenvalue/eigenvector equation, \(A^T \hat{e}_i = \lambda_i \hat{e}_i\), as

\[
A^T A \begin{bmatrix} \hat{e}_0 \\ \hat{e}_1 \\ \hat{e}_2 \end{bmatrix} = A^T A R = \begin{bmatrix} \lambda_0 & 0 & 0 \\ 0 & \lambda_1 & 0 \\ 0 & 0 & \lambda_2 \end{bmatrix} R, \tag{13}
\]

where \(R = [\hat{e}_0, \hat{e}_1, \hat{e}_2]^T\). Thus we can rewrite (9) as:

\[
\begin{bmatrix} \lambda_0 & 0 & 0 \\ 0 & \lambda_1 & 0 \\ 0 & 0 & \lambda_2 \end{bmatrix} V' = B', \tag{14}
\]
where \( \mathbf{V} = (U, V, W) = R^T \mathbf{V} \) and \( B^T = (b_0, b_1, b_2)^T = R^T A^T B \). If \( \lambda_0 \) is small, \( \lambda_0 \leq \tau_{D2}, \lambda_1, \lambda_2 > \tau_{D2} \), we are dealing with a line normal velocity, \( \mathbf{V}_L = (U_L, V_L, W_L) \). Then the 2\(^{nd}\) and 3\(^{rd}\) equations of (14) give two equations that define constraint planes that the normal velocity must lie in. The line normal is given by the point on their intersecting line with minimal distance from the origin. The direction of this line is given by \( \mathbf{e}_0 = \mathbf{e}_1 \times \mathbf{e}_2 \), which yields a third equation. The system of equations to be solved is:

\[
\begin{align*}
V_L &= e_{10} U_L + e_{11} V_L + e_{12} W_L = \frac{b_0}{\lambda_1} \\
W_L &= e_{21} U_L + e_{21} V_L + e_{22} W_L = \frac{b_2}{\lambda_2} \\
e_{01} U_L + e_{01} V_L + e_{02} W_L &= 0.
\end{align*}
\]

If both \( \lambda_0 \) and \( \lambda_1 \) are less than \( \tau_{D2} \) then we can only compute planar normal range flow. In this case, we have one constraint:

\[
e_{20} U_P + e_{21} V_P + e_{22} W_P = \frac{b_2}{\lambda_2}.
\]

The plane normal flow is the point on this plane with minimal distance from the origin:

\[
\mathbf{V}_P = \frac{\frac{b_2}{\lambda_2}}{e_{20} + e_{21} + e_{22}} \begin{bmatrix} e_{20} \\ e_{21} \\ e_{22} \end{bmatrix} = \frac{b_2}{\lambda_2} \begin{bmatrix} e_{20} \\ e_{21} \\ e_{22} \end{bmatrix}.
\]

Since \( A^T A \) is a real, positive semi-definite, symmetric matrix, eigenvalue/eigenvector decomposition always yields real positive eigenvalues.

### 4 Least Squares Image-Range Flow

We note that if we compute derivatives of intensity with respect to \( X \) and \( Y \), rather than \( x \) and \( y \) (the projection of \( X \) and \( Y \) on the sensor grid) the motion constraint equation becomes:

\[
I_X U + I_Y V + I_t = 0,
\]

where \( U \) and \( V \) are the first two components of range flow. Since a Biris sensor's images are made under orthographic projection we use standard optical flow as image flow, \((U, V)\) can then be recovered by a least squares calculation. They are the first two components of range flow and are orthographic image velocity (which we call image flow). If we use equations of the form in (20) and (7) whenever intensity and/or depth derivatives reliably available, we obtain a least squares linear system of equations for \( U, V \) and \( W \) in terms of the spatio-temporal intensity and depth derivatives. We require at least one equation of the form in equation (7) be present to constrain the \( W \) parameter. We use \( \beta \) to weigh the contribution of the depth and intensity derivatives in the computation of \((U, V, W)\) so that they are of equal influence. We solve for \((U, V, W)\) using least squares as outlined above, checking the eigenvalues against a third threshold, \( \tau_{D3} \).
5 Direct Regularized Range Flow

We can compute regularized range flow directly using the spatio-temporal derivatives of $Z$ by minimizing

$$
\int \int \int (Z_{XX}U + Z_{XY}V + W + Z_t)^2 + \alpha^2 (U^2_X + U^2_Y + U^2_t + V^2_X + V^2_Y + V^2_t + W^2_X + W^2_Y + W^2_t + 2U_{XX}W_{XY} + 2U_{XY}W_{Xt} + 2U_{XT}W_{Yt} + 2V_{XX}W_{XY} + 2V_{XY}W_{Xt} + 2V_{XT}W_{Yt}) + \alpha^2 (U^2_X + U^2_Y + U^2_t + V^2_X + V^2_Y + V^2_t + W^2_X + W^2_Y + W^2_t + 2U_{XX}W_{XY} + 2U_{XY}W_{Xt} + 2U_{XT}W_{Yt} + 2V_{XX}W_{XY} + 2V_{XY}W_{Xt} + 2V_{XT}W_{Yt}) \frac{\partial X}{\partial x} \frac{\partial Y}{\partial y} \frac{\partial Z}{\partial t}.
$$

We can write the Euler-Lagrange equations using the approximations $\nabla^2 U = U_{XX} + U_{XY} + U_{ZZ} \approx \bar{U} - U$, $\nabla^2 V = V_{XX} + V_{XY} + V_{ZZ} \approx \bar{V} - V$ and and $\nabla^2 W = W_{XX} + W_{XY} + W_{ZZ} \approx \bar{W} - W$ respectively as:

$$
\begin{bmatrix}
U \\
V \\
W
\end{bmatrix} = \begin{bmatrix}
(\alpha^2 U_{XX} - Z_X Z_t) \\
(\alpha^2 V_{XY} - Z_Y Z_t) \\
(\alpha^2 W_{Xt} - Z_t)
\end{bmatrix}.
$$

(22)

The Gauss-Seidel equations then become:

$$
\begin{bmatrix}
U^{n+1} \\
V^{n+1} \\
W^{n+1}
\end{bmatrix} = A^{-1} \begin{bmatrix}
(\alpha^2 U^n_{XX} - Z_X Z_t) \\
(\alpha^2 V^n_{XY} - Z_Y Z_t) \\
(\alpha^2 W^n_{Xt} - Z_t)
\end{bmatrix}.
$$

(23)

6 Combined Range Flow from Intensity and Range Derivatives

It is possible to compute $V$ using both intensity and range derivatives via equations (1) and (7) and the same smoothness term given in equation (21). We regularize:

$$
\int \int \int (Z_{XX}U + Z_{XY}V + W + Z_t)^2 + \beta (I_{XX}U + I_{XY}V + I_t)^2
$$

$$
+ \alpha^2 (U^2_X + U^2_Y + U^2_t + V^2_X + V^2_Y + V^2_t + W^2_X + W^2_Y + W^2_t + 2U_{XX}W_{XY} + 2U_{XY}W_{Xt} + 2U_{XT}W_{Yt} + 2V_{XX}W_{XY} + 2V_{XY}W_{Xt} + 2V_{XT}W_{Yt}) + \alpha^2 (U^2_X + U^2_Y + U^2_t + V^2_X + V^2_Y + V^2_t + W^2_X + W^2_Y + W^2_t + 2U_{XX}W_{XY} + 2U_{XY}W_{Xt} + 2U_{XT}W_{Yt} + 2V_{XX}W_{XY} + 2V_{XY}W_{Xt} + 2V_{XT}W_{Yt}) \frac{\partial X}{\partial x} \frac{\partial Y}{\partial y} \frac{\partial Z}{\partial t},
$$

(24)

The Euler-Lagrange equations are

$$
A \begin{bmatrix}
U \\
V \\
W
\end{bmatrix} = \begin{bmatrix}
(\alpha^2 U_{XX} - Z_X Z_t - \beta I_{XX} I_t) \\
(\alpha^2 V_{XY} - Z_Y Z_t - \beta I_{XY} I_t) \\
(\alpha^2 W_{Xt} - Z_t - \beta I_t)
\end{bmatrix},
$$

(25)

where $A$ is

$$
\begin{bmatrix}
Z_X^2 + \beta I_X^2 + \alpha^2 Z_X Z_Y + \beta I_X I_Y & Z_X & \\
Z_X Z_Y + \beta I_X I_Y & Z_Y^2 + \beta I_Y^2 + \alpha^2 & Z_Y \\
Z_X & Z_Y & 1 + \alpha^2
\end{bmatrix}.
$$

(26)
The Gauss-Seidel equations are then

\[
\begin{bmatrix}
U^{n+1} \\
V^{n+1} \\
W^{n+1}
\end{bmatrix}
= A^{-1} \begin{bmatrix}
\alpha^2 U^n - Z_X Z_t - \beta^2 I_X I_t \\
\alpha^2 V^n - Z_Y Z_t - \beta^2 I_Y I_t \\
\alpha^2 W^n - Z_t - \beta^2 I_t
\end{bmatrix}.
\]  \hspace{1cm} (27)

The matrix \(A^{-1}\) only has to be computed once in equations (23) and (27), existence of the inverse is guaranteed by the Sherman-Morrison-Woodbury formula [15].

7 Differentiation

The use of a good differential kernel is essential to the accuracy of both image and range flow calculations. We use the balanced/matched filters for prefiltering and differentiation proposed by Simoncelli [10]. A simple averaging filter \([\frac{1}{3}, \frac{1}{3}, \frac{1}{3}]\) was used to slightly blur the images before prefiltering/differentiation. The prefiltering kernel’s coefficients were \((0.0356976, 0.2488746, 0.4308557, 0.2488746\) and \(0.0356976)\) while the differential kernel’s coefficients were \((-0.107663, -0.282671, 0.0, 0.282671\) and \(-0.107663)\). For example, to compute \(I_p\), we first convolve the prefiltering kernel in the \(t\) dimension, then convolve the prefiltering kernel on that result in the \(y\) dimension and finally convolve the differentiation kernel in the \(x\) dimension on that result. We assume a uniform sampling of the \(z\) data in \(X\) and \(Y\); in general, this is not true (but is true for our data).

8 Error Measurement

We report 2D error for image flow and 3D error for range flow using relative magnitude error (as a percentage) and angle error (in degrees). \(\hat{V}_e\) is the correct image/range flow and \(\hat{V}_e\) is the estimated or computed image/range flow in the equations below. For magnitude error we report:

\[
\psi_M = \frac{||\hat{V}_e|| - ||\hat{V}_e||}{||\hat{V}_e||} \times 100%,
\]  \hspace{1cm} (28)

while for angle error we report:

\[
\psi_A = \arccos(\hat{V}_e \cdot \hat{V}_e).
\]  \hspace{1cm} (29)

For line normal range flow we compute an estimated correct line flow as:

\[
\hat{V}_{Lc} = (\hat{V}_c \cdot \hat{e}_1) \hat{e}_1 + (\hat{V}_c \cdot \hat{e}_2) \hat{e}_2.
\]  \hspace{1cm} (30)

Of course \(\hat{e}_1\) and \(\hat{e}_2\) have error in themselves as they are computed from the least squares integration matrix. We then report magnitude and angle error as given in equations (28) and (29). Finally, for planar normal range flow we can only compute the planar magnitude error:

\[
\psi_{P3D} = \frac{|V_c \cdot \hat{V}_P - ||V_P|||}{||V_P||} \times 100%,
\]  \hspace{1cm} (31)
as the direction of the computed and estimated correct plane flow are always the same (the direction of the eigenvector corresponding to the largest eigenvalue).

We also examine $\phi_{abs}$, the average absolute error:

$$\phi_{abs} = \sum_{i=1}^{N} ||V_{e} \cdot \hat{V}_{F}||_{2} - ||V_{F}||_{2}$$  \hspace{1cm} (32)

9 Synthetic Range Flow Results

To test our algorithms, we made a synthetic range sequence where we know the correct 3D translation (0.4,0.6,0.9 units/frame), allowing quantitative error analysis. In Figure 1a we show the depth map synthetically generated while Figure 1b shows the corresponding image data. Each line in the depth map has a Gaussian profile - this is made by simply rotating the coordinates into the line and then applying the appropriate exponential function. The motion in Z is done afterwards by simply adding the appropriate motion, thus $W$ is globally constant for this sequence. The image data was made by simply overlaying two sinusoids with perpendicular orientations with the correct XY motion, thus $V = (U,V,W)$ is globally constant.

![Fig. 1. Synthetic depth map without texture and (b) a sinusoid texture.](image)

Figure 2a through 2e show the computed XY and XZ full, line and planar range flow for this sequence (section 3) while Table 1 gives their quantitative magnitude (percentage) and direction (angle) error measures. We use the projected correct flow in the direction of computed eigenvectors as “correct” line and plane flow. These are good estimates of correct plane range flow but not so good for line flow.

Figures 3a,b shows the computed Horn and Schunck and Lucas and Kanade flow fields (section 2) for the two image sequences while Table 2 shows their quantitative error. The local least squares image-range flow results (section 4) are
\textbf{Fig. 2.} The computed XY and XZ components of full ((a) and (b)), line ((c) and (d)) and planar ((e) and (f)) range flow for the synthetic sequence.
**Fig. 3.** The image flow computed using (a) Horn and Schunck method (1000 iterations) and (b) Lucas and Kanade's method \((\gamma_2 = 1.0)\) for the synthetic sequence. Flows (c) and (d) show the XY and XZ components of range flow for the synthetic sequence computed using the least squares image-range flow calculation.

also shown in Table 2 and Figures 3c,d. These range flow results are quite good, better than Horn and Schunck image flow. This is quite remarkable, considering that we are computing 100% dense 3D range flow (compared with 100% dense 2D image flow). Table 3 shows the magnitude and angle results for the Direct (section 5) and Combined (section 6) regularization methods for 1000 iterations. Results for the Combined regularization are the best (but not as good as the Least Squares optical-range calculation). These results indicate that using both image and range data is the best way to recover accurate 3D velocity fields. We used \(\alpha = \beta = 1.0\) for all regularizations.
10 Real Range Flow Results

We also have one real range sequence which we made in 1997 at NRC in Ottawa\(^1\). Each image of this sequence is 454×1024 and was made by moving a scene (consisting of some boxes wrapped in newspaper) a set of fixed equal distances on a linear positioner and after each movement, taking intensity and range images. Thus, the correct 3D translation (0.095377, 1.424751, 0.087113) mm/frame is known, allowing quantitative error analysis. NRC's Biris range sensor was also mounted on another linear positioner and at each time three sets of four overlapping (intensity and X, Y and Z) images were acquired. These images are then manually viewed and joined into one larger image (some partially overlaid data was discarded). A sheet of white paper was also imaged and used to correct the

\(^1\) Thanks to Luc Cournoyer at NRC for helping us make this data.
Fig. 5. (a) The smoothed subsampled intensity image for frame 25 of the NRC sequence and (b) its corresponding depth (Z) image.

<table>
<thead>
<tr>
<th>Full Range Velocity ($\tau_{D2} = 0.2$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\phi_M$</td>
</tr>
<tr>
<td>$\phi_A$</td>
</tr>
<tr>
<td>Density</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Line Normal Range Velocity ($\tau_{D2} = 0.2$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\phi_M$</td>
</tr>
<tr>
<td>$\phi_A$</td>
</tr>
<tr>
<td>Density</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Plane Normal Range Velocity ($\tau_{D2} = 0.2$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\phi_{FD}$</td>
</tr>
<tr>
<td>$\phi_{SFD}$</td>
</tr>
<tr>
<td>Density</td>
</tr>
</tbody>
</table>

Table 1. Direction and magnitude error of the computed full, line and plane range velocities wrt the estimated "correct" full, line and range flow for the synthetic range sequence.

intensity images by rescaling their intensities so that all intensities were white and then rescaling the acquired images by these same factors.

In retrospect, if we were to make these images again, we would not use only planar surfaces, as only plane range flow can be recovered there. Sparse
Fig. 6. The computed XY components of full and line range flow for the NRC sequence.

<table>
<thead>
<tr>
<th></th>
<th>Horn and Schunck XY Flow (1000 iterations)</th>
<th>Lucas and Kanade XY Flow ($\tau_{D1} = 1.0$)</th>
<th>Least Squares Image-Range 3D Flow ($\tau_{D2} = 0.2$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\phi_M$</td>
<td>0.27% ± 0.88%</td>
<td>0.0004% ± 0.0006%</td>
<td>0.048% ± 0.005%</td>
</tr>
<tr>
<td>$\phi_A$</td>
<td>0.07° ± 0.24°</td>
<td>0.0057° ± 0.0113°</td>
<td>0.007° ± 0.012°</td>
</tr>
<tr>
<td>Density</td>
<td>100%</td>
<td>81.86%</td>
<td>100.0%</td>
</tr>
</tbody>
</table>

Table 2. Direction and magnitude error of the computed Horn and Schunck image flow (for 1000 iterations), Lucas and Kanade image flow (for $\tau_{D1} = 1.0$) and for the 3D range flow computed via the least squares optical-range flow algorithm (for $\tau_{D2} = 0.2$) for the synthetic range sequence.

full and line flow can be recovered, but only at the boxes' corners and edges. Nevertheless, we were able to compute some meaningful and dense full range flow fields using our regularization algorithms. To attenuate the effects of noise artifacts and to improve computational time we used level 1 of the Gaussian
Fig. 7. The image flow computed using (a) Horn and Schunck method (1000 iterations) and (b) Lucas and Kanade’s method ($\gamma_{D1} = 1.0$) on the NRC sequence. Flows (c) and (d) show the XY components of range flow computed using the direct image-range flow calculation and direct regularization.
Fig. 8. (a) Combined regularization XY flow for 1000 iterations and (b) direct regularization with 1000 iterations initialized with combined regularization with 1000 iterations regularization for the NRC sequence.

<table>
<thead>
<tr>
<th>Direct Regularization (1000 iterations)</th>
<th>Combined Regularization (1000 iterations)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\phi_M$ 7.26% ± 8.06%</td>
<td>$\phi_M$ 1.45% ± 3.44%</td>
</tr>
<tr>
<td>$\phi_A$ 6.69° ± 8.34°</td>
<td>$\phi_A$ 0.83° ± 1.96°</td>
</tr>
</tbody>
</table>

Table 3. Direction and magnitude error of the computed flow via the Direct and Combined regularization algorithms for 1000 iterations for the synthetic sequence.

...
values here and to increase computational accuracy and speed we masked out these parts of the image in our flow calculations. To show fully recovered range flow fields we need to show both XY and XZ flow fields; however since the X and Z flow components are only about 6% of the Y component for this sequence, the XZ flows are quite small relative to the XY flows and due to space limitations are not shown here. Figure 6 shows the computed XY full and line range flows (section 3). The plane flows are quite small and not shown here. Table 4 give the quantitative results for these full, line normal and plane normal fields. Because the plane normal flow is so small we just give its absolute error.

Figures 7a,b shows the image flows recovered by Horn and Schunck’s algorithm (1000 iterations) and Lucas and Kanade’s algorithm (\(\tau_{D_2} = 1.0\)) (section (2)) while Figures 7c shows the XY flow using our least squares computation on the intensity and range derivatives. Table 5 show the magnitude and direction errors for these flows.

<table>
<thead>
<tr>
<th>Full Range Flow ((\tau_{D_2} = 0.2))</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\psi_M)</td>
</tr>
<tr>
<td>(\psi_A)</td>
</tr>
<tr>
<td>Density</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Line Normal Range Flow ((\tau_{D_2} = 0.2))</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\phi_M)</td>
</tr>
<tr>
<td>(\phi_A)</td>
</tr>
<tr>
<td>Density</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Plane Normal Range Flow ((\tau_{D_2} = 0.2))</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\phi_{abs})</td>
</tr>
<tr>
<td>Density</td>
</tr>
</tbody>
</table>

Table 4. Direction and magnitude error of the computed full, line and plane range velocities wrt the estimated “correct” full, line and range flow for the NRC real range sequence.

Figures 7d and 8a show the regularized XY range flow fields for the direct (section (5)) and combined (section (6)) algorithms for 1000 iterations while Table 6 shows their magnitude and angle errors. We used \(\alpha = 10.0\) and \(\beta = 1.0\) for all the regularizations. For direct regularization, overall results are poor because most of the image only has plane flow information, the regions surrounding full flow have good velocities. Results improve with more iterations. The combined regularized flows are the best, these use both intensity and range derivative data and yield dense flow. We report one last experiment: we use the flow after 1000 iterations of the combined regularization algorithm to initialize the direct regularization algorithm (also 1000 iterations). The flow is shown in Figure 8b and the error in Table 6. 71.75\% of the flow had 10\% or less magnitude error (average magnitude error of 4.29\% ± 2.51\% and average angle error of 0.24\° ± 1.64\°). This was the best result of all the NRC flows. This use of an initial set of non-zero ve-
Table 5. Direction and magnitude error of the computed Horn and Schunck image flow (for 1000 iterations), Lucas and Kanade image flow (for $\tau_{D1} = 1.0$) and for the 3D range flow computed via the least squares optical-range flow algorithm (for $\tau_{D2} = 0.2$) for the NRC range sequence.

<table>
<thead>
<tr>
<th></th>
<th>Horn and Schunck XY Flow (1000 iterations)</th>
<th>Lucas and Kanade XY Flow ($\tau_{D1} = 1.0$)</th>
<th>Least Squares Image-Range 3D Flow ($\tau_{D2} = 0.2$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\phi_M$</td>
<td>10.33% ± 12.47%</td>
<td>10.51% ± 10.07%</td>
<td>13.80% ± 12.50%</td>
</tr>
<tr>
<td>$\phi_A$</td>
<td>3.48° ± 5.53°</td>
<td>9.68° ± 5.57°</td>
<td>14.04° ± 5.94°</td>
</tr>
<tr>
<td>Density</td>
<td>82.67%</td>
<td>8.11%</td>
<td>65.25%</td>
</tr>
</tbody>
</table>

Locities in the initialization step of regularization seems to be one way to obtain dense accurate flow for the NRC sequence.

Table 6. Direction and magnitude error of the direct and combined regularized flow for 1000 iterations for the NRC sequence. Also shown are the error results when the combined regularized flow is used to initialize the direct regularization. The density of all flow fields (due to masking) is 82.68%.

<table>
<thead>
<tr>
<th></th>
<th>Direct Regularization (1000 iterations)</th>
<th>Combined Regularization (1000 iterations)</th>
<th>Direct Regularization (1000 iterations) initialized by Combined Regularization (1000 iterations)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\phi_M$</td>
<td>39.97% ± 24.52%</td>
<td>15.46% ± 20.06%</td>
<td>9.76% ± 9.19%</td>
</tr>
<tr>
<td>$\phi_A$</td>
<td>7.84° ± 3.97°</td>
<td>16.50° ± 13.08°</td>
<td>5.88° ± 2.97°</td>
</tr>
</tbody>
</table>

11 Conclusions

We have shown the computation of full, line normal and plane normal range flow on a synthetic intensity/range sequence. Our computation was in a least squares framework [2]; total least squares is used in [13, 15, 14] and we are cur-
rently investigating the difference. Line normal flow was the most difficult flow to compute accurately for this sequence.

The NRC sequence is perhaps the most difficult type of range sequence to analyze; most of the surfaces are planar with little or no full or line normal velocity. The direct regularization algorithm were only able to compute full flow in the vicinity of this full and line normal flow. The combined regularization used both intensity and range data to obtain full flow everywhere. The usefulness of combining the two types of data should not be in doubt; its flow was better than that with the use of range data alone and, of course, image flow, by itself cannot be used to recover the 3rd component of range flow. When we initialized direct regularization with combined regularized flow, we obtain the best results.

Acknowledgements

We gratefully acknowledge financial support from NSERC (National Science and Engineering Research Council of Canada) and the DFG (Deutsche Forschungsgemeinschaft) research unit "Image Sequence Analysis to Investigate Dynamic Processes".

References


