Quantitative Regularized Range Flow

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

H. Spies
Interdisciplinary Center for Scientific Computing
University of Heidelberg
INF 368, 69120 Heidelberg, Germany
Hagen.Spies@iwr.uni-heidelberg.de

Abstract We present quantitative results for computing local least squares and global regularized range flow on a real range sequence. We review the computation of local range flow [11], including the two types of normal range flow, computed in a least squares framework, and then show how its computation can be cast in a global Horn and Schunck like regularization framework [13]. This is done by using range data only and by using a combination of image and range data [12]. We present quantitative results for three regularization algorithms and a least squares image-range algorithm for one real range sequence.

1 Introduction

We can use image sequences to compute optical flow in a local least squares framework, for example, Lucas and Kanade [6], or in a global iterative regularization, for example, Horn and Schunck [4]. In addition to the use of image intensity data, it is possible to use densely sampled range sequences [14] to compute **range flow**, the 3D velocity of environmental points relative to the sensor. Range data (from a Biris range sensor [2]) consists of 2D arrays of the 3D coordinates (in millimeters) of a scene, i.e. the 3D X, Y and Z values of each pixel, plus the grayvalue intensity at those points. Since the range sensor we used 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. The Biris range sensor is based on active triangularization 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 [2]. This paper investigates the computation of range flow on a real range sequence made with a Biris range sensor using regularization algorithms on both range or intensity/range derivatives. We 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.

2D optical flow methods have recently been generalized into the 3D domain. Chaudhury et al. [3] formulated a 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 [9, 10, 15, 5]. Since range flow is computed with respect to a moving 3D surface rather than moving 3D volumetric data, derivative data in the Z dimension is not available, resulting in slightly different constraint equations for range flow 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 [11]. Here, we reformulate the range flow calculation in a least squares framework.

3. The direct and indirect regularizations are first presented in [13] for a number of different sequences, including a real sequence made from the 3D motion of a growing castor oil bean leaf using a different 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 [12].

We examine the quantitative performance of all of these algorithms on a real sequence where both the range and intensity data are poor. We know the true 3D velocity and are thus able to quantitatively analyze the flow. The results are quite good, especially when one

considers that the range structure is uniform at most locations (the surfaces are planar).

2 2D Image Flow

The well known motion constraint equation:

$$I_x u + I_y v + I_t = 0 \quad (1)$$

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 $\vec{v} = (u, v)$ 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. Normal velocity (the component of image velocity normal to the local intensity structure can be totally expressed in terms of derivative information:

$$\vec{v}_{rn} = \frac{-I_t(I_x, I_y)^T}{I_x^2 + I_y^2} \quad (2)$$

while tangential velocity, $\vec{v}_t$ cannot, in general, be recovered.

To solve for $\vec{v}$ 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 [6] 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 $A_{n\times 2}\vec{v} = B_{n\times 1}$ as

$$\vec{v} = (A^T A)^{-1} A^T B, \quad (3)$$

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 [7]. Eigenvalue ($\lambda_0 \leq \lambda_1$) and corresponding eigenvector ($\hat{e}_0$ and $\hat{e}_1$) decomposition of the symmetric matrix $A^T A$ yields least squares full image velocity, if both $\lambda_0 > \tau_{D1}$ and $\lambda_1 > \tau_{D1}$, or least squares normal image velocity, $\vec{v_{ln}} = \vec{v} \cdot \hat{e}_1$, if $\lambda_1 > \tau_{D1}$ but $\lambda_0 \leq \tau_{D1}$. On the other hand, Horn and Schunck [4] impose a local smoothness constraint on the optical flow field and minimize:

$$\int\int (I_x u + I_y v + I_t)^2 + \alpha^2(u_x^2 + u_u^2 + v_x^2 + v_y^2)\partial x \partial y. \quad (4)$$

We can write the Euler-Lagrange equations (with $\nabla^2 u$ and $\nabla^2 v$ approximated as $\bar{u} - u$ and $\bar{v} - v$ respectively) for Horn and Schunck as:

$$\underbrace{\begin{bmatrix} (\alpha^2 + I_x^2) & I_x I_y \\ I_x I_y & (\alpha^2 + I_y^2) \end{bmatrix}}_{A} \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} (\alpha^2 \bar{u} - I_x I_t) \\ (\alpha^2 \bar{v} - I_y I_t) \end{bmatrix}, \quad (5)$$

yielding the Gauss Seidel iterative equations:

$$\begin{bmatrix} u^{n+1} \\ v^{n+1} \end{bmatrix} = A^{-1} \begin{bmatrix} (\alpha^2 u^n - I_x I_t) \\ (\alpha^2 v^n - I_y I_t) \end{bmatrix}. \quad (6)$$

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 [14] in 3D:

$$Z_X U + Z_Y V + W + Z_t = 0, \quad (7)$$

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

$$\vec{V}_{rn} = \frac{-Z_t(Z_X, Z_Y, 1)^T}{Z_X^2 + Z_Y^2 + 1}. \quad (8)$$

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

$$\vec{V} = (A^T A)^{-1} A^T B, \quad (9)$$

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. A total least squares approach was also used [11]. 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 3×3 symmetric matrix $A^T A$ and then used to compute least squares full range velocity, $\vec{V}_F$, when $\lambda_0, \lambda_1, \lambda_2 > \tau_{D2}$, an estimate of least squares line normal velocity, $\vec{V}_L$, when $\lambda_1, \lambda_2 > \tau_{D2}$, $\lambda_0 \leq \tau_{D2}$ and an estimate of the least squares plane normal velocity, $\vec{V}_P$, when $\lambda_2 > \tau_{D2}$, $\lambda_0, \lambda_1 \leq \tau_{D2}$. The terms line and plane normal range velocity are motivated by the fact that these types of normal velocity always occur on lines or planes on the 3D surface. That is:

$$\vec{V}_F = (\vec{V} \cdot \hat{e}_0)\hat{e}_0 + (\vec{V} \cdot \hat{e}_1)\hat{e}_1 + (\vec{V} \cdot \hat{e}_2)\hat{e}_2 \quad (10)$$

$$\vec{V}_L = (\vec{V} \cdot \hat{e}_1)\hat{e}_1 + (\vec{V} \cdot \hat{e}_2)\hat{e}_2 \quad (11)$$

$$\vec{V}_P = (\vec{V} \cdot \hat{e}_2)\hat{e}_2. \quad (12)$$

Of course $\vec{V}$ is $\vec{V}_F$. This computational scheme breaks down if $A^T A$ cannot be reliably inverted as then we cannot compute $\vec{V}$ 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, $A^T A \hat{e}_i = \lambda_i \hat{e}_i$, as

$$A^T A \begin{bmatrix} \hat{e}_0 \\ \hat{e}_1 \\ \hat{e}_2 \end{bmatrix} = A^T A R = \begin{bmatrix} \lambda_0 & 0 & 0 \\ 0 & \lambda_1 & 0 \\ 0 & 0 & \lambda_2 \end{bmatrix} R, \quad (13)$$

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

$$\begin{bmatrix} \lambda_0 & 0 & 0 \\ 0 & \lambda_1 & 0 \\ 0 & 0 & \lambda_2 \end{bmatrix} \vec{V}' = B', \quad (14)$$

where $\vec{V}' = (U', V', W') = R^T\vec{V}$ and $B' = (b'_0, b'_1, b'_2)^T = R^T A^T B$. If λ_0 is small, $\lambda_0 \leq \tau_{D2}$, $\lambda_1, \lambda_2 > \tau_{D2}$, we are dealing with a line normal velocity, $\vec{V}_L = (U_L, V_L, W_L)$. 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 $\hat{e}_0 = \hat{e}_1 \times \hat{e}_2$, which yields a third equation. The system of equations to be solved is:

$$V\prime = \epsilon_{10}U_L + \epsilon_{11}V_L + \epsilon_{12}W_L = \frac{b'_1}{\lambda_1} \quad (15)$$

$$W\prime = \epsilon_{21}U_L + \epsilon_{21}V_L + \epsilon_{22}W_L = \frac{b'_2}{\lambda_2} \quad (16)$$

$$\epsilon_{01}U_L + \epsilon_{01}V_L + \epsilon_{02}W_L = 0. \quad (17)$$

If both λ_0 and λ_1 are less than τ_{D2} then we can only compute planar normal range flow. In this case, we have one constraint:

$$\epsilon_{20}U_P + \epsilon_{21}V_P + \epsilon_{22}W_P = \frac{b'_2}{\lambda_2}. \quad (18)$$

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

$$\vec{V}_P = \frac{\frac{b'_2}{\lambda_2}}{\epsilon_{20}^2 + \epsilon_{21}^2 + \epsilon_{22}^2}\begin{bmatrix}\epsilon_{20}\\ \epsilon_{21}\\ \epsilon_{22}\end{bmatrix} = \frac{b'_2}{\lambda_2}\begin{bmatrix}\epsilon_{20}\\ \epsilon_{21}\\ \epsilon_{22}\end{bmatrix}. \quad (19)$$

Since A^TA 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 perspective projection of X and Y) the motion constraint equation becomes:

$$I_X U + I_Y V + I_t = 0, \quad (20)$$

where U and V are the first 2 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 β 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, τ_{D3}.

5 Direct Regularized Range Flow

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

$$\int\int\int\int (Z_X U + Z_Y V + W + Z_t)^2 + \alpha^2(U_X^2 + U_Y^2 + U_Z^2 +$$

$$V_X^2 + V_Y^2 + V_Z^2 + W_X^2 + W_Y^2 + W_Z^2)\partial X\partial Y\partial Z\partial t. \quad (21)$$

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

$$\underbrace{\begin{bmatrix}(\alpha^2 + Z_X^2) & Z_X Z_Y & Z_X\\ Z_X Z_Y & (\alpha^2 + Z_Y^2) & Z_Y\\ Z_X & Z_Y & (\alpha^2 + 1)\end{bmatrix}}_{A}\begin{bmatrix}U\\ V\\ W\end{bmatrix} = \begin{bmatrix}(\alpha^2\bar{U} - Z_X Z_t)\\ (\alpha^2\bar{V} - Z_Y Z_t)\\ (\alpha^2\bar{W} - Z_t)\end{bmatrix}. \quad (22)$$

The Gauss Seidel equations then become:

$$\begin{bmatrix}U^{n+1}\\ V^{n+1}\\ W^{n+1}\end{bmatrix} = A^{-1}\begin{bmatrix}(\alpha^2\bar{U^n} - Z_X Z_t)\\ (\alpha^2\bar{V^n} - Z_Y Z_t)\\ (\alpha^2\bar{W^n} - Z_t)\end{bmatrix}. \quad (23)$$

6 Indirect Regularized Range Flow

Given $\vec{V}_F$, $\vec{V}_L$ and $\vec{V}_P$ as computed above by a least squares computation we can compute range flow by regularizing

$$\int\int\int\int ||t_0(\vec{V}\cdot\hat{e}_0)\hat{e}_0 + t_1(\vec{V}\cdot\hat{e}_1)\hat{e}_1 + t_2(\vec{V}\cdot\hat{e}_2)\hat{e}_2 - \vec{V}_M||_2^2$$

$$+\alpha^2(U_X^2 + U_Y^2 + U_Z^2 + V_X^2 + V_Y^2 + V_Z^2 +$$

$$W_X^2 + W_Y^2 + W_Z^2)\partial X\partial Y\partial Z\partial t, \quad (24)$$

where t_0, t_1 and t_2 are integer values of 1 or 0, depending on whether or not a full, line normal, or plane normal velocity can be computed directly in the least squares framework at a location. If $\vec{V}_F$ can be computed $t_0 = t_1 = t_2 = 1$, if $\vec{V}_L$ only can be computed then $t_0 = 0$ and $t_1 = t_2 = 1$ and if $\vec{V}_P$ only can be computed then $t_0 = t_1 = 0$ and $t_2 = 1$. If $t_0 = t_1 = t_2 = 0$ no velocity information can be recovered. $\vec{V}_M$ is the computed full, line or plane range velocity. The term

in the squared 2-norm in equation (24) can be written as

$$F(\vec{V}) = t_0(\vec{V}\cdot\hat{e}_0)\hat{e}_0 + t_1(\vec{V}\cdot\hat{\epsilon}_1)\hat{e}_1 + t_2(\vec{V}\cdot\hat{e}_2)\hat{e}_2 - \vec{V}_M$$
$$= B\vec{V} - \vec{V}_M. \quad (25)$$

The 3×3 idempotent matrix B can be written as

$$B = t_0\hat{e}_0^T\cdot\hat{e}_0 + t_1\hat{e}_1^T\cdot\hat{e}_1 + t_2\hat{e}_2^T\cdot\hat{e}_2, \quad (26)$$

where $\hat{e}^T$ is a row vector and $\hat{e}$ is a column vector.

We can compute $\frac{\partial F(\vec{V})}{\partial \vec{V}}$ as $2B(B\vec{V} - \vec{V}_M) = 2B(\vec{V} - \vec{V}_M)$. Thus the Euler-Lagrange equations can be written as:

$$\underbrace{(B+\alpha^2 I)}_{A}\vec{V} = \begin{bmatrix} \alpha^2\bar{U} + BU_M \\ \alpha^2\bar{V} + BV_M \\ \alpha^2\bar{W} + BW_M \end{bmatrix}, \quad (27)$$

yielding the Gauss Seidel equations:

$$\begin{bmatrix} U^{n+1} \\ V^{n+1} \\ W^{n+1} \end{bmatrix} = A^{-1}\begin{bmatrix} \alpha^2\bar{U}^n + BU_M \\ \alpha^2\bar{V}^n + BV_M \\ \alpha^2\bar{W}^n + BW_M \end{bmatrix} \quad (28)$$

7 Combined Range Flow from Intensity and Range Derivatives

It is possible to compute $\vec{V}$ using both intensity and range derivatives via equations (20) and (7) and the same smoothness term given in equations (21) and (24). We regularize:

$$\int\int\int\int (Z_X U + Z_Y V + W + Z_t)^2 + \beta^2(I_X U + I_Y V + I_t)^2$$
$$+\alpha^2(U_X^2 + U_Y^2 + U_Z^2 + +V_X^2 + V_Y^2 + V_Z^2 +$$
$$W_X^2 + W_Y^2 + W_Z^2)\partial X\partial Y\partial Z\partial t, \quad (29)$$

The Euler-Lagrange equations are:

$$A\begin{bmatrix} U \\ V \\ W \end{bmatrix} = \begin{bmatrix} \alpha^2\bar{U} - Z_X Z_t - \beta^2 I_X I_t \\ \alpha^2\bar{V} - Z_Y Z_t - \beta^2 I_Y I_t \\ \alpha^2\bar{W} - Z_t - \beta^2 I_t \end{bmatrix}, \quad (30)$$

where A is

$$\begin{bmatrix} Z_X^2 + \beta^2 I_X^2 + \alpha^2 & Z_X Z_Y + \beta^2 I_X I_Y & Z_X \\ Z_X Z_Y + \beta^2 I_X I_Y & Z_Y^2 + \beta^2 I_Y^2 + \alpha^2 & Z_Y \\ Z_X & Z_Y & 1+\alpha^2 \end{bmatrix}. \quad (31)$$

The Gauss Seidel equations are then:

$$\begin{bmatrix} U^{n+1} \\ V^{n+1} \\ W^{n+1} \end{bmatrix} = A^{-1}\begin{bmatrix} \alpha^2\bar{U}^n - Z_X Z_t - \beta^2 I_X I_t \\ \alpha^2\bar{V}^n - Z_Y Z_t - \beta^2 I_Y I_t \\ \alpha^2\bar{W}^n - Z_t - \beta^2 I_t \end{bmatrix}. \quad (32)$$

The matrix A^{-1} only has to be computed once in equations (23), (28) and (32) at the start of the iterations. The existence of the inverse is guaranteed by the Sherman-Morrison-Woodbury formula [13].

8 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 [8]. A simple averaging filter $[\frac{1}{4}, \frac{1}{2}, \frac{1}{4}]$ 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_x, we first convolve the prefiltering kernel in the t dimension, that 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 would not true (but it is almost always true for our data).

9 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). $\vec{V}_c$ is the correct image/range flow and $\vec{V}_e$ is the estimated or computed image/range flow in the equations below. For magnitude error we report:

$$\psi_M = \frac{\left|\,||\vec{V}_c||_2 - ||\vec{V}_e||_2\,\right|}{||\vec{V}_c||_2} \times 100\%, \quad (33)$$

while for angle error we report:

$$\psi_A = \arccos(\hat{V}_c\cdot\hat{V}_e). \quad (34)$$

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

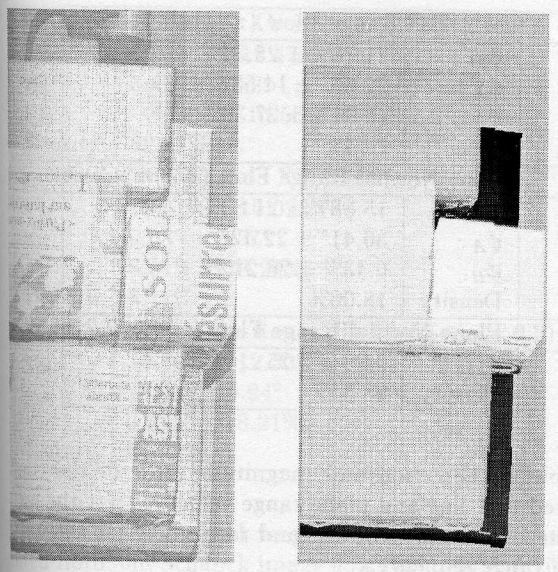
$$\vec{V}_{Lc} = (\vec{V}_c\cdot\hat{\epsilon}_1)\hat{\epsilon}_1 + (\vec{V}_c\cdot\hat{\epsilon}_2)\hat{\epsilon}_2. \quad (35)$$

Of course $\hat{\epsilon}_1$ and $\hat{\epsilon}_2$ 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 (33) and (34). Finally, for planar normal range flow we can only compute the planar magnitude error:

$$\psi_{P3D} = \frac{\left|\,\vec{V}_c\cdot\hat{V}_P - ||\vec{V}_P||_2\,\right|}{||\vec{V}_P||_2} \times 100\%, \quad (36)$$

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 ψ_{abs}, the average absolute error:

$$\psi_{abs} = \sum^{N} ||\vec{V}_c\cdot\hat{V}_P||_2 - ||\vec{V}_P||_2 \quad (37)$$



(a) (b)

Figure 1: (a) The smoothed subsampled intensity image for frame 25 of the NRC sequence and (b) its corresponding depth (Z) image.

Finally we compute bias magnitude error for full and line normal flow as

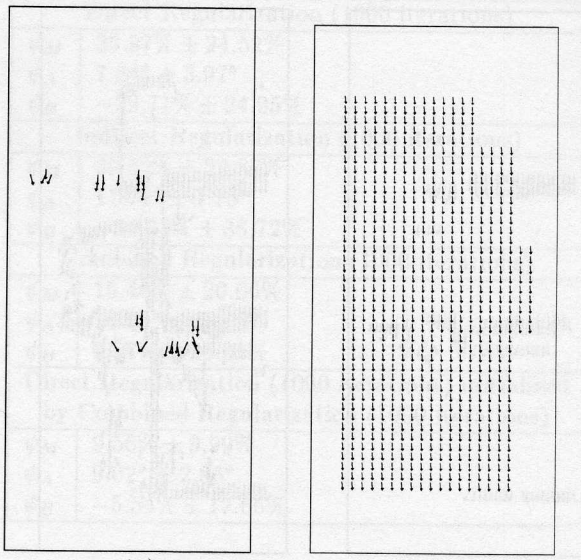
$$\psi_B = \frac{1}{N} \sum^{N} \frac{||\vec{V}_c||_2 - ||\vec{V}_e||_2}{||\vec{V}_c||_2} \times 100\%, \qquad (38)$$

which will tell us if there is a consistent under estimate or over estimate in the magnitude error.

10 Results

We made one real range sequence 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 intensity images by rescaling their intensities so that all intensities were white and then rescaling the acquired images by these same factors.

[1]Thanks to Luc Cournoyer at NRC for helping the 1^{st} author make this data.

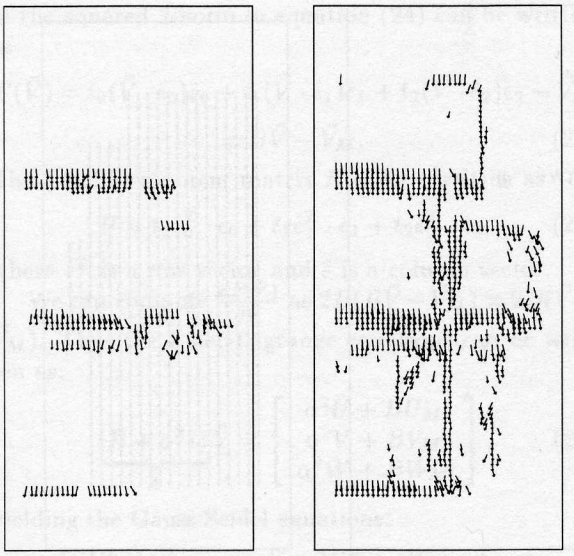


(a) (b)

Figure 2: The (a) computed and (b) correct XY full range flows for the NRC sequence.

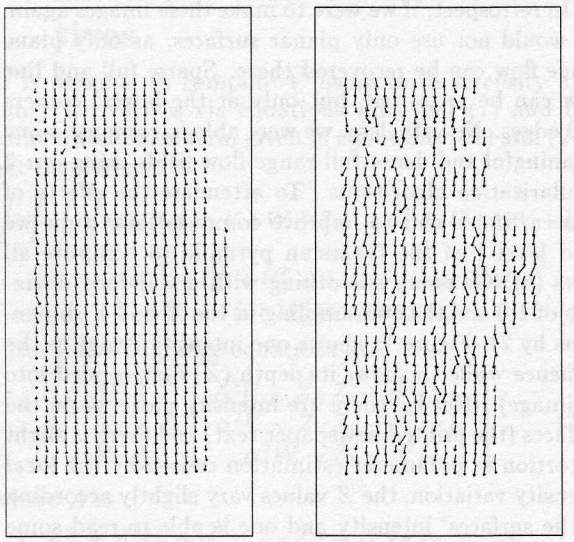
In retrospect, if we were to make these images again, we would not use only planar surfaces, as only plane range flow can be recovered there. Sparse full and line flow can be recovered, but only at the boxes' corners and edges. Nevertheless, we were able to compute some meaningful and dense full range flow fields using our 3 regularization algorithms. To attenuate the effects of noise artifacts and to improve computational time we used level 1 of the Gaussian pyramid to compute all flows (3D Gaussian smoothing with a standard deviation of 1.0 and the subsampling in the X and Y dimensions by 2). Figure 1a shows one intensity image in the sequence while 1b shows its depth (Z) map (scaled into an image). Because there are intensity patterns on the surfaces (the printed newspaper text) and there is slight distortion in parameter estimation caused by the local intensity variation, the Z values vary slightly according to the surfaces' intensity and one is able to read some text in the Z images (see Figure (1b)). The top and bottom parts and some of the right side of the image in Figure (1a) are part of the linear positioner setup; one cannot obtain good derivative values here and to increase computational accuracy and speed we masked out these parts of the image in our flow calculations.

Figures 2 and 3 show the corresponding computed and correct full and line XY flows for this sequence (see section 3). The plane flows (which we don't show here) have very small magnitudes. To fully show the 3D flows, we also need to show the XZ flows; however since the X and Z flow components are only about 6%



(a) (b)

Figure 3: The (a) computed and (b) correct XY line range flows for the NRC sequence.



a b

Figure 4: (a) The image flow computed using Horn and Schunck method (2000 iterations) and (b) the XY range flow using the direct image-range flow calculation.

of the Y component for this sequence the XZ flows are generally quite small relative to the XY flows and due to space limitations are not shown here. The correct line and plane flow are computed using eigenvectors of $A^T A$; hence they are not really "correct" but have the

Full Range Flow ($\tau_{D2} = 0.2$)	
ψ_M	$21.76\% \pm 23.36\%$
ψ_A	$22.66° \pm 14.53°$
ψ_B	$16.41\% \pm 27.39\%$
Density	1.04%
Line Normal Range Flow ($\tau_{D2} = 0.2$)	
ψ_M	$15.88\% \pm 21.83\%$
ψ_A	$36.41° \pm 22.42°$
ψ_B	$6.43\% \pm 26.21\%$
Density	18.06%
Plane Normal Range Flow ($\tau_{D2} = 0.2$)	
ψ_{abs}	0.20 ± 3.25
Density	28.34%

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 NRC real range sequence.

eigenvector error. Table 1 give the quantitative results for these full, line normal and plane normal fields. We report only ψ_{abs} for the plane normal flow; the relative magnitude and angle errors and their standard deviations for this flow are quite large due to the often large relative differences of individual computed and correct plane normal flows. In all cases, the absolute sizes of the computed and correct plane normal velocities were quite small.

Figure 4a shows the image flow recovered by Horn and Schunck's algorithm (2000 iterations) (section (2)) while Figure 4b shows the XY range flow from using a least squares computation on the intensity and range derivatives. Table 2 show the magnitude and direction errors for these flows (plus the flow for Lucas and Kanade). The image flows may appear to be a bit better (although the Lucas and Kanade flow is not dense) but the image-range algorithm actually computed 3D range flow.

Figures 5 and 6a show the regularized range flow fields for the direct (section (5)), combined (section (7)) and indirect regularization (section (6)) algorithms for 1000 iterations while Table 3 shows their magnitude, angle and bias 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. For indirect regularization results, while better, are still not good. The denser flow fields result wherever full or line normal flow can be computed. For the indirect flow field computed with 1000 iterations, 50.46% had an magnitude error more that 50% (average magni-

Horn and Schunck XY Flow (2000 iterations)	
ψ_M	$10.33\% \pm 12.47\%$
ψ_A	$3.48° \pm 5.53°$
ψ_B	$-4.54\% \pm 15.55\%$
Density	82.67%
Lucas and Kanade XY Flow ($\tau_{D1} = 1.0$)	
ψ_M	$10.51\% \pm 10.073\%$
ψ_A	$9.68° \pm 5.57°$
ψ_B	$6.01\% \pm 13.26\%$
Density	8.11%
Least Squares Image-Range 3D Flow ($\tau_{D3} = 0.2$)	
ψ_M	$13.80\% \pm 12.50\%$
ψ_A	$14.04° \pm 5.94°$
ψ_B	$3.87\% \pm 18.21\%$
Density	65.25%

Table 2: Direction and magnitude error of the computed Horn and Schunck image flow (for 1000 and 2000 iterations), Lucas and Kanade image flow (for $\tau_{D1} = 0.1$ and $\tau_{D1} = 1.0$) and for the 3D range flow computed via the least squares optical-range flow algorithm (for $\tau_{D3} = 0.2$) for the NRC range sequence.

tude error of $58.84\% \pm 12.58\%$ and average angle error of $12.70° \pm 9.92°$). 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 6 and the error in Table 3. 71.75% of the flow had 10% or less magnitude error (average magnitude error of $4.29\% \pm 2.51\%$ and average angle error of $0.24° \pm 1.64°$). This was the best result of all the NRC flows when using range flow only. This use of an initial set of non-zero velocities in the initialization step of regularization seems to be one way to obtain dense accurate flow for the NRC sequence.

11 Conclusions

We have shown the computation of full, line normal and plane normal range flow on a real intensity/range sequence. Our computation was in a least squares framework: total least squares is used in [11, 13, 12] and we are currently investigating the difference.

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 and indirect regularization algorithms 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; the 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 3^{rd} component of range flow. Finally, when we initialized direct regularization with combined regularized flow, we obtained the best results.

Direct Regularization (1000 iterations)	
ψ_M	$39.97\% \pm 24.52\%$
ψ_A	$7.84° \pm 3.97°$
ψ_B	$-39.71\% \pm 24.95\%$
Indirect Regularization (1000 iterations)	
ψ_M	$39.59\% \pm 22.87\%$
ψ_A	$11.72° \pm 7.75°$
ψ_B	$-24.31\% \pm 38.72\%$
Combined Regularization (1000 iterations)	
ψ_M	$15.46\% \pm 20.06\%$
ψ_A	$16.50° \pm 13.08°$
ψ_B	$0.97\% \pm 25.31\%$
Direct Regularization (1000 iterations) initialized by Combined Regularization (1000 iterations)	
ψ_M	$9.56\% \pm 9.99\%$
ψ_A	$9.02° \pm 2.86°$
ψ_B	$-5.54\% \pm 12.66\%$

Table 3: Direction and magnitude error of the direct, combined and indirect regularized flow for 1000 iterations for the NRC sequence. Also shown are the error results when the indirect regularized flow is used to initialize the direct regularization. The density of all flow fields (due to masking) is 82.68%.

Acknowledgements: We gratefully acknowledge 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

[1] J. L. Barron, D. J. Fleet, and S. S. Beauchemin. Performance of optical flow techniques. *IJCV*, 12(1):43–77, 1994.

[2] J.-A. Beraldin, S.F. El-Hakim, and F. Blais. Performance evaluation of three active vision systems built at the national research council of canada. In *Conf. on Optical 3D Measurement Techniques III*, pages 352–361, October 1995.

[3] K. Chaudhury and R. Mehrota amd C. Srinivasan. Detecting 3d flow. In *Proc. IEEE Int. Conf.*

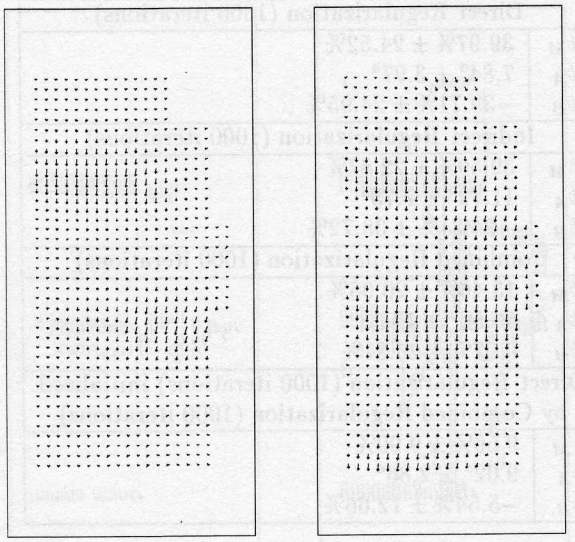


Figure 5: (a) Direct regularization with 1000 iterations and (b) Indirect regularization with 1000 iterations on the NRC sequence.

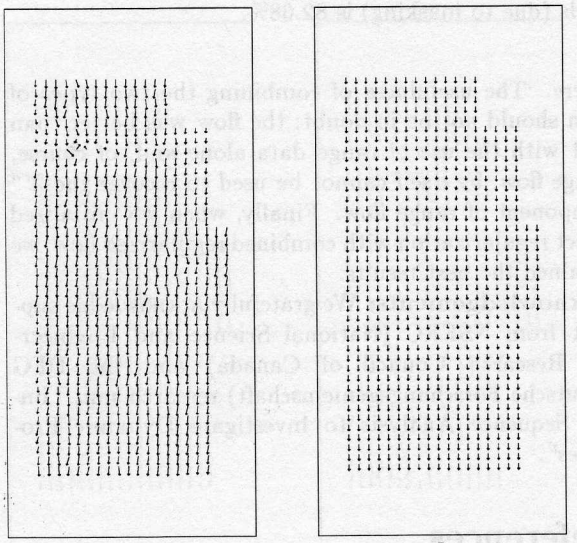


Figure 6: (a) Combined regularization with 1000 iterations and (b) direct regularization with 1000 iterations after initialization by indirect regularization with 1000 iterations for the NRC sequence.

Robotics and Automation, volume 2, pages 1073–1078, May 1994.

[4] B. K. P. Horn and B. G. Schunck. Determining optical flow. *Artificial Intelligence*, 17:185–204, 1981.

[5] Gregory J. Klien and Ronald H. Huesman. A 3d optical approach to addition of deformable pet volumes. In *IEEE Nonrigid and Articulated Motion Workshop*, pages 136–143, June 1997.

[6] B. D. Lucas and T. Kanade. An iterative image-registration technique with an application to stereo vision. In *Image Understanding Workshop*, pages 121–130. DARPA, 1981. (see also IJCAI81, pp674-679).

[7] W.H. Press, B.P. Flannery, S.A. Teukolsky, and W. T. Vetterling. *Numerical Recepes in C: The art of scientific computing*. Cambridge University Press, 1988.

[8] E.P. Simoncelli. Design of multi-dimensional derivative filters. In *IEEE Int. Conf. Image Processing*, volume 1, pages 70–793, 1994.

[9] S.M. Song and R.M. Leahy. Computation of 3d velocity fields from 3d cine ct images of a human heart. *IEEE Trans. Medical Imaging*, 10(1):295–306, 1991.

[10] S.M. Song, R.M. Leahy, D.P Boyd, and B.H. Brundage. Determining cardiac velocity fields and intraventricular pressure distribution from a sequence of ultrafast ct cardiac images. *IEEE Trans. Medical Imaging*, 13(2):386–397, 1994.

[11] H. Spies, H. Haußecker, B. Jähne, and J.L.Barron. Differential range flow estimation. In *21.Symposium fur Mustererkennung, DAGM '1999*, pages 309–316. Springer, September 15-17th 1999. Bonn, Germany.

[12] H. Spies, B. Jähne, and J.L. Barron. Dense range flow from depth and intensity data. In *Int. Conf. on Pattern recognition ICPR2000*, September 2000. submitted.

[13] H. Spies, B. Jähne, and J.L. Barron. Regularised range flow. In *European Conference on Computer Vision ECCV2000*, June 2000.

[14] M. Yamamoto, P. Boulanger, J. Beraldin, and M. Rioux. Direct estimation of range flow on deformable shape from a video rate range camera. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(1):82–89, January 1993.

[15] Z. Zhou, C.E. Synolakis, R.M. Leahy, and S.M. Song. Calculation of 3d internal displacement fields from 3d x-ray computer tomographic images. *Proc. R. Soc. Lond. A*, 449(1937):537–554, 1995.