Regularised Range Flow

Hagen Spies[1,2], Bernd Jähne[1], and John L. Barron[2]

[1] Interdisciplinary Center for Scientific Computing,
University of Heidelberg, INF 368, 69120 Heidelberg, Germany,
{Hagen.Spies,Bernd.Jaehne}@iwr.uni-heidelberg.de
[2] Dept. of Comp. Science, University of Western Ontario,
London, Ontario, N6G 5B7 Canada,
barron@csd.uwo.ca

Abstract. Extending a differential total least squares method for range flow estimation we present an iterative regularisation approach to compute dense range flow fields. We demonstrate how this algorithm can be used to detect motion discontinuities. This can can be used to segment the data into independently moving regions. The different types of aperture problem encountered are discussed. Our regularisation scheme then takes the various types of flow vectors and combines them into a smooth flow field within the previously segmented regions. A quantitative performance analysis is presented on both synthetic and real data. The proposed algorithm is also applied to range data from castor oil plants obtained with the Biris laser range sensor to study the 3-D motion of plant leaves.

Keywords. *range flow, range image sequences, regularisation, shape, visual motion.*

1 Introduction

We are concerned with the estimation of local three-dimensional velocity from a sequence of depth maps. Previously we introduced a total least squares (TLS) algorithm for the estimation of this so called *range flow* [1]. It is shown that the result of this TLS algorithm can be used to detect boundaries between independently moving regions, which enables a segmentation. However, within these regions the computed flow fields are not generally dense. To amend this we present an iterative regularisation method to compute dense full flow fields using the information available from the TLS estimation.

Most previous work on range sequence analysis focuses on the estimation of the 3D motion parameters of either a moving sensor or an object [2,3,4,5,6]. Such approaches implicitly assume global rigidity. In contrast we are dealing with only locally rigid objects moving in an environment observed by a stationary sensor. As with optical flow calculation we initially assume that the flow field can be approximated as being constant within a small local aperture [7,8]. In a second processing step this is replaced by requiring the flow field to be smooth. The work presented here is related to previously reported model based range flow

D. Vernon (Ed.): ECCV 2000, LNCS 1843, pp. 785–799, 2000.

estimation on non-rigid surfaces [9,10]. The 3D range flow can also be recovered from optical flow if other surface properties such as depth or correspondences are available [11]. Some other work includes 2D range flow obtainable from a radial range sensor [12] and car tracking in range image sequences [13].

The underlying constraint equation is introduced in Sect. 2. Then Sect. 3 recapitulates the TLS estimation technique, in particular it is described how sensible parameters can be estimated even if not enough constraints are available, see Sect. 3.2. This is a generalisation of the known normal flow estimation in optical flow algorithms. It is also demonstrated how boundaries in the motion field between differently moving regions can be detected. Section 4 then shows how a dense parameter field can be obtained exploiting the previously calculated information. In Sect. 5 we proceed towards a quantitative performance analysis, where we introduce appropriate error measures for range flow. The methods potential is exploited on both synthetic (Sect. 5.2) and real data (Sect. 5.3). Results of applying our algorithm to sequences of range scans of plant leaves are given in Sect. 6.

The work reported here was performed with data gathered by a *Biris* laser range sensor [14]. The algorithm introduced could, however, be equally well used on dense depth maps obtained from structured lighting, stereo or motion and structure techniques.

2 Constraint Equation

Depth is taken as a function of space and time $Z = Z(X, Y, T)$. From the total derivative with respect to time we derive the *range flow motion constraint* equation [3,9]

$$Z_X \dot{X} + Z_Y \dot{Y} - \dot{Z} + Z_T = 0 \ . \tag{1}$$

Here partial derivatives are denoted by subscripts and time derivatives by using a dot. We call the 3D motion vector *range flow* $\boldsymbol{f}$ and introduce the following abbreviation $\boldsymbol{f} = [U\ V\ W]^T = [\dot{X}\ \dot{Y}\ -\dot{Z}]^T$. The range flow motion constraint (1) then becomes

$$Z_X U + Z_Y V + W + Z_T = [Z_X\ Z_Y\ 1\ Z_T]^T \begin{bmatrix} U \\ V \\ W \\ 1 \end{bmatrix} = 0 \ . \tag{2}$$

As this gives only one constraint equation in three unknowns we need to make further assumptions, this is the aperture problem revisited.

Equation (2) describes a plane in velocity space. If there are three mutually independent planes in a local neighbourhood we can compute *full flow* under the assumption of locally constant flow fields. Obviously this could easily be extended to incorporate linear flow variations. If there is only one repeated constraint in the entire considered neighbourhood only the normal flow can be recovered. As

this occurs on planar surfaces we call this *plane flow*. When two planes meet in the aperture we get two constraint classes, in this case it is possible to determine all but the part of the flow in the direction of the intersection. The flow with minimal norm perpendicular to the intersecting line will be called *line flow* [1]. The following section describes how we can compute the described flow types using a total least squares (TLS) estimator.

3 Total Least Squares Estimation

The TLS solution presented here is an extension of the *structure tensor* algorithm for optical flow estimation [15,8]. The method may also be viewed as a special case of a more general technique for parameter estimation in image sequences [16].

Assuming constant flow in a region containing n pixel we have n equations (2). With $\boldsymbol{d} = [Z_X \; Z_Y \; 1 \; Z_T]^T$, $\boldsymbol{u} = [U \; V \; W \; 1]^T$ and the data matrix $\boldsymbol{D} = [\boldsymbol{d}_1 \dots \boldsymbol{d}_n]^T$, the flow estimation in a total least squares sense can be formulated as

$$||\boldsymbol{D}\boldsymbol{u}||_2 \rightarrow \min \quad \text{subject to} \quad \boldsymbol{u}^T\boldsymbol{u} = 1 \; . \tag{3}$$

The solution is given by the eigenvector $\hat{e}_4$, corresponding to the smallest eigenvalue λ_4 of the generalised structure tensor

$$\boldsymbol{F} = \boldsymbol{D}^T\boldsymbol{D} = \begin{bmatrix} < Z_X Z_X > & < Z_X Z_Y > & < Z_X > & < Z_X Z_T > \\ < Z_Y Z_X > & < Z_Y Z_Y > & < Z_Y > & < Z_Y Z_T > \\ < Z_X > & < Z_Y > & < 1 > & < Z_T > \\ < Z_T Z_X > & < Z_T Z_Y > & < Z_T > & < Z_T Z_T > \end{bmatrix} \; . \tag{4}$$

Here $< \cdot >$ denotes local averaging using a Box or Binomial filter. The desired range flow is then given by

$$\boldsymbol{f}_f = \frac{1}{e_{44}} \begin{bmatrix} e_{14} \\ e_{24} \\ e_{34} \end{bmatrix} \; . \tag{5}$$

As $\boldsymbol{F}$ is real and symmetric the eigenvalues and eigenvectors can easily be computed using Jacobi-Rotations [17]. In order to save execution time we only compute range flow where the trace of the tensor exceeds a threshold τ_1. This eliminates regions with insufficient magnitude of the gradient. The regularisation step described in Sect. 4 subsequently closes these holes.

3.1 Detecting Motion Discontinuities

In the above we are really fitting a local constant flow model to the data. The smallest eigenvalue λ_4 directly measures the quality of this fit. In particular at motion discontinuities the data can not be described by a single flow and the

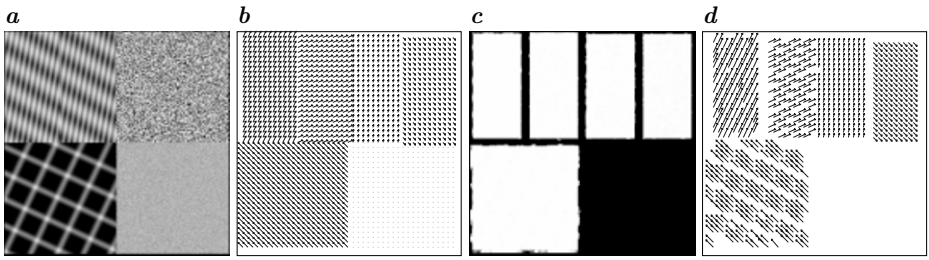

Fig. 1. Using the confidence measure to detect motion discontinuities: **a** synthetic depth map, the lower right quarter contains random noise without coherent motion. **b** X, Y–component of the correct flow field, **c** confidence measure ($\tau_2 = 0.1$) and **d** TLS full flow.

fit fails. This is also the case for pure noise without any coherent motion. To quantify this we introduce a confidence measure

$$\omega = \begin{cases} 0 & \text{if } \lambda_4 > \tau_2 \text{ or } tr(\boldsymbol{D}) < \tau_1 \\ \left(\frac{\tau_2 - \lambda_4}{\tau_2 + \lambda_4}\right)^2 & \text{else} \end{cases} . \tag{6}$$

Figure 1 shows the obtained confidence measure for a synthetic sequence of depth maps. Clearly motion discontinuities and pure noise can be identified. Also the estimated full flow is very close to the correct flow, however this full flow can not be computed everywhere regardless of ω. The next section explains why and how to deal with such situations.

3.2 Normal Flows

Let the eigenvalues of $\boldsymbol{F}$ be sorted: $\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq \lambda_4$. Thus if $\lambda_3 \approx \lambda_4$ no unique solution can be found [18]. More general any vector in the nullspace of $\boldsymbol{F}$ is a possible solution. In this case it is desirable to use the solution with minimal norm. Towards this end the possible solutions are expressed as linear combinations of the relevant eigenvectors and that with minimal Euclidean norm is chosen, see App. A for details.

On planar structures all equations (2) are essentially the same. Only the largest eigenvalue is significantly different ($> \tau_2$) from zero. The so called plane flow can then be found from the corresponding eigenvector $\hat{e}_1 = [e_{11}\ e_{21}\ e_{31}\ e_{41}]^T$ as follows

$$\boldsymbol{f}_p = \frac{e_{41}}{e_{11}^2 + e_{21}^2 + e_{31}^2} \begin{bmatrix} e_{11} \\ e_{21} \\ e_{31} \end{bmatrix} . \tag{7}$$

Linear structures exhibit two types of constraints within the considered aperture, the minimum norm solution (line flow) is found from the eigenvectors $\hat{e}_1,\ \hat{e}_2$

$$\boldsymbol{f}_l = \frac{1}{1 - e_{41}^2 - e_{42}^2} \left[e_{41} \begin{bmatrix} e_{11} \\ e_{21} \\ e_{31} \end{bmatrix} + e_{42} \begin{bmatrix} e_{12} \\ e_{22} \\ e_{32} \end{bmatrix} \right] . \tag{8}$$

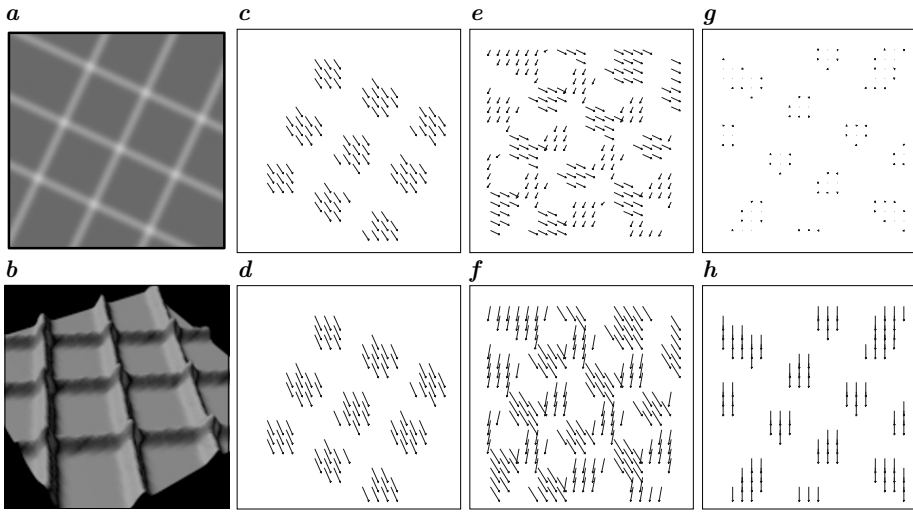

Fig. 2. Example flow types: **a** synthetic depth map, **b** rendered. $X-Y$ components of the estimated flow fields: **c** full flow, **e** line flow and **g** plane flow and $X-Z$ components of the estimated flow fields: **d** full flow, **f** line flow and **h** plane flow.

Figure 2 shows an example of the various flow types.

4 Flow Regularisation

We now introduce a simple iterative regularisation algorithm that computes smoothly varying flow fields in some previously segmented area A. Segmentation of the data into regions of different motions is best accomplished by means of the previously described threshold on the lowest eigenvalue of $\boldsymbol{F}$, see Sect. 3.1. However, if additional knowledge about the scene is available other segmentation schemes may be employed. As we are given depth data such a segmentation is often feasible.

We seek to estimate a dense and smooth flow field $\boldsymbol{v} = [U\ V\ W]^T$. In places where flow estimations from the above TLS algorithm exist we denote them $\boldsymbol{f}$, computed from (5,7,8) as appropriate. As we are now working in 3 dimensions and from the structure of the TLS solution given by (38) we can use the reduced eigenvectors as, not necessarily orthogonal, basis for the desired solution

$$\boldsymbol{b}_i = \frac{1}{\sum_{k=1}^{3} e_{ki}^2} \begin{bmatrix} e_{1i} \\ e_{2i} \\ e_{3i} \end{bmatrix} \quad i = 1, 2, 3 \ . \tag{9}$$

Using this notation we define a projection matrix which projects onto the subspace that was determined by our TLS algorithm

$$\boldsymbol{P} = \bar{\boldsymbol{B}}_p \bar{\boldsymbol{B}}_p^T \quad \text{where} \quad \lambda_p > \lambda_{p+1} \approx \ldots \approx \lambda_4 \approx 0 \ , \tag{10}$$

$$\bar{\boldsymbol{B}}_p = [\hat{b}_1 \ \ldots \ \hat{b}_p] \ . \tag{11}$$

Each estimated flow vector $\boldsymbol{f}_{f,p,l}$ constrains the solution within this subspace. We therefore require the regularised solution to be close in a least squares sense

$$(\boldsymbol{P}\boldsymbol{v} - \boldsymbol{f})^2 \rightarrow \min \ . \tag{12}$$

At locations where no solution has been computed obviously no such data term exists. To ensure smoothly varying parameters we use a smoothness term

$$\sum_{i=1}^{3} (\nabla v_i)^2 \rightarrow \min \ . \tag{13}$$

Obviously the use of this simple membrane model is only justified because we have already segmented the data into differently moving objects. If no such segmentation were available more elaborate schemes would have to be considered [19,20]. The above smoothness term usually considers only spatial neighbourhoods ($\nabla = [\partial x, \partial y]^T$), however this is easily extended to enforce temporal smoothness as well ($\nabla = [\partial x, \partial y, \partial t]^T$).

Combining the data (12) and smoothness (13) terms in the considered area A yields the following minimisation problem

$$\int_A \underbrace{\left\{ \omega \left(\boldsymbol{P}\boldsymbol{v} - \boldsymbol{f}\right)^2 + \alpha \sum_{i=1}^{3} (\nabla v_i)^2 \right\}}_{h(\boldsymbol{v})} \mathrm{d}\boldsymbol{r} \rightarrow \min \ . \tag{14}$$

Where ω, given by equation (6), captures the confidence of the TLS solution. The overall smoothness can be regulated by the constant α. The minimum of (14) is reached when the Euler-Lagrange equations are satisfied

$$\frac{\partial h}{\partial v_i} - \frac{\mathrm{d}}{\mathrm{d}x} \frac{\partial h}{\partial (v_i)_x} - \frac{\mathrm{d}}{\mathrm{d}y} \frac{\partial h}{\partial (v_i)_y} = 0 \quad ; \ i = 1, \ldots, m \ . \tag{15}$$

If an extension in the temporal domain is anticipated another term $-\frac{\mathrm{d}}{\mathrm{d}t} \frac{\partial h}{\partial (v_i)_t}$ has to be added. Subscripts x, y, t denote partial differentiation. Using vector notation we write the Euler-Lagrange equations as follows:

$$\frac{\partial h}{\partial \boldsymbol{v}} - \frac{\mathrm{d}}{\mathrm{d}x} \frac{\partial h}{\partial (\boldsymbol{v}_x)} - \frac{\mathrm{d}}{\mathrm{d}y} \frac{\partial h}{\partial (\boldsymbol{v}_y)} = 0 \ . \tag{16}$$

Computing the derivatives yields

$$2\omega \boldsymbol{P}(\boldsymbol{P}\boldsymbol{v} - \boldsymbol{f}) - 2\alpha \left[\frac{\mathrm{d}}{\mathrm{d}x}(\boldsymbol{v}_x) + \frac{\mathrm{d}}{\mathrm{d}y}(\boldsymbol{v}_y) \right] = 0 \ . \tag{17}$$

Introducing the Laplacian $\Delta \boldsymbol{v} = \boldsymbol{v}_{xx} + \boldsymbol{v}_{yy}$ we get

$$\omega \boldsymbol{P}\boldsymbol{v} - \omega \boldsymbol{P}\boldsymbol{f} - \alpha \Delta \boldsymbol{v} = 0 \ , \tag{18}$$

where the idempotence of the projection matrix $\boldsymbol{P}\boldsymbol{P} = \boldsymbol{P}$ is used. The Laplacian can be approximated as $\Delta\boldsymbol{v} = \bar{\boldsymbol{v}} - \boldsymbol{v}$, where $\bar{\boldsymbol{v}}$ denotes a local average. In principle this average has to be calculated without taking the central pixel into consideration. Using this approximation we arrive at

$$(\omega\boldsymbol{P} + \alpha\mathbb{1})\ \boldsymbol{v} = \alpha\bar{\boldsymbol{v}} + \omega\boldsymbol{P}\boldsymbol{f}\ . \tag{19}$$

This enables an iterative solution to the minimisation problem. We introduce $\boldsymbol{A} = \omega\boldsymbol{P} + \alpha\mathbb{1}$ and get an update $\boldsymbol{v}^{k+1}$ from the solution at step k

$$\boldsymbol{v}^{k+1} = \alpha\boldsymbol{A}^{-1}\bar{\boldsymbol{v}}^k + \omega\boldsymbol{A}^{-1}\boldsymbol{P}\boldsymbol{f}\ . \tag{20}$$

Initialisation is done as $\boldsymbol{v}^0 = \boldsymbol{0}$. The matrix $\boldsymbol{A}^{-1}$ only has to be computed once, existence of the inverse is guaranteed by the Sherman-Morrison-Woodbury formula [21], see Appendix B.

4.1 Direct Regularisation

Instead of performing a TLS analysis first one might want to directly try to find the flow field by imposing the smoothness constraint, in analogy to the well known optical flow algorithm by Horn and Schunk [22]. As mentioned before this simple smoothness term is not generally advisable, mainly because problematic locations ($\lambda_4 > 0$) are equally taken into account. In particular it smoothes across motion discontinuities. On the other hand this regularisation works very well when a segmentation, if at all necessary, can be achieved otherwise.

However, if the TLS algorithm is used for segmenting the data, it makes sense to use the thus available information. The scheme described in Sect. 4 does usually converge much faster than direct regularisation. Yet, it is sometimes advisable to use the direct regularisation as a final processing step. The already dense and smooth flow field is used to initialise this step. Especially on real data, where the TLS estimate occasionally produces outliers, this post-processing improves the result, see Sect. 5.3.

Therefore we briefly discuss how such a direct regularisation can be applied to sequences of depth maps. The minimisation in this case reads

$$\int_A \underbrace{\left\{ (\boldsymbol{d}^T\boldsymbol{u})^2 + \alpha \sum_{i=1}^{3} (\nabla v_i)^2 \right\}}_{h(\boldsymbol{v})} \mathrm{d}r \rightarrow \min\ . \tag{21}$$

Here we only work on the first n-1 components of $\boldsymbol{u} = [\boldsymbol{v}^T, 1]^T$. Looking at the Euler-Lagrange equations (15) we get

$$2\boldsymbol{d'}(\boldsymbol{d'}^T\boldsymbol{v} - d_4) - 2\alpha\Delta\boldsymbol{v} = 0 \quad \text{where} \quad \boldsymbol{d} = [\boldsymbol{d'}^T, d_4]^T\ . \tag{22}$$

Again approximating the Laplacian as difference $\Delta\boldsymbol{v} = \bar{\boldsymbol{v}} - \boldsymbol{v}$ this can be rewritten as

$$(\boldsymbol{d'}\boldsymbol{d'}^T)\boldsymbol{v} - \alpha\bar{\boldsymbol{v}} + \alpha\boldsymbol{v} = \boldsymbol{d'}d_4 \tag{23}$$

$$\underbrace{\alpha\mathbb{1} + \boldsymbol{d'}\boldsymbol{d'}^T}_{\boldsymbol{A}_1}\boldsymbol{v} = \alpha\bar{\boldsymbol{v}} + \boldsymbol{d'}d_4\ . \tag{24}$$

Table 1. Results on synthetic data using $\alpha = 10,\ \tau_1 = 15$ and $\tau_2 = 0.1$.

sequence	iterations	E_r [%]	E_d [°]	E_b [%]
Fig. 1	100	0.6 ± 2.6	0.4 ± 1.6	-0.4
Fig. 1	500	0.5 ± 2.6	0.4 ± 1.6	-0.3
Fig. 1	1000	0.5 ± 2.6	0.4 ± 1.6	-0.3
Fig. 2	100	5.3 ± 4.3	6.4 ± 5.5	-5.3
Fig. 2	500	0.7 ± 0.8	0.6 ± 0.8	-0.5
Fig. 2	1000	0.4 ± 0.3	0.2 ± 0.2	-0.2

An iterative solution is found using the following update

$$\boldsymbol{v}^{k+1} = \alpha \boldsymbol{A}_1^{-1} \bar{\boldsymbol{p}}^k + d_4 \boldsymbol{A}_1^{-1} \boldsymbol{d'} \ . \tag{25}$$

Initialisation can be done by the direct minimum norm solution $\boldsymbol{n}$ to $\boldsymbol{d}^T \boldsymbol{p} = 0$ given by:

$$\boldsymbol{n} = \frac{-d_4}{\sum_{i=1}^{3} d_i^2} \begin{bmatrix} d_1 \\ d_2 \\ d_3 \end{bmatrix} = \frac{-Z_T}{Z_X^2 + Z_Y^2 + 1} \begin{bmatrix} Z_X \\ Z_Y \\ 1 \end{bmatrix} \ . \tag{26}$$

The existence of the inverse of $\boldsymbol{A}_1$ is guaranteed, see Appendix B.

5 Quantitative Performance Analysis

We now give a quantitative analysis of the proposed algorithm. Even though our algorithm can be used with any kind of differential depth maps we focus on depth maps taken with a laser range finder. In particular we are concerned with a *Biris* sensor [14]. First we introduce the error measures used. Due to experimental limitations the available real data with known ground truth only contains pure translational movements. Thus we also look at one synthetic sequence with a motion field that exhibits some divergence and rotation.

5.1 Error Measures

In order to quantify the obtained results three error measures are used. Let the correct range flow be called $\boldsymbol{f}_c$ and the estimated flow $\boldsymbol{f}_e$. The first error measure describes the relative error in magnitude

$$E_r = \frac{|\,(\|\boldsymbol{f}_c\| - \|\boldsymbol{f}_e\|)\,|}{\|\boldsymbol{f}_c\|} \cdot 100\ [\%] \ . \tag{27}$$

The deviation in the direction of the flow is captured by the directional error

$$E_d = \arccos\left(\frac{\boldsymbol{f}_c \cdot \boldsymbol{f}_e}{\|\boldsymbol{f}_c\|\ \|\boldsymbol{f}_e\|}\right)\ [^\circ] \ . \tag{28}$$

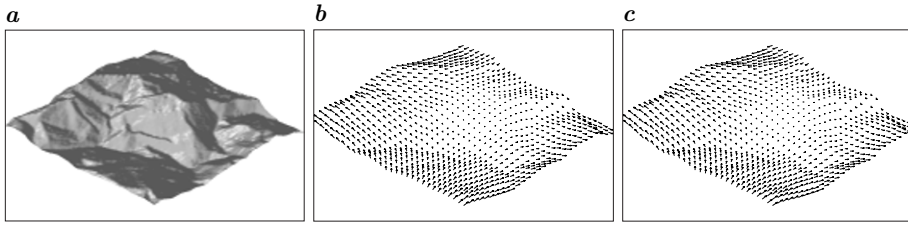

Fig. 3. Synthetic sequence generated from a real depth map. **a** rendered depth map, **b** correct flow and **c** estimated flow field ($\alpha = 10,\ \tau_1 = 25$ and $\tau_2 = 0.01$).

Even though both E_r and E_d are available at each location we only report their average values in the following. It is also interesting to see if the flow is consistently over- or underestimated. This is measured by a bias error measure

$$E_b = \frac{1}{N} \sum^{N} \left(\frac{\|\boldsymbol{f}_e\| - \|\boldsymbol{f}_c\|}{\|\boldsymbol{f}_c\|} \cdot 100 \right) \ [\%] \ , \tag{29}$$

where the summation is caried out over the entire considered region. This measure will be negative if the estimated flow magnitude is systematically smaller than the correct magnitude.

5.2 Synthetic Test Data

The performance of the TLS algorithm, described in Sect. 3, has previously been analysed on synthetic data [1]. Here we simply repeat the results that for low noise levels of less than 2%, laser range data is typically a factor 10 less noisy, all flow types (full, line, plane) can be estimated with less than 5% relative error E_r and less than 5° error E_d in the velocity range of 0.5 to 3 units per frame. Here unit stands for the mean distance between adjacent data points, typically ≈ 0.3mm.

The regularisation algorithm produces excellent results on pure synthetic data. Instead of giving numerous such results we simply state that for the sequences shown in figures 1 and 2 we achieve the results given in table 1. It can be seen that when starting with a relatively dense flow field as in Fig. 1 the use of 100 iterations provides good results. The remaining error here is mainly due to small mistakes in the segmentation into different regions. On data like that of Fig. 2, where we have large areas to be filled in, far more iterations are necessary. Convergence can be accelerated by starting with an interpolated full flow field instead of a zero flow field or by employing a hierarchical method [23].

As we are unable to make real test data with other than translational motion, we took the depth map from one scan and warped the data[1] with a known flow field. Figure 3a shows the depth map taken from a crumpled sheet of paper. It can be seen that the estimated flow is very close to the correct flow. In numbers

[1] Using bicubic interpolation.

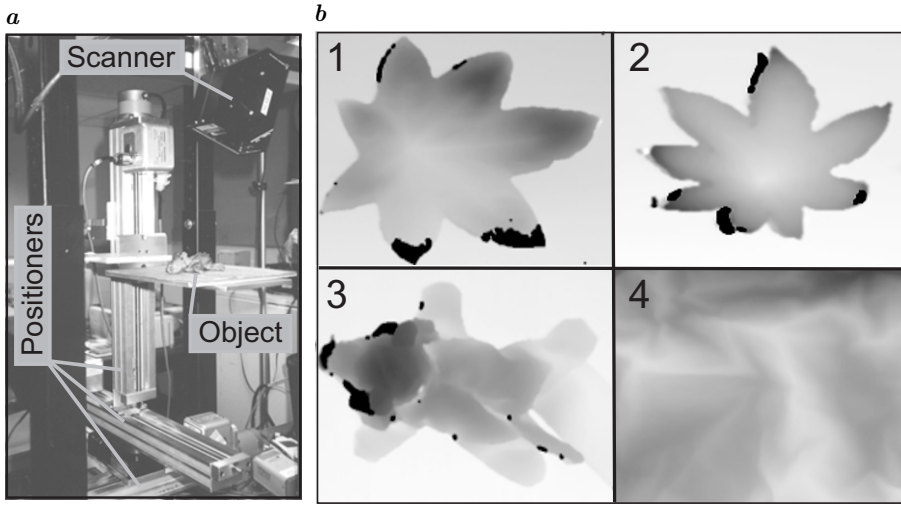

Fig. 4. Real test data: **a** laser scanner and positioners and **b** depth maps of the used test objects. Object 1 and 2 are freshly cut castor bean leaves, object 3 a toy tiger and object 4 a sheet of crumpled newspaper.

we get $E_r = 2.1 \pm 1.6\%$, $E_d = 2.3 \pm 0.8°$ and $E_b = 1.9\%$ after 100 iterations. From the last number we see that in this particular example the estimated flow is systematically larger than the correct flow, this can be attributed to the very small velocities present in this case.

5.3 Real Test Data

In order to get real test data we placed some test objects on a set of linear positioners, see Fig. 4. The positioners allow for translations along all three axes. As the objects are placed on a flat surface we segmented them prior to any computation. There is no motion discontinuity in this case and without segmenting we would have to use the background as well. Due to the lack of structure there this would make convergence extremely slow. Table 2 gives some results, here first the indirect regularisation (Sect. 4) is employed for 300 iterations with $\alpha = 10$, $\tau_1 = 15$ and $\tau_2 = 0.01$. Then the direct regularisation (Sect. 4.1) is used for another 200 iterations with $\alpha = 5$.This post-processing typically improves the result (E_r) by about 1-2%.

Given the fact that we are dealing with real data these results are quite encouraging. The average distance between two data points is 0.46mm in X-direction and 0.35mm in Y-direction, which shows that we are able to estimate sub-pixel displacements. One has to keep in mind that even slight misalignments of the positioner and the laser scanner introduce systematic errors.

Table 2. Results on real data.

object	correct flow [mm]	E_r [%]	E_d [°]	E_b [%]
1	$[0.0\ 0.0\ 0.48]^T$	3.0 ± 3.2	4.5 ± 3.0	1.5
1	$[0.32\ -0.38\ 0.32]^T$	9.2 ± 5.9	11.2 ± 6.0	1.3
2	$[-0.32\ 0.0\ 0.0]^T$	8.0 ± 11.6	7.5 ± 4.6	2.2
2	$[-0.64\ 0.0\ 0.0]^T$	6.1 ± 8.5	6.0 ± 3.8	-0.3
3	$[-0.16\ -0.19\ -0.32]^T$	3.5 ± 2.5	3.2 ± 2.4	-2.7
4	$[0.25\ 0.31\ 0.0]^T$	8.8 ± 5.4	4.3 ± 3.3	5.1

6 Plant Leaf Motion

This section finally presents some flow fields found by observing living castor oil leaves. Figure 5 shows four examples of the type of data and flow fields encountered in this application. The folding of the outer lobes is clearly visible, also a fair bit of lateral motion of the leaf. The data sets considered here are taken at night with a sampling rate of 5 minutes. Analysis is done using the same parameters as in the previous section. In Fig. 5a two overlapping leaves are observed, it is such cases that makes a segmentation based on the TLS algorithm very useful. If the leaves are actually touching each other it is quite involved to separate them otherwise.

In collaboration with the botanical institute at the University of Heidelberg and the Agriculture and Agri-Food Canada research station in Harrow, Ontario we seek to establish the leafs diurnal motion patterns. We also hope to examine the growth rate of an undisturbed leaf with a previously impossible spatial and temporal resolution. Up to now related experiments required the leaf to be fixed in a plane [15].

7 Conclusions

An algorithm to compute dense 3D range flow fields from a sequence of depth maps has been presented. It is shown how the sparse information from a TLS based technique can be combined to yield dense full flow fields everywhere within a selected area. The segmentation into regions corresponding to different motions can easily be done based on the quality of the initial TLS estimation. The performance is quantitatively assessed on synthetic and real data and the algorithm is found to give excellent results. Finally it could be shown that the motion of a living castor oil leaf can be nicely captured.

Future work includes the interpretation of the obtained flow fields from a botanical point of view. We also plan to test the method on depth data from structured lighting and stereo.

Acknowledgements. Part of this work has been funded under the DFG research unit "Image Sequence Analysis to Investigate Dynamic Processes" (Ja395/6) and by the federal government of Canada via two NSERC grants.

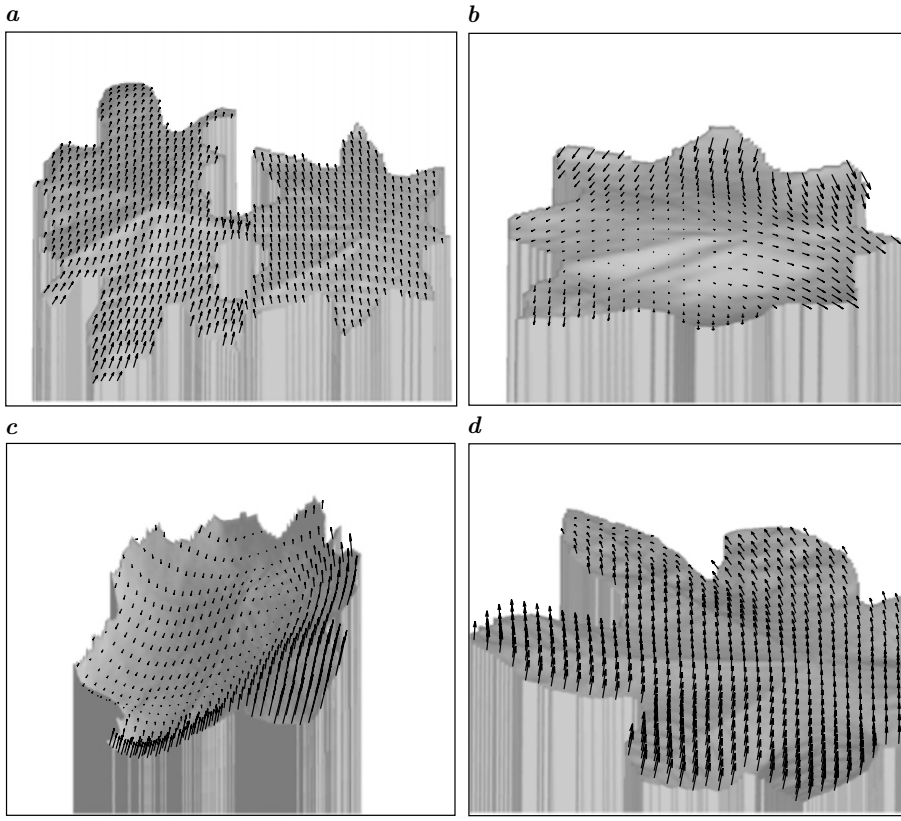

Fig. 5. Example movements of castor oil plant leaves.

A Minimum Norm Solution

Let's assume we have $\lambda_1 > \ldots > \lambda_p > \lambda_{p+1} \approx \ldots \approx \lambda_n \approx 0$ then any linear combination of the eigenvectors $\hat{e}_i;\ i > p$ is a solution to (3). Following [18] (Theorem 3.8) we now describe a way to find the minimum norm solution.

First the possible solutions are expressed as linear combinations of the relevant eigenvectors

$$\boldsymbol{p} = \sum_{i=p+1}^{n} g_i \hat{e}_i = \boldsymbol{E}_p \boldsymbol{g} \quad \text{where} \quad \boldsymbol{E}_p = [\hat{e}_{p+1}, \ldots, \hat{e}_n] = \begin{bmatrix} e_{1(p+1)} & \ldots & e_{1n} \\ \vdots & \ddots & \vdots \\ e_{n(p+1)} & \ldots & e_{nn} \end{bmatrix} . \tag{30}$$

The norm of $\boldsymbol{p}$ is then given by

$$\|\boldsymbol{p}\| = \boldsymbol{g}^T \boldsymbol{E}_p^T \boldsymbol{E}_p \boldsymbol{g} = \boldsymbol{g}^T \boldsymbol{g} = \sum_i g_i^2 \ . \tag{31}$$

The additional constraint $p_n = 1$ can be expressed as

$$p_n = \left(\sum_{i=p+1}^{n} g_i \hat{e}_i \right)_n = \sum_{i=p+1}^{n} g_i e_{ni} = \boldsymbol{v}_n^T \boldsymbol{E}_p \boldsymbol{g} = 1 \ , \tag{32}$$

where $\boldsymbol{v}_n = [0, \ldots, 0, 1]^T$. Equations (31) and (32) can be combined using a Lagrange multiplier

$$F(\boldsymbol{g}) = \boldsymbol{g}^T \boldsymbol{g} + \lambda(\boldsymbol{v}_n^T \boldsymbol{E}_p \boldsymbol{g}) \ . \tag{33}$$

The minimum is found by setting the partial derivatives of F with respect to the g_i to zero. Doing so yields

$$2g_i + \lambda e_{ni} = 0 \quad \rightarrow \quad g_i = -\frac{\lambda}{2} e_{ni} \quad \rightarrow \quad \boldsymbol{g} = -\frac{\lambda}{2} \boldsymbol{E}_p^T \boldsymbol{v}_n \ . \tag{34}$$

Substitution into (32) gives

$$-\frac{\lambda}{2} \sum_{i=p+1}^{n} e_{ni} e_{ni} = 1 \quad \rightarrow \quad \lambda = \frac{-2}{\sum_i e_{ni} e_{ni}} = \frac{-2}{\boldsymbol{v}_n^T \boldsymbol{E}_p \boldsymbol{E}_p^T \boldsymbol{v}_n} \ . \tag{35}$$

The minimum norm solution then equates to

$$\boldsymbol{p} = \begin{bmatrix} \boldsymbol{p'} \\ 1 \end{bmatrix} = \frac{\boldsymbol{E}_p \boldsymbol{E}_p^T \boldsymbol{v}_n}{\boldsymbol{v}_n^T \boldsymbol{E}_p \boldsymbol{E}_p^T \boldsymbol{v}_n} \ . \tag{36}$$

In components this equals

$$p_k = \frac{\sum_{i=p+1}^{n} e_{ki} e_{ni}}{\sum_{i=p+1}^{n} e_{ni}^2}, \tag{37}$$

or as vector equation

$$\boldsymbol{p'} = \frac{\sum_{i=p+1}^{n} e_{ni} [e_{1i}, \ldots, e_{(n-1)i}]^T}{\sum_{i=p+1}^{n} e_{ni}^2} = \frac{\sum_{i=1}^{p} e_{ni} [e_{1i}, \ldots, e_{(n-1)i}]^T}{1 - \sum_{i=1}^{p} e_{ni}^2} \ . \tag{38}$$

Where we used $\boldsymbol{E}_p \boldsymbol{E}_p^T = \mathbb{1} - \bar{\boldsymbol{E}}_p \bar{\boldsymbol{E}}_p^T$, with $\bar{\boldsymbol{E}}_p = [\boldsymbol{e}_1, \ldots, \boldsymbol{e}_p]$, in the last equality.

B Inversion of $\boldsymbol{A}$

To show that $\boldsymbol{A}$ is always regular we use the Sherman-Morrison-Woodbury Lemma [21]. It states that for a regular (n,n) matrix $\boldsymbol{Q}$, two (n,m) matrices $\boldsymbol{R}, \boldsymbol{T}$ and a regular (m,m) matrix $\boldsymbol{S}$ the combination

$$\bar{\boldsymbol{Q}} = \boldsymbol{Q} + \boldsymbol{R}\boldsymbol{S}\boldsymbol{T}^T \tag{39}$$

is regular if it can be shown that $\boldsymbol{U} := \boldsymbol{S}^{-1} + \boldsymbol{T}^T \boldsymbol{Q}^{-1} \boldsymbol{R}$ is regular. To apply this to $\bar{\boldsymbol{Q}} = \boldsymbol{A} = \alpha \mathbb{1} + \omega \boldsymbol{P}$ we rewrite $\boldsymbol{A}$ as

$$\boldsymbol{A} = \mathbb{1}_n + \bar{\boldsymbol{B}}_p \mathbb{1}_m \bar{\boldsymbol{B}}_p^T \ . \tag{40}$$

Here we dropped the constants α and ω without loss of generality. Thus we have to examine

$$\boldsymbol{U} = \mathbb{1}_m + \bar{\boldsymbol{B}}_p^T \mathbb{1}_n \bar{\boldsymbol{B}}_p \ . \tag{41}$$

The off-diagonal elements of $\boldsymbol{U}$ are given by $\boldsymbol{b}_i^T \boldsymbol{b}_j = \cos(\beta_{ij}) \leq 1$, where β_{ij} is the angle between $\boldsymbol{b}_i$ and $\boldsymbol{b}_j$. Thus $\boldsymbol{U}$ is diagonal dominant with all diagonal elements stricly positive and hence a symmetric positive definite matrix [21]. This implies $\boldsymbol{U}$ is regular. Thus we conclude that for $\alpha > 0$ the matrix $\boldsymbol{A}$ can always be inverted.

In the direct regularisation case described in Sect. 4.1 we encounter $\boldsymbol{A}_1 = \mathbb{1}_3 + \boldsymbol{d'} \mathbb{1}_1 \boldsymbol{d'}^T$. Thus we have to look at $\boldsymbol{U} = \mathbb{1}_1 + \boldsymbol{d'}^T \mathbb{1}_3 \boldsymbol{d'}$ which is simply a scalar. Hence $\boldsymbol{A}_1$ can always be inverted provided $\alpha > 0$.

References

1. H. Spies, H. Haußecker, B. Jähne and J. L. Barron: Differential Range Flow Estimation. 21.Symposium für Mustererkennung DAGM'1999. Bonn (1999) 309–316
2. R. Szeliski: Estimating Motion from Sparse Range Data without Correspondence. ICCV'88. (1988) 207–216
3. B.K.P. Horn and J.G. Harris: Rigid Body Motion from Range Image Sequences. CVGIP: Image Understanding. **53**(1) (1991) 1–13
4. B. Sabata and J.K. Aggarwal: Estimation of Motion from a Pair of Range Images: A Review. CVGIP: Image Understanding. **54**(3) (1991) 309–324
5. L. Lucchese, G.M. Cortelazzo and A. Vettore: Estimating 3-D Roto-translations from Range Data by a Frequency Domain Technique. Conf. on Optical 3-D Measurement Techniques IV. Zürich (1997) 444–453
6. M. Harville, A. Rahimi, T. Darrell, G. Gordon and J. Woodfill: 3D Pose Tracking with Linear Depth and Brightness Constraints. ICCV'99, (1999) 206–213
7. B. Lucas and T. Kanade: An Iterative Image Registration Technique with an Application to Stereo Vision. Int. Joint Conf. on Artificial Intelligence. (1981) 674–679
8. H. Haußecker and H. Spies: Motion. In Handbook on Computer Vision and Applications, Eds.: B, Jähne, H. Haußecker and P. Geißler. Academic Press. (1999)
9. M. Yamamoto, P. Boulanger, J. Beraldin and M. Rioux: Direct Estimation of Range Flow on Deformable Shape from a Video Rate Range Camera. PAMI. **15**(1) (1993) 82–89
10. L.V Tsap, D.B. Goldgof and S. Sarkar: Model-Based Force-Driven Nonrigid Motion Recovery from Sequences of Range Images without Point Correspondences. Image and Vision Computing, **17**(14) (1999) 997–1007
11. S. Vedula, S. Baker, P. Rander, R. Collins and T. Kanade: Three-Dimensional Scene Flow. ICCV'99, (1999) 722–729
12. J. Gonzalez: Recovering Motion Parameters from a 2D Range Image Sequence. ICPR'96. (1996) 82–89

13. L. Zhao and C. Thorpe: Qualitative and Quantitative Car Tracking from a Range Image Sequence. CVPR'98, (1998) 496–501
14. 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. Conf. on Optical 3-D Measurement Techniques III. Vienna (1995) 352–361
15. B. Jähne, H. Haußecker, H. Scharr, H. Spies, D. Schmundt and U. Schurr: Study of Dynamical Processes with Tensor-Based Spatiotemporal Image Processing Techniques. ECCV '98. (1998) 322–336
16. H. Haußecker, C. Garbe, H. Spies and B. Jähne: A Total Least Squares Framework for Low-Level Analysis of Dynamic Scenes and Processes. 21.Symposium für Mustererkennung DAGM'1999. (1999) 240–249
17. W. H. Press, S. A. Teukolsky, W.T. Vetterling and B.P. Flannery: Numerical Recipes in C: The Art of Scientific Computing. Cambridge University Press. (1992)
18. S. Van Huffel and J. Vandewalle: The Total Least Squares Problem: Computational Aspects and Analysis. Society for Industrial and Applied Mathematics. (1991)
19. Ch. Schnörr: Variational Methods for Adaptive Image Smoothing and Segmentation. In Handbook on Computer Vision and Applications, Eds.: B, Jähne, H. Haußecker and P. Geißler. Academic Press. (1999)
20. J. Weickert: On Discontinuity-Preserving Optic Flow. Proc. CVMR '98. (1998) 115–122
21. G. H. Golub and C. F. van Loan: Matrix Computations (3rd edition). The Johns Hopkins University Press. (1996)
22. B.K.P. Horn and B.G. Schunk: Determining Optical Flow. Artificial Intelligence **17** (1981) 185–204
23. J.R. Bergen, p. Anandan, K.J. Hanna and R. Hingorani: Hierarchical Model-Based Motion Estimation. ECCV'92. (1992) 237–252