INAUGURAL - DISSERTATION
zur
Erlangung der Doktorwürde
der
Naturwissenschaftlich-Mathematischen
Gesamtfakultät
der
Ruprecht-Karls-Universität
Heidelberg

vorgelegt von
Dipl.-Phys. Hagen Spies
aus Heidelberg

Tag der mündlichen Prüfung: 04.07.2001

Die Analyse dynamischer Prozesse in Sequenzen von Tiefenkarten

Gutachter: Prof. Dr. Bernd Jähne
Prof. Dr. Josef Bille

Dissertation
submitted to the
Combined Faculties for the Natural Sciences and for Mathematics
of the Rupertus Carola University of
Heidelberg, Germany
for the degree of
Doctor of Natural Sciences

Analysing Dynamic Processes in Range Data Sequences

presented by
Diplom-Physicist: Hagen Spies
born in: Heidelberg

Heidelberg, July 4th 2001

Referees:	Prof. Dr.	Bernd Jähne
	Prof. Dr.	Josef Bille

Zusammenfassung

In der vorliegenden Arbeit wird eine Technik zur Erfassung dynamischer Prozesse in Tiefenkarten entwickelt. Im speziellen wird das dreidimensionale Bewegungsfeld und die lokale Flächenänderung deformierbarer Oberflächen näher untersucht. Zur Bewegungsschätzung werden Bestimmungsgleichungen in Form von partiellen Differentialgleichungen für die Tiefen- und Helligkeitsdaten eingeführt. Diese Gleichungen werden simultan in einem Total Least Squares (TLS) Verfahren gelöst. Es wird gezeigt daß mit diesem Verfahren die Parameter dynamischer Prozesse in multidimensionalen Meßdaten geschätzt werden können, falls diese sich als lineare partielle Differentialgleichung formulieren lassen. Der entwickelte Algorithmus liefert neben einem Zuverlässigkeitsmaß auch Angaben über lineare Abhängigkeiten in den Daten und berechnet entsprechend eingeschränkte Lösungen. Aufgrund solcher linearer Abhängigkeiten ist es nicht generell möglich alle Parameter lokal zu schätzen. Um trotzdem dichte Parameterfelder zu erhalten wird ein Regularisierungsverfahren vorgestellt das die Information des vorhergehenden TLS Algorithmus nutzt. Als Anwendung der so regularisierten Geschwindigkeitsfelder wird gezeigt wie die Oberflächenänderung bestimmt werden kann. Nach einer Genauigkeitsanalyse der entwickelten Algorithmen und der verwendeten Meßgeräte werden sie zur Untersuchung von Pflanzenblättern eingesetzt. Diese Experimente zeigen das damit die Bewegung und das Wachstum solcher Blätter mit hoher zeitlicher und räumlicher Auflösung erfaßt werden kann.

Abstract

In this thesis a technique to estimate dynamic processes in range data sequences is developed. This includes the instantaneous velocity field (range flow) of a deformable surface and local expansion rates. For the velocity estimation novel differential constraint equations for the depth and intensity data are introduced. These constraint equations are then combined in a general total least squares parameter estimation framework. It turns out that this method can be used for a much broader class of problems where the parameters describing dynamic changes in multi-dimensional data are to be estimated. In addition to a confidence measure does the algorithm yield type measures indicating whether and to what degree there are linear dependencies in the data. Due to these dependencies the full parameter (range flow) set can usually not be computed at all observed data points. To overcome this a special regularisation scheme is developed that makes use of the determined local data structure. Surface expansion rates can then be computed locally from such regularised range flow fields. After an accuracy analysis of the presented algorithms they are applied to study living castor bean leaves. It is shown that this method can be used to investigate the movement and growth of such leaves with high spatial and temporal resolution.

To Ronja

Contents

Notation

I tried to adhere to the notation put forward here. However, I am aware that I probably failed to do so somewhere and do hereby apologise for any confusion this may have caused.

vectors $\boldsymbol{a}, \boldsymbol{x}$

normalised vectors $\hat{\boldsymbol{a}}, \hat{\boldsymbol{x}}$

matrices (including images) $\boldsymbol{A}, \boldsymbol{B}$

world coordinates X, Y, Z

intensity I

sensor or image coordinates x, y

convolution $\boldsymbol{A} * \boldsymbol{B}$

pointwise multiplication $\boldsymbol{A} \cdot \boldsymbol{B}$

partial derivatives $\dfrac{\partial a}{\partial x} = \partial_x a = a_x$

range flow $\boldsymbol{f} = [U\,V\,W]^T$

optical flow $\boldsymbol{f} = [u\,v]^T$

3D surface $\boldsymbol{s} = \boldsymbol{s}(x, y) = [X(x, y)\,Y(x, y)\,Z(x, y)]^T$

Chapter 1

Introduction

Image processing presents a powerful technique for non-invasive measurements in a huge variety of applications ranging from large scale satellite data to microscopic images. The scientific processing of images started with the analysis of single still images. It has then evolved to include time sequences of such images and the study of the spatio-temporal dynamics within them. Recently advancing sensor technology and increasing computational power made it possible to examine time series of three-dimensional data sets. Examples are volumetric data from sources such as CT, NMR or PET. Another source of time varying 3D data is the sequential acquisition of surface data. The analysis of the dynamic changes in such range data sequences is what this thesis is concerned with.

Most previous work on range sequence analysis seeks to estimate the sensor position with respect to an assumed rigid environment [e.g. Szeliski, 1988; Horn and Harris, 1991; Sabata and Aggarwal, 1991; Lucchese et al., 1997; Harville et al., 1999; Liu and Rodrigues, 1999]. The methods put forward here do on the other hand estimate the instantaneous velocity field of a moving deformable surface. This velocity field has been called *range flow* as it is computed from range data. The presented work is related to previously reported motion estimation on non-rigid surfaces using finite element models [Yamamoto et al., 1993; Tsap et al., 1999, 2000].

The study of range flow can be seen as an extension to the 2D optical flow (an approximation of the local image motion) estimation in image sequences [Haußecker and Spies, 1999]. Such image motion estimation can either be feature based, where points of interest are first segmented and then tracked through the sequence. Alternatively it is possible to determine where a region in one image moves based on the greyvalue information. Such area based approaches can again be subdivided into matching and differential methods. The former estimates the displacement by minimising an energy functional comprising the similarity of two candidate regions. Examples are cross-correlation, sum-of-squared differences and the mean absolute difference. Differential methods on the other hand compute that displacement vector which best satisfies a greyvalue continuity equation. In principle any optical flow method can be extended to work with range data. In this context the presented range flow estimation framework can be considered a differential technique.

The motivation for the presented research stems from the need for a new method to study the dynamic behaviour of plant leaves. There has been a long term cooperation between the group of Uli Schurr at the Institute for Botany and the image processing research group headed by Bernd Jähne

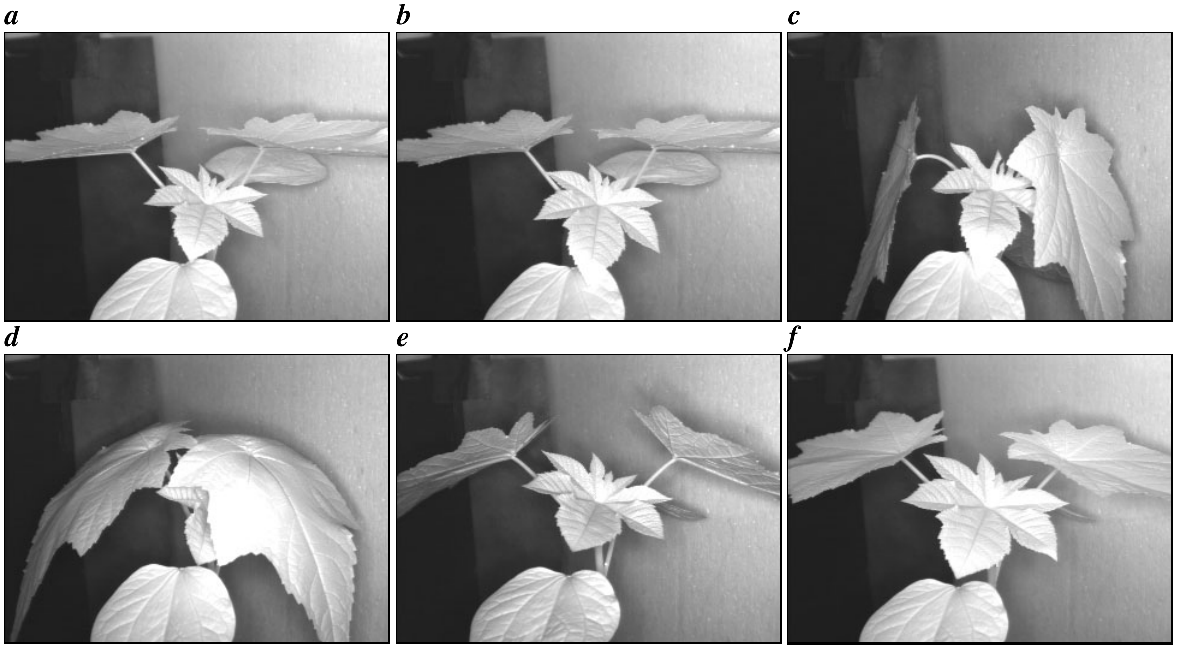


Figure 1.1: *A freely moving castor bean plant leaf during one day. The images are taken 4 hours apart.*

at the Interdisciplinary Centre for Scientific Computing, both at the University of Heidelberg, within the DFG[1] research unit *image sequence analysis to study dynamical processes* [e.g. Schmundt et al., 1998; Jähne et al., 1998; Kirchgeßner et al., 2000].

Independently John Barron at the University of Western Ontario and Albert Liptay of Agriculture Canada worked together with the aim to develop image processing tools for the automated study of plants [e.g. Barron and Liptay, 1997; Barron et al., 2000]. Both groups wanted to study the dynamics of three-dimensional leaf surfaces. This has jointly been achieved in this work, partly during a one year research stay in the laboratory of John Barron.

This study of plant leaves connects all parts of the thesis. Even though the algorithms are presented in their general form many detailed design decisions are based on this special application. Therefore the botanical context will be briefly described in the following section.

1.1 Botanical Application

For the estimation of areal growth rates, a system consisting of a single camera imaging a leaf has previously been developed [Schmundt, 1995, 1999; Schmundt et al., 1998]. The divergence of the observed velocity field then directly yields the sought growth rate. This system has then been adapted to allow routine measurements to solve fundamental botanical questions about the growth patterns found in castor bean and tobacco leaves [Walter, 2001].

However, any change in the distance between camera and leaf results in a diverging optical flow field. Such a divergence can not be distinguished from an actual growth. Therefore the motion of the

[1] German Science Foundation

leaf has to be restricted to a horizontal plane. This is achieved by pulling at the lobes so that the leaf remains more or less flat.

Figure 1.1 shows an example free leaf and its movement during one day. Obviously the unconstrained leaf undergos substantial vertical movements and bending. Thus it seems questionable if the growth observed on a constrained leaf is indeed representative for unconstrained leaves. In a number of control measurements the growth as measured by the image analysis has been compared to that found on free leaves by measuring the length of a manually applied thread to the main vein [Walter and Schurr, 2000a]. It turns out that at the onset of the night where the free leaves are bending downwards the growth is inhibited in fixed leaves. As a result the final length is smaller and an undulation of the fixed leaves occurs.

In another series of experiments it has been shown that for a special choice of pulling force ($\approx 200\,mN$) one can indeed reproduce the same growth patterns on constrained leaves as that found on freely moving leaves [Walter, 2001]. Even the same end size is reached. This result illustrates the importance of movements and mechanical stresses in plant growth [Walter and Schurr, 2000b].

As there are no such pulling forces in nature it is of great interest to study the movements and growth exhibited by unconstrained leaves. This is exactly what the present thesis is about. We will develop a method that allows to capture the 3D movements of deformable surfaces (such as leaves). From the thus available velocity field it is then also possible to extract the local expansion or growth rate.

1.2 Thesis Outline

There are two main parts to this thesis. The first one, chapters two to six, is dedicated to the development of the algorithms needed for range flow and growth estimation. This includes some necessary background on range data and its acquisition. The second part, chapters seven through nine, treats the experimental realisation of the proposed framework.

Chapter 2 starts by recapitulating the process of 3D data acquisition, in particular triangulation methods are discussed. Some example range data sets are shown and a method to interpolate missing holes in the data is presented. This is needed as typically range sensors do not yield completely dense data sets.

In Chap. 3 the concept of range flow is discussed and a novel range flow constraint equation introduced. Because this is only one equation in three unknowns further assumptions are needed. We analyse possible linear dependencies when pooling the constraints over a small local neighbourhood. Furthermore a second constraint equation on the simultaneously available intensity data is derived.

After the constraint equations have been introduced we present a general total least squares (TLS) estimation framework in Chap. 4. It is shown how this framework can be used to estimate the parameters of any linear partial differential equation describing the dynamic changes in some data array. It is demonstrated how linear dependencies can be detected and how appropriate minimal norm solutions can be computed in this case. It is also shown how the framework extends to multiple data channels and how robust statistics can be incorporated.

Due to the mentioned linear dependencies not all parameters can be computed independently at all data points. Chapter 5 introduces a regularisation scheme that exploits the structure of the TLS solution to compute dense full parameter fields. To do so a new data constraint that can be used in a variational approach is introduced. Apart from this data term does the structure of the TLS solution also allow to construct a sensible anisotropic smoothness term which leads to a new version of anisotropic diffusion.

From dense range flow fields we can locally recover the surface expansion rate of the observed surface. A formula to compute such growth rates is developed in Chap. 6 and the error in the thus estimated growth rates caused by the uncertainty in the range flow and range data is analysed.

The second part of this thesis starts with a description of the employed range sensors in Chap. 7. After explaining the experimental setup for a laser scanner and a structured light system we examine the accuracy in the measured range data.

Based on the thus available knowledge of the noise in the range data a performance analysis of the computed range flow and expansion rates is given in Chapter 8. Here first synthetic data that also contains growth and rotational movements is used. Then the range flow accuracy is analysed on a few real sequences.

First results from the botanical application are given in Chap. 9. Here both the motion and growth patterns as found on the leaves of living castor bean plants are analysed. Both constrained and unconstrained leaves are examined.

Finally, a concluding summary and possible routes for future work are given in Chap. 10.

Part I

Theory

Chapter 2

Range Data

This thesis is concerned with the estimation of the movements of 3D surfaces from range data. In the context considered here we thereby mean the world coordinates X, Y, Z and an intensity value I of an observed surface, all sampled on the same rectangular grid. The following chapter describes how such 3D data sets can be captured and what can be done to recover dense data from the sometimes sparse measurements. Such dense data sets are the input to all subsequent analysis steps that aim to compute the dynamic behaviour of thus observed surfaces.

2.1 Range Estimation

There is a wealth of different contact-free techniques that may be used to recover the three-dimensional structure of surfaces. For a recent summary see Schwarte et al. [1999] and Luhmann [2000]. Among the methods working with visible light three different measurement principles are employed: triangulation, interferometry (or optical coherent time-of-flight) and incoherent time-of-flight measurements. The algorithms presented in this thesis do in principle work with range data obtained using any such method. However, as the two sensors used for the presented experiments are both triangulation based we only describe this technique. Then the two employed sensors are discussed.

2.1.1 Triangulation

Behind all triangulation methods lies the observation that two point positions and the angles at these points are sufficient to derive the 3D position of the third point in a triangle. In computer vision this usually equates to the establishment of the positions of two corresponding points in the images of two cameras or a camera and a projector. When all camera parameters are known the 3D location of the observed point is recovered as the intersection of the two rays of sight.

In order to examine the thus achievable accuracy we look at the so called standard stereo setup where both cameras are parallel and their coordinate systems are aligned, see Fig. 2.1 a. We further assume a pinhole model with equal focal length for both cameras neglecting any geometric distortions.

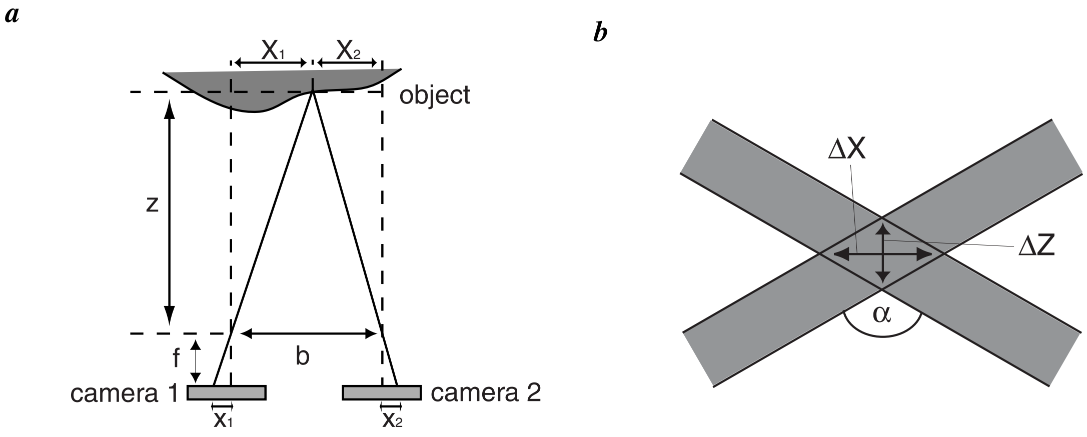


Figure 2.1: *Triangulation:* ***a*** *Triangulation using two cameras.* ***b*** *The intersection of two rays with equal uncertainty radii results in different uncertainties of the recovered intersection coordinates.*

The connection between image and world coordinates is then given by perspective projection:

$$x_i = f\frac{X_i}{Z_i}; \quad \text{and} \quad y_i = f\frac{Y_i}{Z_i}\,, \tag{2.1}$$

where $i = 1, 2$ denotes the cameras. Sensor coordinates are given in small letters and world coordinates are denoted by capitals. The general relation between the two camera coordinate systems $\boldsymbol{X}_2 = \boldsymbol{R}\boldsymbol{X}_1 + \boldsymbol{T}$ reduces to $X_2 = X_1 + b$, $Y_2 = Y_1$ and $Z_2 = Z_1$ in the standard setup. Here b denotes the *baseline*, i.e. the distance between the two projection centres. The depth then becomes:

$$Z = Z_1 = \frac{b\,f}{x_2 - x_1}\,. \tag{2.2}$$

The inverse of the so called *disparity* $d = x_2 - x_1$ is proportional to the depth ($d = \frac{b\,f}{Z}$). Under the assumption that this disparity can be estimated with an uncertainty of σ_d then the relative error in Z becomes:

$$\frac{\sigma_z}{Z} = \frac{\sigma_d}{d} = \frac{Z}{b\,f}\sigma_d\,. \tag{2.3}$$

For a fixed focal length f the relative error increases with depth Z and decreases with the baseline b. Hence we conclude that using a wide baseline and placing the observed object as close as possible in front of the sensor gives best results. However in a practical application limitations arise from the appearance of occlusions and the depth of field.

The question arises whether all three coordinates of the observed point can be estimated with the same precision. This depends very much on the geometry of the used setup. To illustrate this we examine the intersection of two rays that have the same lateral extension due to the uncertainty with which the points can be detected in the images, see Fig. 2.1 b. From similar triangles the ratio of the uncertainties is found to be:

$$\frac{\Delta X}{\Delta Z} = \tan\frac{\alpha}{2}\,. \tag{2.4}$$

Hence the intersection angle α determines the relative accuracy of the X,Y coordinates compared to that in the Z coordinate. As expected all coordinates can be obtained with the same accuracy for a vergence angle of $\alpha = 90^\circ$. However, in practical applications we often encounter angles of $\alpha = 10^\circ - 20^\circ$. This results in an accuracy ratio of $0.09 - 0.18$. Thus we conclude that the X and Y estimates are typically by one order of magnitude more accurate than the depth estimate.

It is interesting to note that apart from the described stereo system even methods such as depth from defocus and depth from focus use the same triangulation principle [Schechner and Kiryati, 2000]. But because in these cases the baseline is given by the lens aperture the relative error in the depth estimation is usually higher, see also Sect. 2.1.2.

From the corresponding points in the two images the associated 3D point is given by the intersection of the rays of sight. Due to noise these lines do not usually intersect exactly. A simple solution is given by the middle point on the line orthogonal to the two rays of sight. A better method has been given by Hartley and Sturm [1997], who also showed that the mid-point method is usually far from optimal.

The main difficulty faced by computer vision stereo algorithms is the need to precisely determine which point in the second image corresponds to a given point in the first image and vice versa. There is a huge number of methods that have been proposed to tackle this problem [e.g. Faugeras, 1993; Klette et al., 1996]. So called depth from motion approaches use a moving camera and correspondence reduces to the estimation of the image velocity. As the velocity estimation requires small changes this results in a small baseline. Therefore such measurements have to be stabilised for instance by using a Kalman filter over an entire sequence [Matthies et al., 1989; Xiong and Shafer, 1998]. The baseline can also be increased by accumulating the velocity fields between consecutive frames [Spies et al., 2000b]. However then the original velocity estimation has to be very accurate to start with [Kirchgeßner et al., 2000].

An obvious way to solve the correspondence problem even on unstructured surfaces is to use an active illumination setup. The two range sensors used in this work are both active and are now described in more detail. A discussion of the physical limits of active triangulation methods can be found in [Häusler, 1999].

2.1.2 Biris Laser Range Finder

The Biris range sensor technology developed at the National Research Council (NRC) of Canada combines two measurements into a single sensor. The first uses a double aperture mask that replaces the iris of a standard camera lens, hence the name Bi-Iris [Blais and Rioux, 1986; Rioux and Blais, 1986]. As illustrated in Fig. 2.2 this causes an object point to be seen twice in the image. There are many possible variations on the use of such a special aperture mask, examples are three pin holes [Willert and Gharib, 1992], a circular mask [Loranger et al., 1997] and a triangular mask [Gröning, 2001].

From the position of the two points the three-dimensional coordinates of the imaged object point

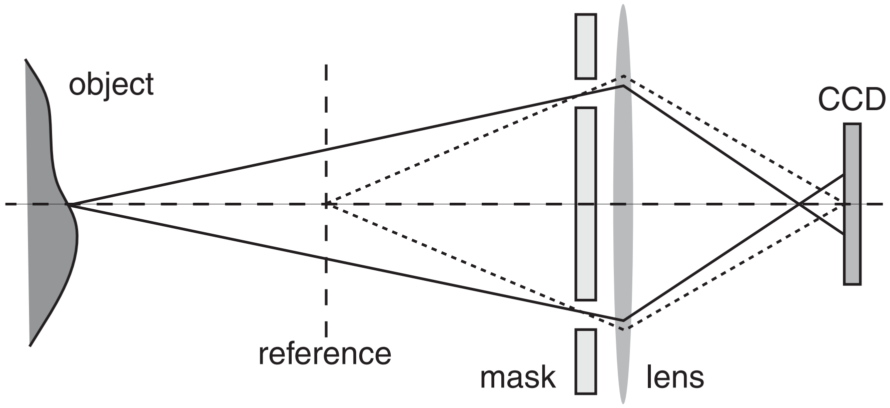


Figure 2.2: *Optical Principle of the dual aperture mask.*

can be computed. With the notation used in Fig. 2.3 the lens law yields:

$$\frac{1}{f} = \frac{1}{g} + \frac{1}{l} \qquad \rightarrow \qquad g = \frac{fl}{l-f} \, , \tag{2.5}$$

$$\frac{1}{f} = \frac{1}{g'} + \frac{1}{Z} \qquad \rightarrow \qquad Z = \frac{fg'}{g'-f} \, . \tag{2.6}$$

From self-similar triangles we find that:

$$\frac{b}{g'} = \frac{d}{g'-g} \qquad \rightarrow \qquad g' = \frac{bg}{b-d} \, , \tag{2.7}$$

with $d = p_1 - p_2$. Insertion into (2.6) results, after a few algebraic manipulations, in:

$$Z = \frac{bfl}{bf + dl - fd} = \frac{1}{l^{-1} + kd} \quad \text{with} \quad k = \frac{l-f}{bfl} \, . \tag{2.8}$$

The other coordinates are then readily obtained from the geometrical centre (x_0, y_0) of the line connecting p_1 and p_2:

$$\frac{Z}{X} = \frac{g}{x_0} \qquad \rightarrow \qquad X = \frac{x_0 Z(l-f)}{fl} \, , \tag{2.9}$$

$$\frac{Z}{Y} = \frac{g}{y_0} \qquad \rightarrow \qquad Y = \frac{y_0 Z(l-f)}{fl} \, . \tag{2.10}$$

For use as a scientific measurement device the expected errors in the estimated 3D coordinates are required. The uncertainty in x_0, y_0 and d are of the same magnitude and will be denoted as δ. From (2.9) or (2.10) we find the uncertainty in X and Y and from (2.8) that in the Z-coordinate:

$$\delta_{xy} = \frac{Z(l-f)}{fl} \delta = Z\, b\, k\, \delta \, , \quad \text{and} \quad \delta_z = \frac{\partial Z}{\partial d} \delta = k\, Z^2\, \delta \, . \tag{2.11}$$

Thus the ratio of the two uncertainties turns out to be:

$$\frac{\delta_{xy}}{\delta_z} = \frac{b}{Z} \, . \tag{2.12}$$

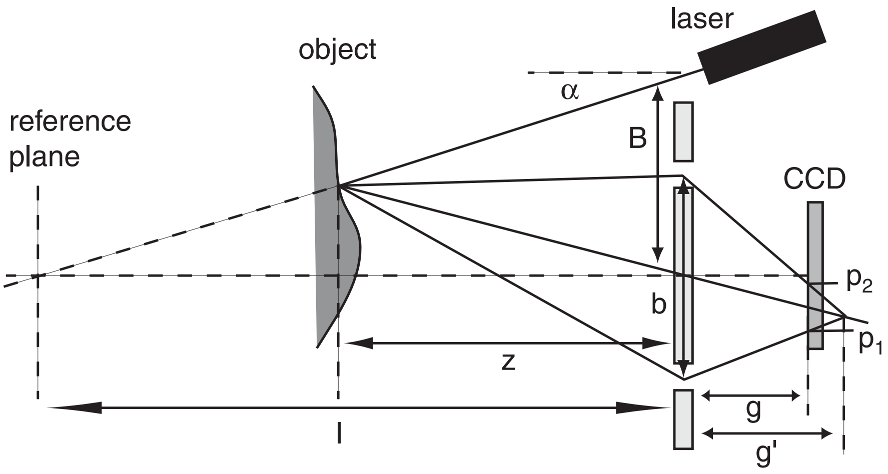


***Figure 2.3:** Sketch of the Biris laser range finder.*

For the observation of plant leaves we typically have $b \approx 20\,mm$ and $Z \approx 500\,mm$, compare Sect. 7.1. This implies that the accuracy in the depth estimation is more than one order of magnitude below that in the other coordinates. Because the 3D structure estimation solely from the line separation is not very accurate the laser is placed at a distance B on the side of the camera. This allows for another measurement based on the centre position $x_0 = \frac{1}{2}(p_1 - p_2)$ of the two lines:

$$Z = \frac{1}{\hat{l}^{-1} + k'x_0} \quad \text{with} \quad k' = \frac{l-f}{Bfl} \quad \text{and} \quad \hat{l} = \frac{B}{\tan\alpha} \,. \tag{2.13}$$

As B is in the order of $100\,mm$ this estimate is more accurate. For a final depth value both measurements (2.8) and (2.13) are weighted:

$$Z = \frac{1}{\psi(l^{-1} + kd) + (1-\psi)(\hat{l}^{-1} + k'x_0)} \,. \tag{2.14}$$

Because the second measurement is about 10 times more accurate one typically uses $\psi = 0.1$ [Blais et al., 1992]. As there are two independent measurements the combined estimate is rejected when they do not agree within the error tolerance. This combination makes the Biris sensor very robust with respect to changing illumination conditions. Two-dimensional data arrays are obtained by moving the sensor across the scene. This is usually done by high-precision linear positioners, see Sect. 7.1.

2.1.3 Structured Lighting

Instead of a single line one can simultaneously project a whole set of lines onto the observed scene. Varying scene depth results in a deformation of the lines seen by the camera (Fig. 2.4). In order to reconstruct the 3D point seen at each pixel the exact position of the corresponding line in the projector has to be known. This can be achieved by first projecting a coded light pattern [Wahl, 1986]. From the resulting binary code the position can uniquely be determined within one pixel. In order to achieve subpixel accuracy the phase-shift approach models the intensity of the line n seen by the camera as a

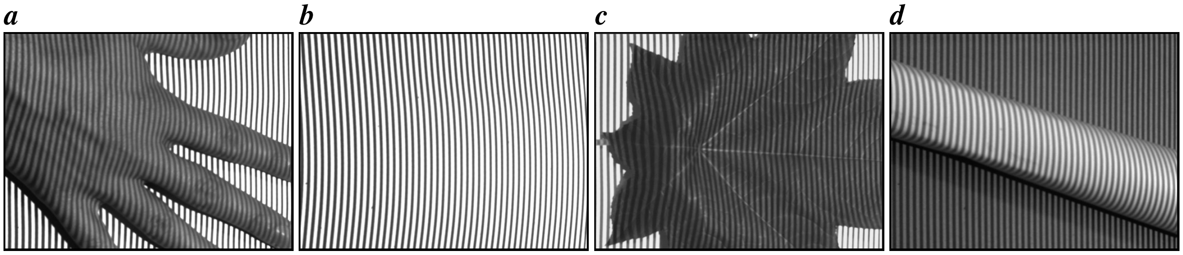


Figure 2.4: *Example images from a structured light system.*

trigonometric function [Luhmann, 2000]:

$$I_n(x,y) = I_0(1 + \gamma(x,y)\cos(\delta(x,y) + \phi_m)) \ . \tag{2.15}$$

Here I_0 is the background intensity, $\gamma(x,y)$ the line modulation, $\delta(x,y)$ the line phase and ϕ_m the phase difference. In order to estimate the unknowns I_0, $\gamma(x,y)$ and $\delta(x,y)$ at least three equations of type (2.15) are needed. This is achieved by m phase-shifts of the projected line pattern by the difference $\phi_m = (n-1)\phi_0$ with $n = 1 \dots m$. For the commonly used case of 4 such shifts the phase is given by:

$$\delta(x,y) = \arctan \frac{I_2(x,y) - I_4(x,y)}{I_3(x,y) - I_1(x,y)} \ . \tag{2.16}$$

From the known wavelength λ of the lines the depth relative to a reference plane can be recovered as $Z(x,y) = \frac{\lambda}{2 \cdot 2\pi}\delta(x,y)$. This phase shift method allows for a very accurate location on the image. Because the projected lines are usually not exactly describes by a cosine function systematic errors occur which should be corrected for [Wiora, 2001].

2.2 Missing Data

Typically range sensors do not yield completely dense data, compare Fig. 2.5a. There are a number of reasons that can cause the range estimation to fail. Examples are insufficient texture, specular reflexes, occlusion and too large inclination angles. However the movement estimation framework put forward in the next chapters relies on the availability of derivative estimates. As we compute such derivatives by means of convolution with derivative kernels dense data sets are required, compare also Sect. 3.1. Thus a pre-processing step is needed that closes the remaining holes. How this can be achieved is presented in the following.

The good news is that the range estimation processes usually identifies problematic locations. Rather then giving a useless value no value is estimated at all. This gives an initial mask indicating the valid pixel. Gross outliers are detected by thresholding the depth data to lie either in the calibrated or otherwise known range. A mask obtained in this way is shown in Fig. 2.5b. However the data near the invalid points is often corrupted as well. Because we rather remove a few good points than using false values the segmented area is slightly enlarged by means of a morphological erosion, see Fig. 2.5c for the result. In rare cases there are still some outliers left. These outliers can be removed using a

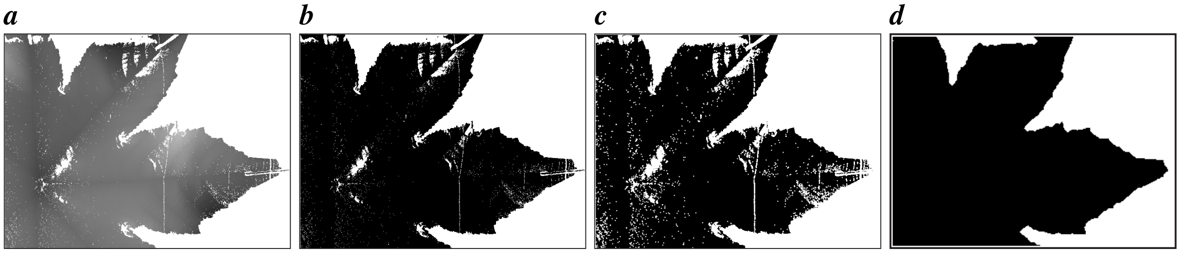


Figure 2.5: *Example missing data: **a** original depth data, **b** initial mask from the Z-value range, **c** mask after erosion and **d** leaf mask after closing the holes.*

small (here 3×3) median filter [Davies, 1990] that explicitly takes the missing data into account. This is easily done by excluding this data from the local histogram[1].

After the above steps the identified reliable measurements can be used in further processing steps. Before interpolating the remaining areas we need to identify the objects we want to study. Of course it is possible to take the complete data array. However for our main application, namely the study of plant leaves, it is desirable to segment the observed leaf first and then interpolate only this region. To segment the object of interest first a morphological closing is applied to the binary mask. This closes all but some bigger holes in the objects interior. Those holes are then filled simply by adding all connected areas with a size below a certain maximum size, typically 1000 pixel. Finally a small erosion is used in order to remove border artifacts. The resulting mask is given in Fig. 2.5d.

At this stage both the reliable measurements and the area of interest are identified. Next the data is interpolated as to completely fill this area. Towards this end we use a combination of normalised averaging and a membrane fit. Both are described separately first before the employed interpolation scheme is explained.

2.2.1 Normalised Averaging

As a result of the identification of reliable measurements we can associate a certainty ω with each measurement. In this special case these certainties only take values of zero and one, however the concept is more general. When computing an average it is sensible to use the certainties as weights. Furthermore it is well known that Box filters do not have very good properties from a signal processing point of view [Jähne, 1995]. Thus we want to incorporate another set of weights in the averaging procedure by using a filter such as a Binomial. The resulting operation has been termed normalised averaging [Granlund and Knutsson, 1995]. Here the employed filter is called the applicability $\boldsymbol{A}$. Normalised averaging is a special case of a more general filtering technique called normalised convolution [Knutsson and Westin, 1993; Westin, 1994; Farnebäck, 1999]. This interesting general technique is not described here and the reader is referred to the mentioned references. If the measurement data are denoted with $\boldsymbol{M}$ and the weight matrix with $\boldsymbol{\Omega}$, normalised averaging reads:

$$\boldsymbol{M}' = \frac{\boldsymbol{A} * (\boldsymbol{\Omega} \cdot \boldsymbol{M})}{\boldsymbol{A} * \boldsymbol{\Omega}} \,. \tag{2.17}$$

[1] A median filter returns the median as opposed to the mean of the local histogram.

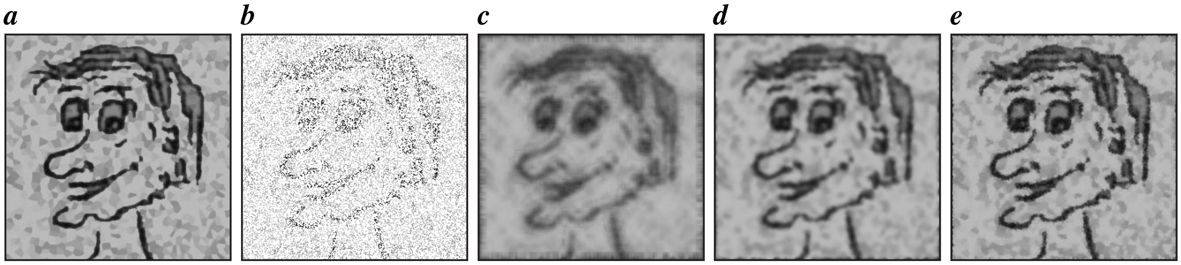


Figure 2.6: *Normalised averaging to close holes:* ***a*** *original image,* ***b*** *image with 70% of the pixel removed. Reconstruction using different applicabilities, all masks are of the size* 19×19*:* ***c*** *(2.18),* ***d*** *Binomial and* ***e*** *(2.19).*

Apart from a Gaussian the use of other applicabilities are quite possible. Examples for the elements a of $\boldsymbol{A}$, taken from Farnebäck [1999], are:

$$a = \begin{cases} \cos^2(\frac{\pi r}{16}) & \text{if } r < 8 \\ 0 & \text{else} \end{cases} \tag{2.18}$$

and

$$a = \begin{cases} 1 & \text{if } r < 1 \\ 0.5r^{-3} & \text{if } 1 \leq r < 8 \\ 0 & \text{else} \end{cases} . \tag{2.19}$$

Here r measures the distance form the centre of the applicability mask. Figure 2.6 shows the thus obtained reconstruction on a test image where 70% of the pixel have been removed. As expected it can be seen that a more localised applicability such as given in (2.19) yields a sharper image. When a low-pass filtering of the image is intended the Binomial as an approximation of a Gaussian is preferable to the function given in (2.18).

2.2.2 Membrane Fit

Let's say we have a set of measurements m with confidence values ω. In an image matrix this can also mean that the confidence is one where there is data and zero elsewhere. To compute an estimate e, in a region A, that is both close to the available data and also smooth we can minimise the following energy functional:

$$\int_A \left[\underbrace{\omega(e-m)^2 + \alpha(e_x^2 + e_y^2)}_{h(e)} \right] \mathrm{d}\boldsymbol{x} \quad \rightarrow \quad \min . \tag{2.20}$$

The considered region A could be the entire image or a previously selected subset. The smoothness is regulated via α and the model minimises the derivatives of the estimate. This is commonly termed membrane model, if a continuously varying surface normal is required this model can be extended to the second order or thin plate model [Terzopoulos, 1986, 1988]. Equation (2.20) is a first simple example of an variational approach to obtain dense and smooth data, a more general discussion will be presented in Chap. 5.

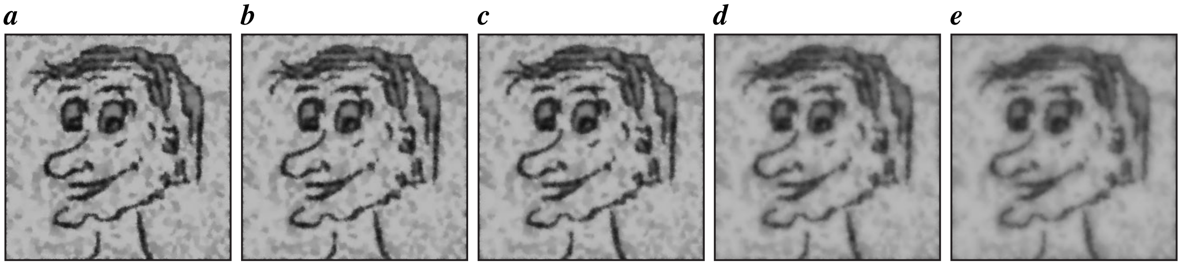


Figure 2.7:** Membrane fit to close holes using different smoothness strength:* ***a $\alpha = 0.1$*,* ***b*** $\alpha = 0.5$*,* ***c*** $\alpha = 1.0$*,* ***d*** $\alpha = 5.0$ *and* ***e*** $\alpha = 10.0$*.*

The minimum (extremum) of (2.20) is found by solving the Euler-Lagrange equations:

$$\frac{\partial h}{\partial e} - \frac{\mathrm{d}}{\mathrm{d}x}\frac{\partial h}{\partial e_x} - \frac{\mathrm{d}}{\mathrm{d}y}\frac{\partial h}{\partial e_y} = 0 \tag{2.21}$$

for each pixel. Thus we have the following equation for each pixel:

$$2\omega(e - m) - 2\alpha e_{xx} - 2\alpha e_{yy} = 0 \tag{2.22}$$

$$\omega e - \omega m - \alpha \Delta e = 0\ . \tag{2.23}$$

In a discrete implementation the Laplacian can be estimated by the difference of a local average and the central value: $\Delta e = \bar{e} - e$. When the considered area is not the complete image the local average should be computed using normalised averaging (2.17) with the confidence set to the area mask. Using this Laplacian approximation the Euler-Lagrange equations can be written as follows:

$$(\omega + \alpha)e = \alpha\bar{e} + \omega m\ , \tag{2.24}$$

which can easily be solved iteratively. An update at step $k + 1$ is found from step k using:

$$e^{k+1} = \frac{\alpha}{\omega + \alpha}\bar{e}^k + \frac{\omega m}{\omega + \alpha}\ . \tag{2.25}$$

As we are dealing with a convex energy functional the iterative process is bound to converge [Schnörr, 1999]. Thus initialization can be done with zero. However, this may not lead to a fast convergence. One could initialize with an estimate found by linearly interpolating the available data points. As seen in the next section the use of the above described normalised averaging is a good choice. The iterations are run until either a maximum number of iterations (typically 1000) are reached or until the mean change between iterations is less than a threshold (here $1 \cdot 10^{-6}$).

Figure 2.7 shows the results of applying the membrane fit to the example image with 70% removed pixel (Fig. 2.6b). The resulting reconstruction is very close to the original for small smoothness factors. For higher values of α the result resembles a smoothed version of the original data as expected. The main drawback of this method is its comparably high computational effort. Also there are occasions where a directed smoothing is preferable to isotropic smoothing, this leads to the concept of anisotropic diffusion, compare Chap. 5. However when dealing with leaves it is reasonable to assume that they can indeed be thought of as a smooth membrane.

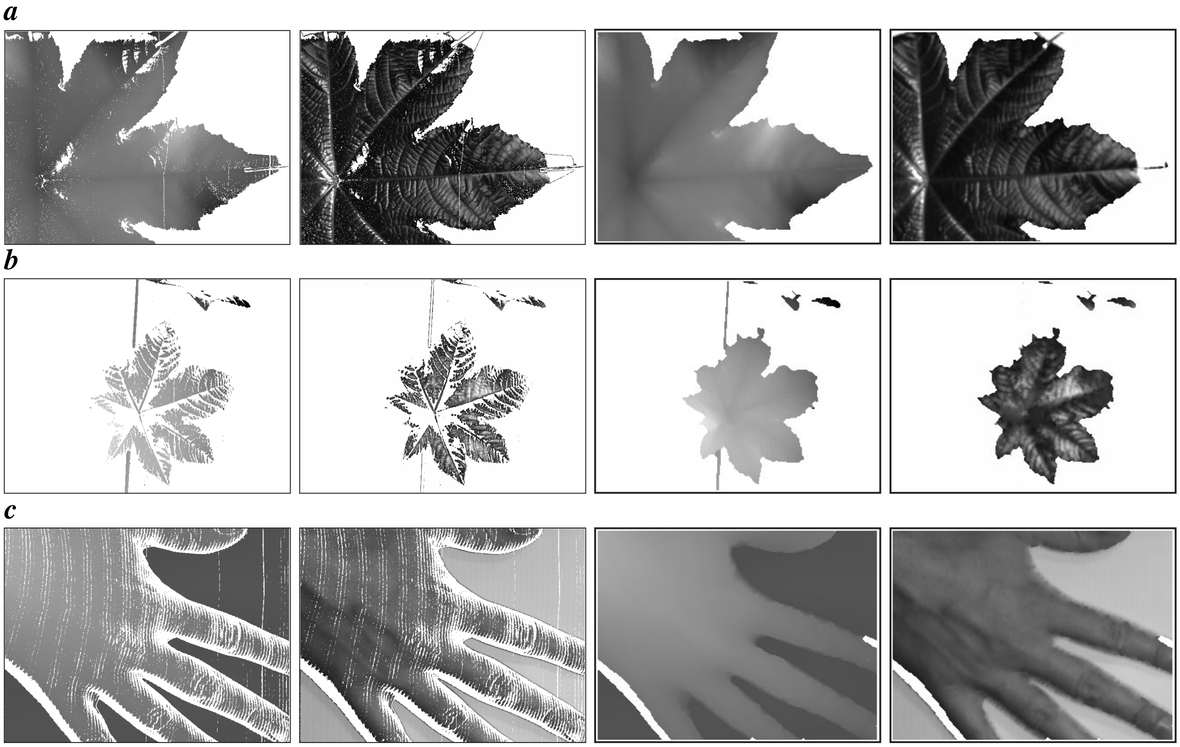


Figure 2.8: *Example raw range data and the resulting interpolated data. From left to right the original depth map, intensity, interpolated depth and interpolated intensity are shown.* ***a*** *A typical leaf taken with the structured light sensor,* ***b*** *another leaf taken with the same sensor that has rather large holes and* ***c*** *data taken with the structured light sensor of a hand.*

2.2.3 Interpolation Scheme for Leaf Data

Note that in order to reduce computational cost and noise it is favourable to do all processing on the first level of the Gaussian pyramid [Jähne, 1995]. This has the additional benefit to reduce the sometimes large displacements encountered so that they can be better estimated in a differential framework, see Chap. 9.

The main drawback of normalised averaging is that only holes which are completely covered by the applicability mask are correctly interpolated. On the other hand it can be computed much faster then an iterative membrane fit. Thus we first do a normalised averaging step where the applicability is chosen as the first level of the Gaussian pyramid, the original data forms the zeroth level. In this context we somewhat sloppy identify the convolution in (2.17) as a combined smoothing and downsampling operation.

Rather then using the thus obtained data simply for initialisation it is treated as the original data that we want to interpolate. The confidence values are processed in the same way to obtain confidence values at the pyramid level. The area mask marking the observed object (leaf) is simply downsampled. Because the averaging step usually closes most holes a subsequent membrane fit converges quickly. In most cases around 100 iterations are sufficient. As the data does already have the required smoothness a factor of $\alpha = 0.1$ is chosen.

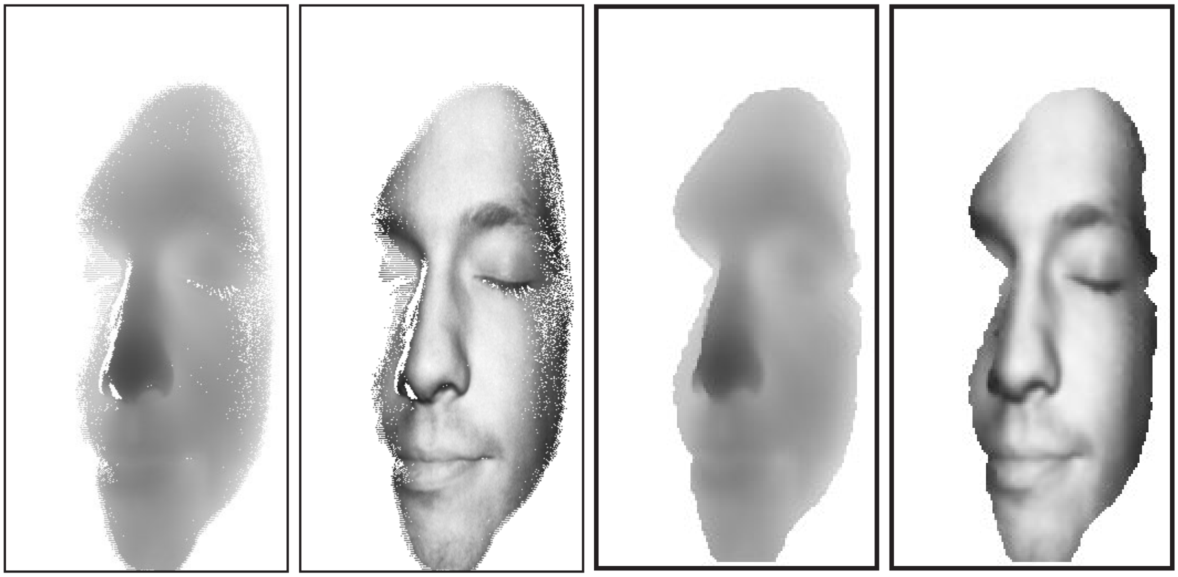


Figure 2.9: *Example raw range data of a face taken with the Biris laser range finder and the resulting interpolated data. From left to right the original depth map, intensity, interpolated depth and interpolated intensity are shown.*

Figure 2.8 shows three different examples and the resulting interpolated data for both the depth and the intensity as obtained from the structured lighting technique. The first is a typical leaf (Fig. 2.8a). The encountered holes are not very wide and can easily be closed. However it is sometimes not sufficient to rely on the normalised averaging via a Pyramid. The smaller leaf in Fig. 2.8b was measured with far more holes remaining. In this case the membrane fit is absolutely essential. From the interpolated data it can be seen that the depth data is already so smooth that the holes are hardly detectable after closing. However, as the intensity data varies more the filled holes appear much smoother than the rest of the image. Even though it is possible to close holes of that size it is certainly more advisable to attempt to reduce them in the first place by a careful selection of the measurement environment.

Figure 2.8c shows a data set of a hand also taken using the structured light sensor. It can be observed that data of this kind can be interpolated just as good as the leaf data. Finally an example face scanned using the Biris laser range finder is given in Fig. 2.9. There is no essential difference for this type of data and the gaps can neatly be filled using the technique described here.

2.3 Example Leaf Range Data

We conclude this chapter by showing some example range data sets as obtained from castor oil leaves (Fig. 2.10). The first example shows an older leaf where the bending of the lobes is clearly visible (Fig. 2.10 a). A young leaf at daytime where it is oriented relatively flat is shown in Fig. 2.10 b. We can clearly see that the region of the leaf base is strongly bent downwards. While the first two examples are taken with the structured light sensor, we give an example from the Biris sensor in Fig. 2.10 c. Because in this case the intensity data is obtained from the intensity of the laser line we have in fact a monochromatic image which appears differently. We also note that the coordinate systems of the two

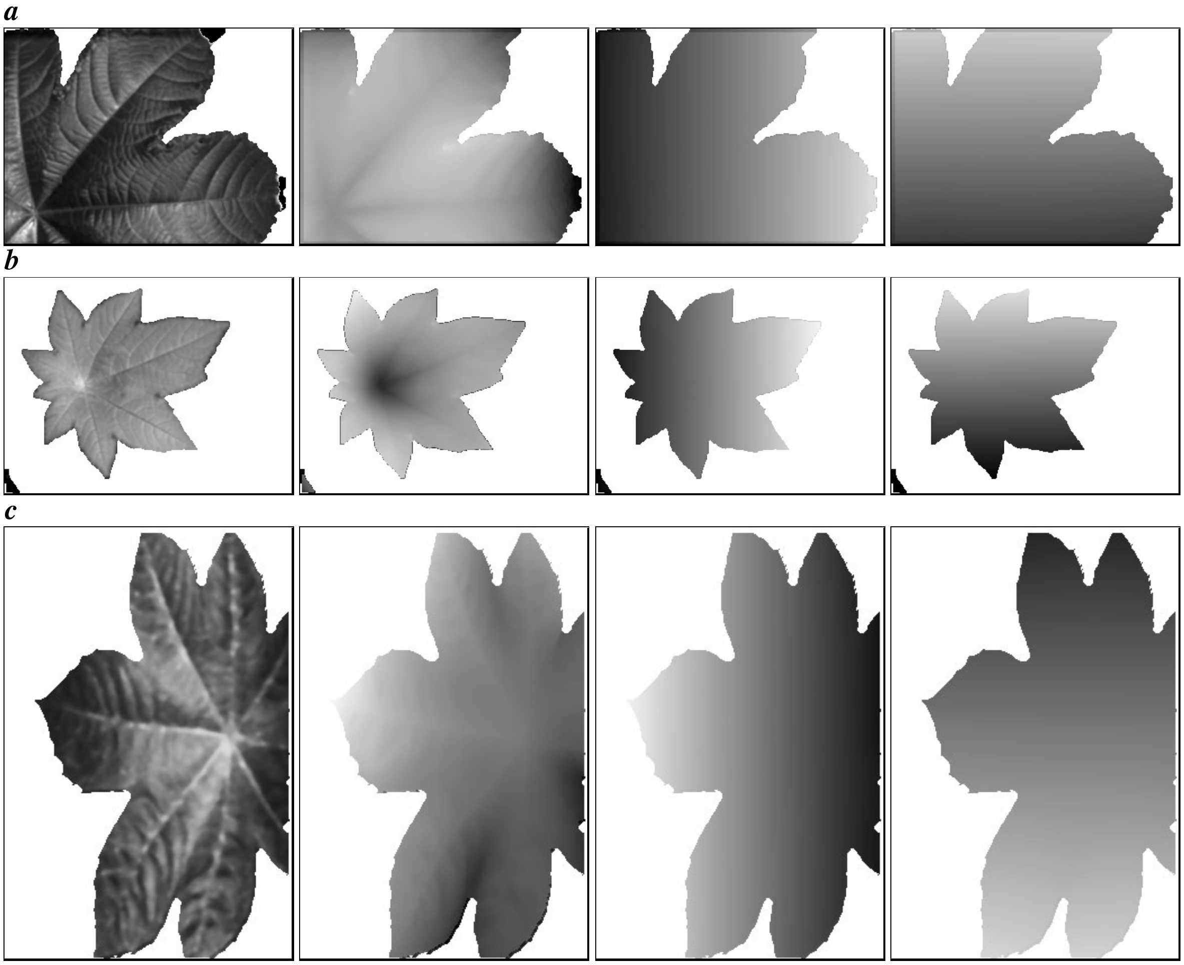


Figure 2.10: *Example range data sets of castor oil leaves, from left to right: intensity, Z-, X-, and Y-coordinate.* ***a*** *An older leaf taken with the structured light sensor,* ***b*** *a young leaf and* ***c*** *a leaf measured with the Biris scanner.*

sensors are inverted. This has to be taken into account when analysing the resulting movements.

2.4 Summary

For the estimation of range data the process of triangulation has been reviewed. The two examples used here, namely a laser range finder and structured lighting, have been discussed in more detail. As range sensors do not produce dense data we looked at methods to deal with such incomplete data sets. In particular normalised averaging and a membrane fit have been explained. The combined use of them is shown to give smooth and dense data sets with an affordable computational load. Finally some examples of leaf range data sets have been given.

Chapter 3

Range Flow

The instantaneous velocity field that describes the motion of a deformable surface is denoted *range flow*. The term range flow is used as this velocity field is derived from sequences of range data sets. Together with the 3D structure the range flow field can be used to study the dynamic changes of such surfaces. The same displacement vector field has also been called scene flow when computed directly from stereo image sequences [Vedula et al., 2000; Zhang and Kambhamettu, 2000].

This chapter introduces the fundamental differential constraint equation on range flow from range data sequences. A similar equation for the intensity data is given as well. Then possible linear dependencies, i.e. the aperture problem, for the given three-dimensional data are discussed. In order to examine the changes in the range data we need to establish derivatives. Thus a discussion of possible derivative estimation techniques for range data is given first.

3.1 Derivative Estimation

The velocity estimation algorithm presented later is based on a partial differential equation. Because range data is not necessarily evenly sampled calculating derivatives is not as straightforward as it is with image data from standard cameras. The filter kernels traditionally employed in computer vision implicitly assume the data points to be evenly spaced. The arising problem is illustrated for a range sensor that uses a pinhole camera.

> **Example:** *Standard perspective projection on the CCD chip along a scan line.* In the triangulation this is of course accounted for, yet the X-coordinates now depend on the Z-coordinates as well:
>
> $$x = f\frac{X}{Z} \;\rightarrow\; X = x\frac{Z}{f}\,. \tag{3.1}$$
>
> Here x denotes the pixel coordinate, which is intrinsically evenly sampled. If we consider two adjacent points on the image sensor we can compute the 3D displacement in X-direction between the corresponding world points, see Fig. 3.1:
>
> $$\Delta X = \frac{1}{f}(x_1 Z_1 - x_2 Z_2)\,. \tag{3.2}$$

Let us assume a locally linear depth structure with slope k: $Z_2 = Z_1 + k\Delta X$, later we'll consider the limit $Z_2 \to Z_1$, then we can express ΔX as:

$$\begin{aligned} \Delta X &= \frac{1}{f}\left(x_1 Z_1 - x_2(Z_1 + k\Delta X)\right) \ , && (3.3)\\ &= \frac{1}{f}\left((x_1 - x_2)Z_1 - x_2 k\Delta X)\right) \ . && (3.4) \end{aligned}$$

Now consider the limits $\Delta X \to \mathrm{d}X$ and $x_2 = x + \mathrm{d}x$ to obtain ($Z = Z_1$):

$$\mathrm{d}X = \frac{1}{f}\left(Z\mathrm{d}x - (x + \mathrm{d}x)k\,\mathrm{d}X\right) \ . \tag{3.5}$$

Thus we can express the change in the X-coordinate as:

$$\mathrm{d}X = \frac{Z}{f}\frac{\mathrm{d}x}{1 + \frac{1}{f}(kx + k\mathrm{d}x)} = \mathrm{d}x\frac{Z}{f(1 + k\tilde{x})} \ , \tag{3.6}$$

where we used normalised image coordinates $\tilde{x} = \frac{x}{f}$ and $\mathrm{d}x \to 0$ in the last equality. For the change in X with respect to the image coordinate x we get:

$$\frac{\mathrm{d}X}{\mathrm{d}x} = \frac{Z}{f}\frac{1}{1 + k\tilde{x}} \ . \tag{3.7}$$

In case the object is parallel to the image plane ($k = 0$) we arrive at the standard case that $\frac{\mathrm{d}X}{\mathrm{d}x}$ is constant and the X values evenly spaced. This is the implicit assumption made in the standard image processing routines. In case this assumptions does not hold we can derive a relative deviation from the even sampling and thus a systematic relative error in the derivative estimation of:

$$\sigma_{rel} = \|1 - \frac{1}{1 + k\tilde{x}}\| = \|\frac{k\tilde{x}}{1 + k\tilde{x}}\| \leq \|k\tilde{x}\| \ . \tag{3.8}$$

A typical value for the focal length is $f = 20\,mm$, the image coordinates, i.e. the size of the CCD chip is usually limited by $x < 2\,mm$. This leads to $\|\tilde{x}\| < 0.1$. If we further assume that objects are oriented at a maximum angle of $45°$ (for larger angles the range sensors are usually not able to give reliable estimates) we have $\|k\| \leq 1$. Thus we conclude that the maximum relative error is around 10% which could occur at the image borders. Figure 3.1 shows the relative error as a function of image position and object slope.

As the previous example illustrates one is well advised to take the uneven sampling into account. Range sensors might have a different geometry than the discussed perspective projection, think of a pan-tilt unit or a mirror device. We will discuss some general approaches to overcome the uneven sampling problem in the following sections.

3.1.1 Resampling

When the range data is not evenly sampled it is of course possible to resample the data on a regular grid. One can for example use the previously described membrane model (Sect. 2.2.2) [Horn and Harris, 1991] or thin plate splines [Szeliski, 1988] to interpolate the data on a regular grid. Then the condition of evenly sampled data is met again and standard derivative filters can be used. However, such a resampling implies an interpolation step which by itself introduces errors. Therefore and because convolutions are more efficient we will not consider resampling any further here.

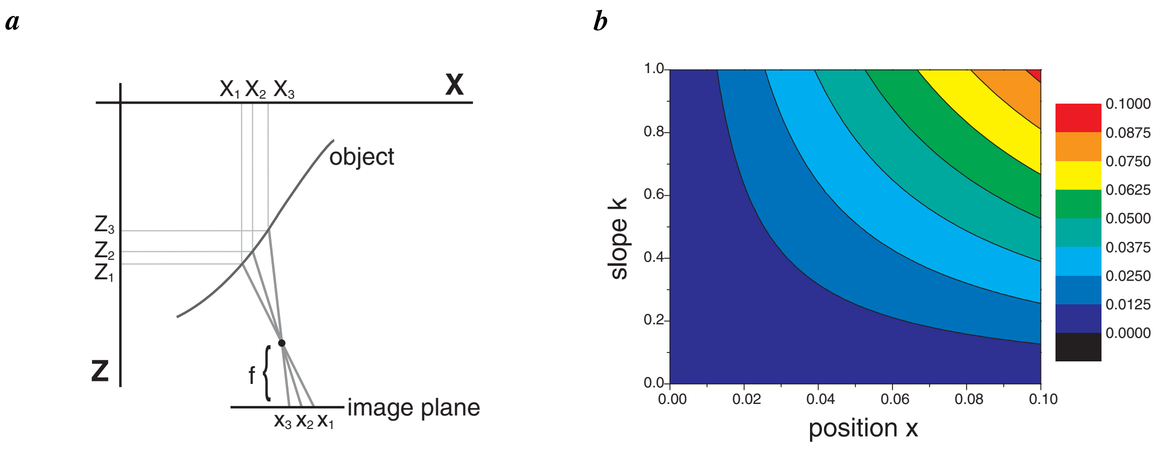


***Figure 3.1:** **a** Illustration of the sampling problem under perspective projection. **b** Relative error in the derivative estimation due to uneven sampling as a function of slope and image position.*

3.1.2 Surface Patches

Another approach to compute derivatives from range data is to locally approximate the 3D surface by a parameterised model [Chaudhury et al., 1994]. The Taylor series does provide a polynomial approximation of the surface function and its derivatives [Klette et al., 1996]. For example does the linear facet model yield an approximation of the depth function given by:

$$Z(X,Y) = p\,X + q\,Y + Z_0 \quad \text{with} \quad p = \frac{\partial Z}{\partial X} \quad \text{and} \quad q = \frac{\partial Z}{\partial Y} \,. \tag{3.9}$$

The sought parameters can easily be estimated in a small neighbourhood by a least squares technique. However, as will be seen in Chap. 4 the use of total least squares is more appropriate. How this method can be used to estimate such local surface patches is described in Sect. 4.2.1. Figure 3.2 shows an example of thus estimated derivatives based on locally planar facets. The residual (Fig. 3.2 d) of this estimation clearly shows the veins, here a higher order model would be more appropriate. We will see in Sect. 3.2 that we can actually compute the surface velocity directly from the derivatives of X, Y and Z on the sensor grid. Therefore local surface approximations are not considered any further.

3.1.3 Filters

Because of their fast application via convolutions and the availability of highly optimised filter kernels we would like to employ derivative filters. The question then becomes if the deviation due to uneven sampling can be compensated for. Here the objects of interest are 2D surfaces in the 3D world, these can be parameterised as $Z = Z(X,Y)$. The measuring process produces data points at locations on the sensor array which in turn depend on the 3D data points observed: $x = x(X,Y,Z)$; $y = y(X,Y,Z)$, e.g. (3.1). To compute the derivatives of Z with respect to world coordinates the chain rule yields:

$$\frac{\mathrm{d}Z}{\mathrm{d}X} = \frac{\partial Z}{\partial x}\frac{\mathrm{d}x}{\mathrm{d}X} + \frac{\partial Z}{\partial y}\frac{\mathrm{d}y}{\mathrm{d}X} \,, \tag{3.10}$$

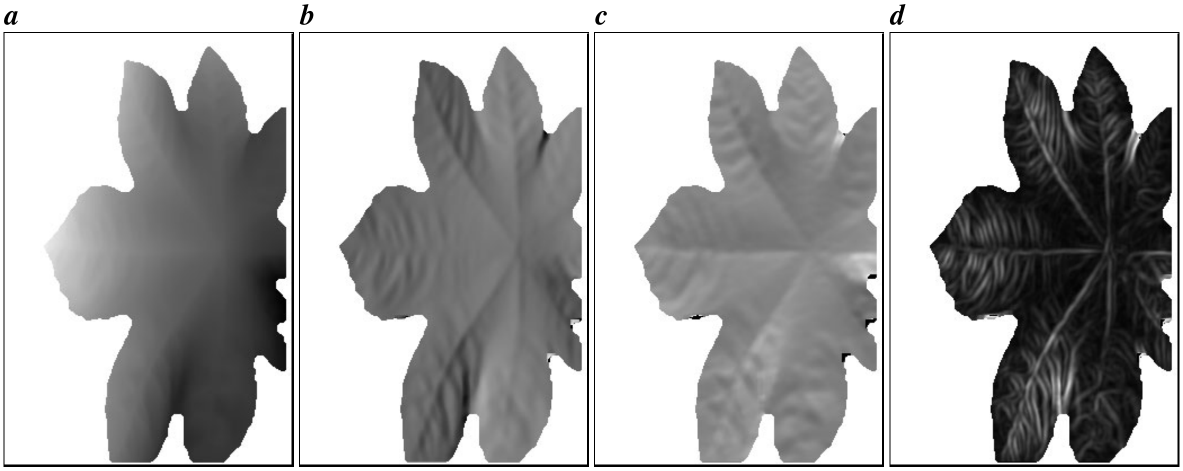


*Figure 3.2: Estimation of derivatives based on locally planar surface patches: **a** example depth map, derivatives in the range of -0.3 to 0.3 mm per mm: **b** $\frac{\partial Z}{\partial X}$, **c** $\frac{\partial Z}{\partial Y}$ and **d** residual of the fit.*

and the analogous for the other derivatives. As we can only compute the derivatives with respect to sensor coordinates using filter kernels, (3.10) can not be evaluated without the knowledge of the dependency of x and y on the world coordinates.

To overcome this problem we notice that the range sensors considered here do not yield a depth function in terms of X and Y but rather produce one data set for each of X, Y and Z on its sampling grid ($X = X(x, y)$ etc.). Here sensor coordinates are denoted by (x,y). For the total variation of the three data sets we obtain:

$$\begin{aligned} \mathrm{d}X &= \partial_x X \,\mathrm{d}x + \partial_y X \,\mathrm{d}y \,, && (3.11)\\ \mathrm{d}Y &= \partial_x Y \,\mathrm{d}x + \partial_y Y \,\mathrm{d}y \,, && (3.12)\\ \mathrm{d}Z &= \partial_x Z \,\mathrm{d}x + \partial_y Z \,\mathrm{d}y \,. && (3.13)\end{aligned}$$

Eliminating $\mathrm{d}x$ and $\mathrm{d}y$ from (3.13) results in:

$$\mathrm{d}Z = \left(\frac{\partial_y Y \partial_x Z - \partial_x Y \partial_y Z}{\partial_y Y \partial_x X - \partial_y X \partial_x Y}\right) \mathrm{d}X + \left(\frac{\partial_x X \partial_y Z - \partial_y X \partial_x Z}{\partial_y Y \partial_x X - \partial_y X \partial_x Y}\right) \mathrm{d}Y \,. \qquad (3.14)$$

As this has to be equal to $\mathrm{d}Z = \partial_X Z \,\mathrm{d}X + \partial_Y Z \,\mathrm{d}Y$ we can identify the partial derivatives of Z with respect to the other world coordinates. These awkward expressions can be considerably simplified if the sensor coordinate system is aligned with the world coordinate system, as is often the case. This implies $\partial_y X = \partial_x Y = 0$ and the partial derivatives of Z become:

$$\begin{aligned} \frac{\partial Z}{\partial X} &= \frac{\partial_y Y \partial_x Z - \partial_x Y \partial_y Z}{\partial_y Y \partial_x X - \partial_y X \partial_x Y} = \frac{\partial_x Z}{\partial_x X} \,, && (3.15)\\ \frac{\partial Z}{\partial Y} &= \frac{\partial_x X \partial_y Z - \partial_y X \partial_x Z}{\partial_y Y \partial_x X - \partial_y X \partial_x Y} = \frac{\partial_y Z}{\partial_y Y} \,. && (3.16)\end{aligned}$$

3.2 The Range Flow Constraint Equation

A time varying surface may be viewed as a depth function $Z(X, Y, t)$. If the object under consideration is made up of local planar patches then this function can be expressed by its first order Taylor series expansion:

$$Z(X,Y,t) = Z_0(t) + Z_X X(t) + Z_Y Y(t) \; , \tag{3.17}$$

where the Z_X, Z_Y denote the partial derivatives with respect to X and Y respectively. Here X, Y are the local world coordinates around the point of interest. The change in depth with time then becomes:

$$\frac{\mathrm{d}Z}{\mathrm{d}t} = \frac{\partial Z_0}{\partial t} + Z_X \frac{\mathrm{d}X}{\mathrm{d}t} + Z_Y \frac{\mathrm{d}Y}{\mathrm{d}t} + X \frac{\mathrm{d}Z_X}{\mathrm{d}t} + Y \frac{\mathrm{d}Z_Y}{\mathrm{d}t} \; . \tag{3.18}$$

Under the assumption that the infinitesimal motion of the patch is a pure translation, i.e. the slope does not change ($\frac{\mathrm{d}Z_X}{\mathrm{d}t} = \frac{\mathrm{d}Z_Y}{\mathrm{d}t} = 0$), this can be written as:

$$W = Z_X U + Z_Y U + Z_t \; . \tag{3.19}$$

The local displacements form the range flow denoted by: $\boldsymbol{f} = [U \; V \; W]^T = \frac{\mathrm{d}}{\mathrm{d}t}[X \; Y \; Z]^T$. Equation (3.19) is called *range flow constraint equation* [Yamamoto et al., 1993], the same equation has also been termed *elevation rate constraint equation* [Horn and Harris, 1991]. It is the analogon to the *brightness change constraint equation* used in optical flow calculation which we briefly recall next.

In usual optical flow or image velocity estimation in image sequences it is assumed that the brightness of an observed object does not change. Then all intensity changes in the images can be attributed to movement. For the intensity g this assumption reads as $\frac{\mathrm{d}g}{\mathrm{d}t} = 0$. This is the well known brightness change constraint equation [Horn and Schunk, 1981; Schunk, 1986]:

$$g_x u + g_y v + g_t = 0 \; . \tag{3.20}$$

There are many methods to recover the optical flow $\boldsymbol{f} = [u \; v]^T$ based on this equation. Summaries may be found in Barron et al. [1994]; Mitiche and Bouthemy [1996]; Haußecker and Spies [1999]. The method for range flow estimation presented in Chap. 4 is a general algorithm that can also be used to compute optical flow even in cases where the assumption of conserved brightness is violated.

In order to evaluate the range flow motion constraint equation (3.19) the derivatives of the depth function with respect to the other world coordinates have to be computed. As discussed in Sect. 3.1 this is not entirely straightforward for unevenly sampled data. We can compute the derivatives from a surface fit, see Sect. 3.1.2 and Sect. 4.2.1. Or use the formulas (3.15) and (3.16) based on the derivatives with respect to the sensor grid.

Following the same argument as that leading to the derivative estimates in Sect. 3.1.3 we will now derive a new version of the range flow constraint equation. The new constraint can be evaluated entirely on the sensor grid by means of convolutions. This allows to draw on the well established linear filter theory and can be implemented very efficiently. The three components of the range flow field are the total derivatives of the world coordinates with respect to time ($U = \frac{\mathrm{d}X}{\mathrm{d}t}$ etc.). Taking these derivatives yields the following equations:

$$\begin{aligned} U &= \partial_x X \, \dot{x} + \partial_y X \, \dot{y} + \partial_t X \ , && (3.21) \\ V &= \partial_x Y \, \dot{x} + \partial_y Y \, \dot{y} + \partial_t Y \ , && (3.22) \\ W &= \partial_x Z \, \dot{x} + \partial_y Z \, \dot{y} + \partial_t Z \ . && (3.23) \end{aligned}$$

Total derivatives with respect to time are indicated by a dot and partial derivatives as $\partial_x X = \frac{\partial X}{\partial x}$. As we are not interested in the rates of change on the sensor coordinate frame we eliminate $\dot{x}$ and $\dot{y}$ to obtain the range flow motion constraint expressed in sensor coordinates:

$$\frac{\partial(Z,Y)}{\partial(x,y)} U + \frac{\partial(X,Z)}{\partial(x,y)} V + \frac{\partial(Y,X)}{\partial(x,y)} W + \frac{\partial(X,Y,Z)}{\partial(x,y,t)} = 0 \ , \tag{3.24}$$

where

$$\frac{\partial(Z,Y)}{\partial(x,y)} = \partial_x Z \partial_y Y - \partial_y Z \partial_x Y \tag{3.25}$$

is the Jacobian of Z, Y with respect to x, y. Notice that the Jacobians are readily computed from the derivatives of X, Y, Z in the sensor frame obtained by convolving the data sets with derivative kernels. Equation (3.24) poses the general constraint equation independent of a particular sensor. As stated before many sensors have aligned world and sensor coordinate systems ($\partial_y X = \partial_x Y = 0$). This simplifies the range flow constraint equation considerably:

$$(\partial_x X \partial_y Y)\, W = (\partial_y Y \partial_x Z)\, U + (\partial_x X \partial_y Z)\, V + (\partial_x X \partial_y Y \partial_t Z - \partial_x X \partial_t Y \partial_y Z + \partial_t X \partial_y Y \partial_x Z) \ . \tag{3.26}$$

Not surprisingly, dividing through $(\partial_x X \partial_y Y)$ yields the initial constraint equation (3.19). But notice that the partial derivative of Z with respect to time depends on the parameterisation:

$$(\partial_t Z)_{XY} = (\partial_t Z)_{xy} + (\partial_X Z)_{XY} (\partial_t X)_{xy} + (\partial_X Z)_{XY} (\partial_t Y)_{xy} \ . \tag{3.27}$$

Here we still made the assumption of aligned sensor and world coordinate systems. In some cases one can further assume orthogonal projection onto the sensor, for instance when a telecentric lens is used. This implies: $\partial_x X = \Delta x$, $\partial_y Y = \Delta y$ and $\partial_t X = \partial_t Y = 0$, where Δx, Δy are constants. In this case the constraint equation simply becomes:

$$\frac{\partial_x Z}{\Delta x} U + \frac{\partial_y Z}{\Delta y} V - W + \partial_t Z = 0 \ . \tag{3.28}$$

3.3 The Aperture Problem

Unfortunately (3.19) or (3.24) pose only one constraint in three unknowns, this may be attributed to our initial assumption of locally planar surface patches. On a plane only the movement perpendicular to the plane may be observed which is essentially a three-dimensional version of the aperture problem known from optical flow [Jähne, 1995]. Just as the optical flow constraint equation (3.20) describes a line in (u, v)-space, so does the range flow constraint equation (3.19) describe a plane in (U, V, W)-space with surface normal $[Z_X \ Z_Y \ \ -1]^T$. The best solution that can be achieved under

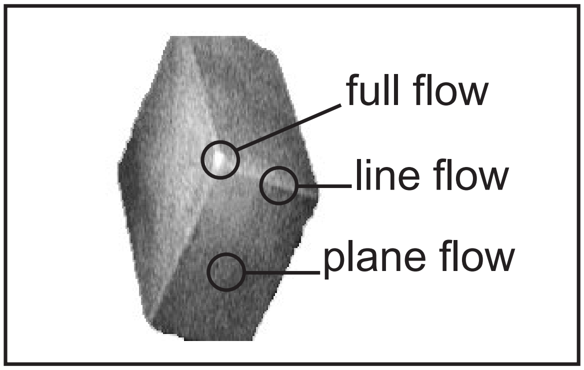


Figure 3.3: *Example range data from a cube that illustrates the three different types of neighbourhoods encountered in three dimensional data.*

these circumstances will be the minimal vector between the origin and the constraint plane. This solution defines the *raw normal flow*:

$$\boldsymbol{f}_r = \frac{-Z_t}{Z_X^2 + Z_Y^2 + 1} \begin{bmatrix} Z_X \\ Z_Y \\ -1 \end{bmatrix} . \tag{3.29}$$

In the following the linear dependencies that may occur when the range flow constraint equations are pooled over a certain region are examined. Here we only discuss range flow, however we note that for volumetric flow from time varying 3D data such as CT scans the same problems occur. The three possible types of neighbourhoods are illustrated in Fig. 3.3. On corner- or point-like structures clearly all three components of the movement can be determined even locally, such an estimate will in the following be called *full flow*. Any movement along edges or linear structures can not be resolved by a local analysis, but only the velocity in the perpendicular plane which we call *line flow*. On planar surfaces any velocity within the plane is not detectable. Only the component perpendicular to the surface can be determined, which will be denoted *plane flow*. The implications of the three region types on the constraint equations are discussed next, remember that they define planes in velocity space.

As we are dealing with three dimensional flow, at least three mutually distinct, i.e. non-parallel, planes are needed to combine the constraint equations from a local region to yield full 3D-flow. If enough such linearly independent constraint equations are available in the considered neighbourhood the full velocity vector is readily computed by the intersection of the constraint planes as illustrated in Fig. 3.4 a. Chapter 4 describes how such an estimate is obtained by means of a total least squares technique. A least squares solution for range flow is given in Appendix A.

Linear structures such as intersecting planes result in two distinct classes of constraint planes in the examined aperture, see Fig. 3.4 b. The point on the common line closest to the origin gives the appropriate line flow. This line flow lies in the plane perpendicular to the linear structure and only the motion along the direction of the structure can not be resolved. Let us look at the problem in (U, V, W)-space where the two equations define two constraint planes. For ease of notation we write

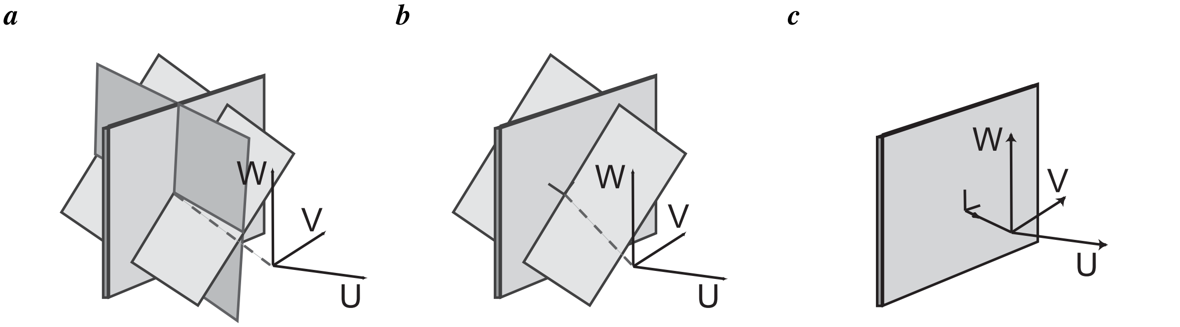


Figure 3.4: Illustration of the independent constraint equations present in a neighbourhood for ***a*** *full flow,* ***b*** *line flow and* ***c*** *plane flow.*

$a = Z_X,\ b = Z_Y,\ c = -1,\ d = Z_t$, the two constraints then become:

$$\begin{aligned} a_1 U + b_1 V + c_1 W + d_1 &= 0\ , \\ a_2 U + b_2 V + c_2 W + d_2 &= 0\ . \end{aligned} \tag{3.30}$$

These two planes describe a line and the sought solution is the point on the line with minimum distance to the origin. This point has to lie in the plane spanned by the two surface normals. This plane's normal is in direction $\boldsymbol{n}_3 = \boldsymbol{n}_1 \times \boldsymbol{n}_2$, where $\boldsymbol{n}_1, \boldsymbol{n}_2$ denote the surface normals of the two constraint planes respectively. Introducing the further abbreviations $a_3 = b_1 c_2 - c_1 b_2,\ b_3 = c_1 a_2 - a_1 c_2,\ c_3 = a_1 b_2 - b_1 a_2,\ d_3 = 0$ we arrive at a third constraint:

$$a_3 U + b_3 V + c_3 W + d_3 = 0\ . \tag{3.31}$$

In total there are now three constraints in the three unknowns x, y, z, which may be written as a linear equation system:

$$\underbrace{\begin{bmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \end{bmatrix}}_{\boldsymbol{A}} \underbrace{\begin{bmatrix} U \\ V \\ W \end{bmatrix}}_{\boldsymbol{f}} = - \underbrace{\begin{bmatrix} d_1 \\ d_2 \\ d_3 \end{bmatrix}}_{\boldsymbol{d}} \quad \rightarrow \quad \boldsymbol{f} = -\boldsymbol{A}^{-1}\boldsymbol{d}, \quad \text{if } |\boldsymbol{A}| \neq 0\ . \tag{3.32}$$

The matrix $\boldsymbol{A}$ is only invertible if the two surface normals $\boldsymbol{n}_1, \boldsymbol{n}_2$ are linearly independent. This is illuminated by looking at the dependency of the determinant on the two surface normals. Note that the surface normals on the constraint planes are identical to those on the planes in the actual range data. After some simple manipulations we obtain the determinant as:

$$|\boldsymbol{A}| = (a_1 - a_2)^2 + (b_1 - b_2)^2 + (a_1 b_2 - a_2 b_1)^2 = |\boldsymbol{n}_1 \times \boldsymbol{n}_2|^2\ . \tag{3.33}$$

The magnitude of the cross-product is proportional to $\sin(\alpha)$ where α is the angle between the two surface normals. Hence the determinant is proportional to: $|\boldsymbol{A}| \propto \sin^2(\alpha)$. As expected $\boldsymbol{A}$ becomes singular if the two surface normals are parallel.

Obviously the considered neighbourhood may be made up of a single plane. Then all constraint equations are essentially the same and we only have one averaged constraint plane, see Fig. 3.4 c. The point on this plane closest to the origin defines the velocity normal to the planar surface. This plane flow can be obtained by using averaged derivative values in (3.29). Note that by considering an entire neighbourhood such a plane flow estimate is less sensitive to noise than the raw normal flow computed via (3.29).

In Sect. 4.1.1 it is shown how such linear dependencies can be automatically detected within a total least squares estimation framework. How to compute the appropriate normal flows will be shown in Sect. 4.1.2 for the general case and in Sect. 4.3.1 for range flow in particular.

3.4 Intensity as a Second Data Source

Apart from the 3D structure information discussed so far, optical range sensors also return an intensity value of the observed surface, see Chap. 2. Clearly the thus available information should be exploited as well. In order to derive another constraint equation it is reasonable to assume that the intensity of an observed point remains the same for moderate depth changes. Thus, like for optical flow, we attribute all changes in intensity to motion in the horizontal plane. This yields another constraint equation:

$$\frac{\mathrm{d}I}{\mathrm{d}t} = \partial_x I \, \dot{x} + \partial_y I \, \dot{y} + \partial_t I = 0 \; . \tag{3.34}$$

Combined with equations (3.21) and (3.22) we obtain an additional constraint on U and V:

$$\frac{\partial(I,Y)}{\partial(x,y)}U + \frac{\partial(X,I)}{\partial(x,y)}V + \frac{\partial(X,Y,I)}{\partial(x,y,t)} = 0 \; . \tag{3.35}$$

As before this can also be written in the alternative form:

$$0 = I_X U + I_Y U + I_t \; . \tag{3.36}$$

Because we assumed the intensity is not altered by changes in the distance we can not infer any information about the vertical motion W. However the intensity constraints both U and V. In an actual range flow estimation algorithm we seek a solution that fulfills both (3.35) and (3.24). How this can be achieved in a total least squares framework will be explained in Sect. 4.1.3 and Sect. 4.3.2. Apart from making an estimation more robust due to more available data the additional intensity constraint often enables an estimate of all three range flow components in the first place.

This can nicely be illustrated in the example given in Fig. 3.5. Based on the depth data alone there are very few locations that form a corner so that a full flow estimate is possible (red spots in Fig. 3.5 c). There are also many ridges where only line flow can be found (green areas). Furthermore we find a lot of planar regions indicated in blue. This classification has been done automatically using the algorithm presented in the next chapter, see in particular Sect. 4.1.1. Taking the intensity data (Fig. 3.5 b) into account as well a full flow estimate is possible everywhere. The reason is that in this case both linear and planar structures are oriented in such a way that they already alow the computation of the vertical movement. The intensity has enough texture to enable the estimation of both horizontal velocity components. Hence a combination of the two allows to recover all three components. Clearly this does not always have to be the case and the intensity may not always help to overcome the aperture problem. Yet usually the incorporation of the intensity improves both the accuracy and density of the estimated range flow [Spies et al., 2000a], compare also Sect. 4.3.2.

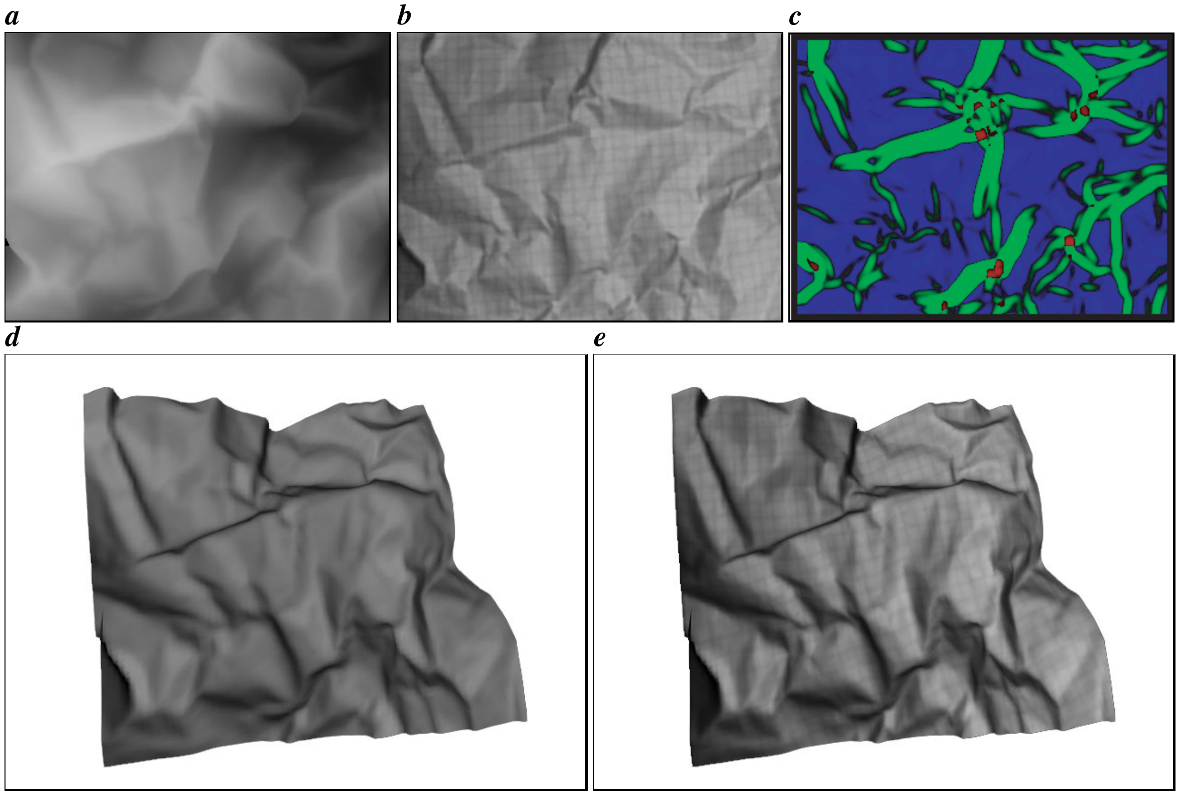


Figure 3.5: *Using the intensity data as a second data source.* ***a*** *depth data,* ***b*** *intensity data and* ***c*** *types of flow for the 3D data alone: red = full flow, green = line flow and blue = plane flow. Rendered data:* ***d*** *depth alone and* ***e*** *with intensity texture.*

3.5 Summary

As a prerequisite for range flow the problem of derivative estimation on unvevenly sampled data has been discussed. Then a differential equation that poses a constraint for the estimation of the local range flow on deformable surfaces based on sequences of range data sets has been introduced. The three versions of the aperture problem encountered in three-dimensional data have been analysed. On the usually available intensity data we could formulate an additional constraint for the X and Y components of the 3D velocity. It has been illustrated that a combination of both constraints can often help to reduce the aperture problem.

Chapter 4

TLS Parameter Estimation

The following chapter describes how range flow can be computed via the range flow motion constraint equation from a range data sequence. Towards this end a general total least squares estimation framework to fit a model to measurement data is introduced. While not restricted to, it is particularly useful for the analysis of spatially and temporally varying signals. As will be illustrated, this technique is easily amended to include multiple data channels. Another very important feature is the availability of confidence and type measures that indicate the success of the fit and reveal possible linear dependencies in the data. The presented estimation framework extends previous work on total least squares optical flow estimation [Weber and Malik, 1993, 1995; Haußecker and Jähne, 1997; Nagel and Gehrke, 1998; Haußecker and Spies, 1999].

4.1 General Concept

In scientific measurements one often encounters a linear equation for a set of model parameters $\boldsymbol{m}$ at each data point (pixel) of type:

$$\boldsymbol{a}^T\boldsymbol{m} = b\,; \quad \boldsymbol{a},\boldsymbol{m} \in \mathbb{R}^m\,. \tag{4.1}$$

Examples are the brightness change constraint equation BCCE (3.20) and the range flow motion constraint equation RFMC (3.24). Ordinary least squares (OLS) methods solve the following minimisation problem also known as linear regression [Stoer, 1999]:

$$\min\,(\tilde{b} - b)^2 \quad \text{where} \quad \tilde{b} = \boldsymbol{a}^T\boldsymbol{m}\,. \tag{4.2}$$

This implies that only b is considered erroneous while $\boldsymbol{a}$ is assumed to be exact. While this assumption is true for many applications it certainly does not hold for range flow where $\boldsymbol{a}$ contains derivatives. Therefore we introduce a new parameter vector $\boldsymbol{p}$ of dimension $n = m+1$ and consider the following minimisation problem:

$$\min\,(\boldsymbol{d}^T\boldsymbol{p})^2 \quad \text{subject to} \quad \boldsymbol{p}^T\boldsymbol{p} = c\,, \tag{4.3}$$

where the data vector is given by $\boldsymbol{d} = (\boldsymbol{a}^T, -b)^T$. In order to avoid the trivial solution we require the norm of the parameter vector to be constant ($= \sqrt{c}$), say $c = 1$. This regression has been termed total least squares (TLS) [Golub and van Loan, 1980]. The sought model parameters are then readily determined from the estimated parameter vector:

$$m_i = \frac{p_i}{p_n} \quad i = 1, \ldots , n-1 \; . \tag{4.4}$$

In the following we seek to estimate the parameters constraint by the basic TLS equation:

$$\boldsymbol{d}^T \boldsymbol{p} = 0 \; . \tag{4.5}$$

This concept is very general in the sense that the parameters of any dynamic process that can be modeled by a linear partial differential equation falls into this scheme. Since most physical, chemical, and biological processes can be described by equations of this type, it covers many applications [Haußecker et al., 1999; Spies and Jähne, 2001]. Note that the RFMC (3.24), as well as the BCCE (3.20) can be written in the form $\boldsymbol{d}^T \tilde{\boldsymbol{f}} = 0$. Where we have $\boldsymbol{d} = [g_x \, g_y \, g_t]^T$ and $\tilde{\boldsymbol{f}} = [u \, v \, 1]^T$ in the optical flow case. For range flow based on (3.24) we obtain:

$$\boldsymbol{d} = \left[\frac{\partial(Z,Y)}{\partial(x,y)} \; \frac{\partial(X,Z)}{\partial(x,y)} \; \frac{\partial(Y,X)}{\partial(x,y)} \; \frac{\partial(X,Y,Z)}{\partial(x,y,t)} \right]^T \quad \text{and} \quad \tilde{\boldsymbol{f}} = [U \, V \, W \, 1]^T \; . \tag{4.6}$$

More examples are given below.

As (4.5) gives only one constraint for every pixel we need to make further assumptions in order to solve for the complete parameter field. For a least squares estimation we pool the constraints over a local neighbourhood and assume the parameters to be constant within this area. Another possible assumption is made by variational approaches that impose smoothness constraints, such a method is described in Appendix B. The assumption of locally constant parameters leads to the following minimisation problem:

$$\boldsymbol{p} = \text{arg min} \int_{-\infty}^{\infty} w(\boldsymbol{x} - \boldsymbol{x}', t - t')(\boldsymbol{d}^T \boldsymbol{p})^2 \mathrm{d}\boldsymbol{x}' \mathrm{d}t' \quad \text{subject to} \quad \boldsymbol{p}^T \boldsymbol{p} = 1 \; . \tag{4.7}$$

Here $w(\boldsymbol{x} - \boldsymbol{x}', t - t')$ is a weighting function that defines the spatio-temporal neighbourhood where the parameters are estimated. Because we require $||\boldsymbol{p}|| = 1$ the trivial solution is avoided. This requirement can be incorporated by means of a Lagrangian multiplier, to yield the unconstrained energy functional:

$$E = \int_{-\infty}^{\infty} w(\boldsymbol{x} - \boldsymbol{x}', t - t') \left[(\boldsymbol{d}^T \boldsymbol{p})^2 + \lambda(1 - \sum_{i=1}^{n} p_i^2) \right] \mathrm{d}\boldsymbol{x}' \mathrm{d}t' \; . \tag{4.8}$$

It is known from basic calculus that this energy functional reaches a minimum when the derivatives with respect to all variables vanish. These derivatives take the following form:

$$\frac{\partial E}{\partial p_i} = 2 \int_{-\infty}^{\infty} w(\boldsymbol{x} - \boldsymbol{x}', t - t') \left[d_i(\boldsymbol{d}^T \boldsymbol{p}) - \lambda p_i \right] \mathrm{d}\boldsymbol{x}' \mathrm{d}t' \doteq 0 \; ; \quad \forall \, i = 1 \ldots n \; . \tag{4.9}$$

We are assuming a locally constant parameter field $\boldsymbol{p}$, hence the p_i can be taken out of the integral:

$$(\int d_i d_1) p_1 + \ldots + (\int d_i d_n) p_n = \lambda p_i ; \quad \forall \, i = 1 \ldots n \; . \tag{4.10}$$

For the right hand side we assumed the weight function to be normalised: $\int w = 1$. These n equations can be written in matrix form:

$$\boldsymbol{J}\boldsymbol{p} = \lambda \boldsymbol{p} \quad \text{with} \quad \boldsymbol{J} = \int_{-\infty}^{\infty} w\left(\boldsymbol{x} - \boldsymbol{x}', t - t'\right) \left(\boldsymbol{d}\,\boldsymbol{d}^T\right) \mathrm{d}\boldsymbol{x}'\mathrm{d}t' \; . \tag{4.11}$$

This is an eigenvalue equation of the extended structure tensor $\boldsymbol{J}$. For each of the n eigenvectors we get an extremum of the energy functional E. In a discrete implementation, the components of $\boldsymbol{J}$ can be readily computed using standard image processing operations applied to the data vector $\boldsymbol{d}$: $\boldsymbol{J} = \boldsymbol{B} * (\boldsymbol{d}\,\boldsymbol{d}^T)$, where $*$ denotes convolution. $\boldsymbol{B}$ is an averaging filter containing the weights w, typically Binomial or Box. $\boldsymbol{J}$ contains all the necessary information about the local spatio-temporal structure of the data, it is an extension of the structure tensor encountered in optical flow computation [Haußecker and Spies, 1999].

The energy (4.7) to be minimised can be written as follows:

$$\int_{-\infty}^{\infty} w\left(\boldsymbol{d}^T\boldsymbol{p}\right)^2 = \boldsymbol{p}^T\boldsymbol{J}\boldsymbol{p} = \lambda \; . \tag{4.12}$$

Because $\boldsymbol{p}$ is normalised this energy simply becomes the corresponding eigenvalue for each solution. It follows that (4.7) is minimised by the eigenvector $\hat{\boldsymbol{e}}_n$ to the smallest eigenvalue λ_n of $\boldsymbol{J}$.

This minimization problem (4.7) is also known as the orthogonal l_2 approximation problem and the solution coincides with the TLS-solution up to a scaling factor [Van Huffel and Vandewalle, 1991]. As $\boldsymbol{J}$ is real and symmetric the eigenvalues and -vectors can easily be computed using Jacobi-Rotations [Press et al., 1992]. Here an eigenvalue analysis is used, however for symmetric matrices, with which we are concerned, this is equal to a singular value decomposition. Thus instead of eigenvalues and eigenvectors we could also use singular values and right singular vectors.

Geometric Interpretation.

It is intuitive to examine the geometric implications of using TLS versus OLS. We have already seen that OLS considers errors only in b, but what about TLS? From the above we conclude that finding the discrete TLS solution corresponds to finding an $\boldsymbol{m} \in \mathbb{R}^m$ that minimises:

$$\frac{\sum_{i=1}^{N} |\boldsymbol{d}_i^T\boldsymbol{p}|^2}{|\boldsymbol{p}|^2} = \sum_{i=1}^{N} \frac{|\boldsymbol{a}_i^T\boldsymbol{m} - b_i|^2}{\boldsymbol{m}^T\boldsymbol{m} + 1} \; . \tag{4.13}$$

The term $\frac{|\boldsymbol{a}_i^T\boldsymbol{m} - b_i|^2}{\boldsymbol{m}^T\boldsymbol{m}+1}$ is the square of the distance from the point $[\boldsymbol{a}_i \; b_i]^T \in \mathbb{R}^n$ to the nearest point in a hyperplane P_m, which is defined by [Fischer, 1997]:

$$P_m = \left\{ [\boldsymbol{a}\; b]^T |\; \boldsymbol{a} \in \mathbb{R}^{n-1},\; b \in \mathbb{R},\; b = \boldsymbol{m}^T\boldsymbol{a} \right\} \; . \tag{4.14}$$

Hence the TLS estimation results in the ”closest” subspace (hyperplane) P_m to the data points d_i. This is best illustrated in the simple case when there is only one parameter m, i.e. fitting a straight line through the origin: $y = m \cdot x$. While OLS minimises the vertical distance to this line we see that TLS does minimise the orthogonal distance, see Fig. 4.1a.

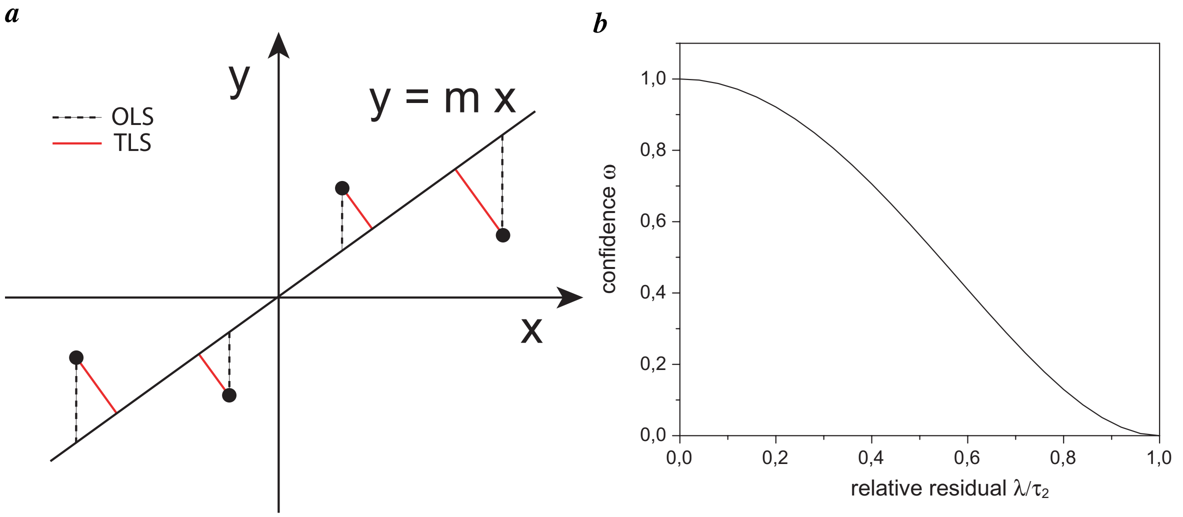


Figure 4.1: ***a*** *Illustration of OLS versus TLS for a simple line fit.* ***b*** *Confidence value as a function of the smallest eigenvalue relative to the threshold* λ_n/τ_2.

4.1.1 Confidence and Type Measure

We will now discuss how the spectrum of $\boldsymbol{J}$ can be used to define a confidence value and to detect linear dependencies in the data. For the later case we introduce type measures that indicate the extend of the dependency. This section also introduces two threshold parameters that are used in the later experiments. In the following we assume the eigenvalues of $\boldsymbol{J}$ to be sorted in descending order:

$$\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_{n-1} \geq \lambda_n \ . \tag{4.15}$$

Certainly the trace of $\boldsymbol{J}$ can be used to determine if there is enough variation (texture) in the data for the computation to make sense. We can think of the structure tensor as the covariance matrix of the data vectors ***d***, hence the trace simply measures the sum of the variances of its components. By using a threshold on the trace the case when there is no or insufficient variation in the data can be left out. As the trace is invariant under coordinate transformation it can be computed before the eigenvalue analysis is done. Depending on the data this can significantly speed up the computation, think of images that mainly contain a homogeneous background. Thus we only proceed where $tr(\boldsymbol{J}) > \tau_1$. On image and range sequences a value of $\tau_1 \approx 15$ has been found to be a good choice, if a normalised averaging filter $\boldsymbol{B}$ is used. Note that the choice of τ_1 is not very critical, its main purpose is to speed up the computation by excluding uniform areas.

As stated in (4.12) the smallest eigenvalue directly yields the residual of the parameter fit. It therefore makes sense to use this eigenvalue to reject unreliable estimates and define a normalised confidence measure. Towards this end we introduce another threshold τ_2 and reject the calculation if $\lambda_n > \tau_2$. In order to define a confidence in the remaining estimates we measure how close λ_n came to the threshold. A possible choice for a normalised confidence measure thus becomes:

$$\omega_c = \left\{ \begin{array}{cl} 0 & \text{if } \lambda_n > \tau_2 \text{ or } tr(\boldsymbol{J}) < \tau_1 \\ \left(\frac{\tau_2 - \lambda_n}{\tau_2 + \lambda_n}\right)^2 & \text{else} \end{array} \right. . \tag{4.16}$$

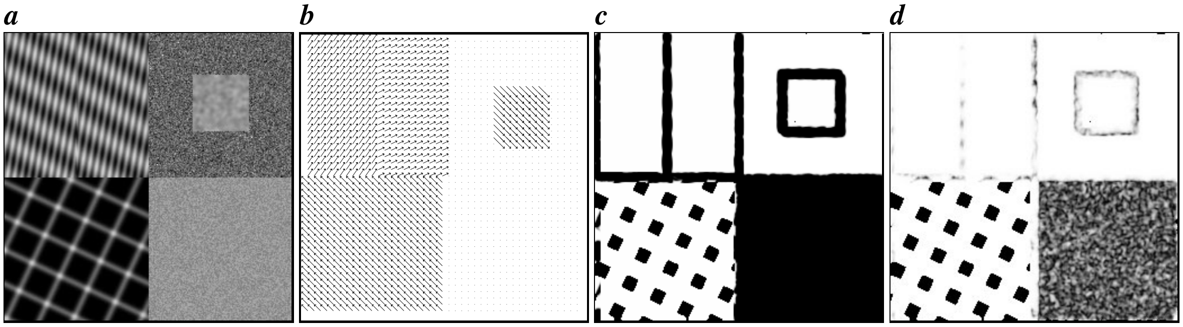


Figure 4.2: *Test image sequence, the lower right quadrant contains noise without movement:* ***a*** *center frame,* ***b*** *correct optical flow field* ***c*** *confidence measure ω_c ($\tau_2 = 0.2$) and* ***d*** *total coherence measure c_t.*

For $\lambda_n = 0$ we have a confidence of one and ω_c reaches zero for $\lambda_n = \tau_2$, see Fig. 4.1. Apart from the case where the trace falls below τ_1 this confidence measure gets small when the assumption of a locally constant parameter field is violated. This happens at discontinuities where the data can not be described by a single set of parameters. If there is no coherent motion but only noise ω_c will also vanish. This is illustrated for an optical flow example in Fig. 4.2c, we see that both the noisy quadrant and the motion discontinuities can clearly be marked. Rather than estimating a wrong solution we do not compute anything at these locations. How to deal with discontinuities in optical flow estimation is a question of ongoing research [e.g. Proesmans et al., 1994; Black, 1999; Black et al., 2000; Yu et al., 1999; Beauchemin and Barron, 2000]. Because there are no such discontinuities on leaf surfaces we will restrict us to the rejection of such problematic points for the moment. We will come back to this problem in Sect. 4.1.5 and Sect. 5.2.3.

Here we only use the residual to characterise the quality of the TLS estimation. For a more sophisticated analysis one needs to extract covariance matrices for the estimated parameters. How such covariance matrices can be obtained in a TLS framework has recently been described by Nestares et al. [2000].

In previous descriptions of the basic structure tensor algorithm a different certainty measure has been advocated [Jähne et al., 1998; Haußecker and Spies, 1999]. This *total coherence* measures the spread between the largest and the smallest eigenvalue:

$$c_t = \left(\frac{\lambda_1 - \lambda_n}{\lambda_1 + \lambda_n} \right)^2 . \tag{4.17}$$

This measure also decreases with an increasing residual, i.e. λ_n. However there is no a priori reason why λ_1 should not be far greater than λ_n even when the latter is already quite high. This is exactly what often happens at discontinuities as can be seen in Fig. 4.2d. Pure noise leads to a smaller c_t because in this case all the eigenvalues are of the same order of magnitude. Yet for the relatively high noise level the fluctuations in the eigenvalues become so great that a reliable detection of this problematic case is not possible. Due to the explained problems we will not use the total coherence measure in the rest of this thesis.

Having defined a suitable confidence measure the difficulty of having to choose the threshold τ_2 remains. In order to establish a guideline we assume that the always present noise is of additive nature following a Gaussian probability distribution. We further assume the noise on all components of the data vector to be independent and of the same variance:

$$< d_k, d_l >= \sigma_d^2 \, \delta_{kl} \; . \tag{4.18}$$

If this assumption is strongly violated other steps have to be done, compare Sect. 4.1.4. We identify the expectation $< \cdot >$ with the local average computed using the filter $\boldsymbol{B}$ in the structure tensor. Then noise simply results in an additive diagonal matrix: $\boldsymbol{J}' = \boldsymbol{J} + \sigma_d^2 1\!\!1$. The diagonalisation is done by applying suitable orthogonal matrices:

$$\boldsymbol{D}' = \boldsymbol{R}^T \boldsymbol{J}' \boldsymbol{R} = \boldsymbol{R}^T \boldsymbol{J} \boldsymbol{R} + \sigma_d^2 \boldsymbol{R}^T 1\!\!1 \boldsymbol{R} = \boldsymbol{D} + \sigma_d^2 1\!\!1 \, . \tag{4.19}$$

Hence the eigenvalues become $\lambda_i + \sigma_d^2$. Ideally the smallest eigenvalue vanishes and so we conclude that in the presence of noise its value will be given by the noise variance $\lambda_n \approx \sigma_d^2$. On this basis a threshold can be chosen, for example $\tau_2 = (3 \cdot \sigma_d)^2 = 9 \cdot \sigma_d^2$. Of course there still is the need for an informed guess about the noise variance. In practice one will often choose a threshold based on trial and error.

We present a simplified example for optical flow. Here the data vector consists of the spatio-temporal gradient of the intensity data. Under good conditions CCD cameras have a noise level in the order of the quantisation resolution $\sigma_g^2 = 1.0$ [Jähne, 1996]. A linear filter made up of n coefficients α_i is a function of the greyvalues $f(g) = \sum_{i=1}^{n} \alpha_i g_i$. From the basic law of error propagation the noise in the filter output is described by:

$$\sigma_f = \sqrt{\sum_{i=1}^{n} \left(\frac{\mathrm{d} f(g)}{\mathrm{d} g_i} \right)^2 \sigma_g} = \sqrt{\sum_{i=1}^{n} \alpha_i^2 \sigma_g} = a \, \sigma_g \; . \tag{4.20}$$

Say the derivatives are computed using a standard 5-tap derivative filter$\frac{1}{12}[-1 \;\; 8 \; 0 \;\; -8 \; 1]$. Then the noise in the gradient becomes $\sigma_f^2 \approx 0.9 \cdot \sigma_g^2$. Setting $\boldsymbol{B}$ to be a 7×7 Binomial filter acting only in the spatial domain results in $\sigma_d^2 \approx 0.05 \cdot \sigma_f^2 \approx 0.045$. Thus we conclude that $\tau_2 \approx 0.2$ is a good choice in this case, which is confirmed in practical applications.

A high confidence value does by no means ensure that all the parameters can be estimated independently from each other. The reason is that more than one eigenvalue may be close to zero ($< \tau_2$) and we can no longer uniquely pick an eigenvector as a solution. More general any vector in the nullspace of $\boldsymbol{J}$ is a possible solution. This is the aperture problem revisited, compare Sect. 3.3. We can nevertheless compute a sensible estimate as shown in Section 4.1.2. But first we want to give a type measure than indicates how well the dimensionality of the nullspace of $\boldsymbol{J}$ has been determined. This can be done by examining how much the first relevant eigenvalue λ_p is above the threshold. It is thus sensible to define:

$$\omega_t = \left(\frac{\lambda_p - \tau_2}{\lambda_p} \right)^2 , \tag{4.21}$$

as a normalised measure for each encountered type. Figure 4.3a and c show the thus computed type measures for full and normal flow on the test sequence given in Fig. 4.2. We see that the normal type measure is only high on the linear structures that occur on the grid in the lower left quadrant. The full type measure on the other hand is high on the grid intersections.

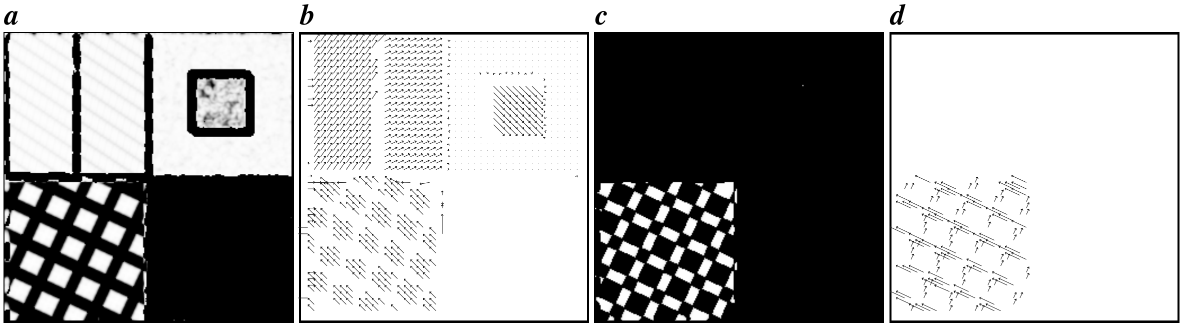


*Figure 4.3: Type measures and corresponding optical flow fields: **a** full flow type measure, **b** estimated full flow field **c** normal flow type measure and **d** estimated normal flow.*

4.1.2 Minimum Norm Solutions

We present a way to estimate a parameter vector even when there are linear dependencies in the data. As stated above any vector in the nullspace of $\boldsymbol{J}$ is a possible solution. The only sensible choice is that resulting in an estimate with minimal norm - here we always use the Euclidean norm. 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 (4.7):

$$\boldsymbol{p} = \sum_{i=p+1}^{n} g_i \hat{e}_i = \boldsymbol{E}_p \boldsymbol{g} \quad \text{where} \quad \boldsymbol{E}_p = [\hat{e}_{p+1}, \ldots, \hat{e}_n] = \begin{bmatrix} e_{(p+1)1} & \cdots & e_{n1} \\ \vdots & \ddots & \vdots \\ e_{(p+1)n} & \cdots & e_{nn} \end{bmatrix} . \tag{4.22}$$

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

$$\|\boldsymbol{p}\| = \boldsymbol{g}^T \boldsymbol{E}_p^T \boldsymbol{E}_p \boldsymbol{g} = \boldsymbol{g}^T \boldsymbol{g} = \sum_i g_i^2 . \tag{4.23}$$

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

$$p_n = \left(\sum_{i=p+1}^{n} g_i \hat{e}_i \right)_n = \sum_{i=p+1}^{n} g_i e_{in} = \boldsymbol{v}_n^T \boldsymbol{E}_p \boldsymbol{g} = 1 , \tag{4.24}$$

where $\boldsymbol{v}_n = [0, \ldots, 0, 1]^T$. Equations (4.23) and (4.24) can be combined using a Lagrange multiplier to give an energy functional to be minimised:

$$E(\boldsymbol{g}) = \boldsymbol{g}^T \boldsymbol{g} + \lambda(\boldsymbol{v}_n^T \boldsymbol{E}_p \boldsymbol{g}) \rightarrow \min . \tag{4.25}$$

The minimum is found by setting the partial derivatives of E with respect to the g to zero. Doing so yields:

$$2g_i + \lambda e_{in} = 0 \quad \rightarrow \quad g_i = -\frac{\lambda}{2} e_{in} \quad \rightarrow \quad \boldsymbol{g} = -\frac{\lambda}{2} \boldsymbol{E}_p^T \boldsymbol{v}_n . \tag{4.26}$$

Substitution into (4.24) gives:

$$\lambda = \frac{-2}{\boldsymbol{v}_n^T \boldsymbol{E}_p \boldsymbol{E}_p^T \boldsymbol{v}_n} . \tag{4.27}$$

The minimum norm solution then equates to:

$$\boldsymbol{p} = \boldsymbol{E}_p \boldsymbol{g} = \frac{\boldsymbol{E}_p \boldsymbol{E}_p^T \boldsymbol{v}_n}{\boldsymbol{v}_n^T \boldsymbol{E}_p \boldsymbol{E}_p^T \boldsymbol{v}_n} \quad \text{or} \quad p_k = \frac{\sum_{i=p+1}^{n} e_{ik} e_{in}}{\sum_{i=p+1}^{n} e_{in}^2} \; ; \quad k = 1 \dots n \; . \tag{4.28}$$

It is sometimes advantageous to express this solution in terms of the eigenvectors corresponding to the non-vanishing eigenvalues. To do so we note that $\boldsymbol{E}_p \boldsymbol{E}_p^T = \mathbb{1} - \bar{\boldsymbol{E}}_p \bar{\boldsymbol{E}}_p^T$, with $\bar{\boldsymbol{E}}_p = [\boldsymbol{e}_1, \dots, \boldsymbol{e}_p]$. Then (4.28) becomes:

$$\boldsymbol{p} = \frac{(\mathbb{1} - \bar{\boldsymbol{E}}_p \bar{\boldsymbol{E}}_p^T) \boldsymbol{v}_n}{\boldsymbol{v}_n^T \boldsymbol{E}_p \boldsymbol{E}_p^T \boldsymbol{v}_n} \; . \tag{4.29}$$

In components this equals:

$$p_k = \frac{\sum_{i=1}^{p} e_{ik} e_{in}}{1 - \sum_{i=1}^{p} e_{in}^2} \; ; \quad k = 1 \dots n-1 \quad \text{and} \quad p_n = \frac{1 - \sum_{i=1}^{p} e_{in}^2}{1 - \sum_{i=1}^{p} e_{in}^2} = 1 \; . \tag{4.30}$$

The actual parameter vector $\boldsymbol{m} = [p_1 \dots p_{n-1}]^T$ can also be expressed in a vector equation:

$$\boldsymbol{m} = \frac{\sum_{i=p+1}^{n} e_{in} [e_{i1}, \dots, e_{i(n-1)}]^T}{\sum_{i=p+1}^{n} e_{in}^2} = \frac{\sum_{i=1}^{p} e_{in} [e_{i1}, \dots, e_{i(n-1)}]^T}{1 - \sum_{i=1}^{p} e_{in}^2} \; . \tag{4.31}$$

Thus the parameter solution can be expressed as a linear combination of the ”reduced” eigenvectors $\hat{\boldsymbol{b}}_k$. By reduced we mean those vectors obtained from the first $n-1$ elements of the eigenvectors: $\hat{\boldsymbol{b}}_k = \frac{1}{\sum_{i=1}^{n-1} e_{ki}^2} [e_{k1} \dots e_{k(n-1)}]^T$. In case only the smallest eigenvalue vanishes, i.e. all parameters can be estimated, (4.31) becomes ($p = n-1$):

$$\boldsymbol{m} = \frac{e_{nn} [e_{n1}, \dots, e_{n(n-1)}]^T}{e_{nn}^2} = \frac{1}{e_{nn}} \begin{bmatrix} e_{n1} \\ \vdots \\ e_{n(n-1)} \end{bmatrix} \; . \tag{4.32}$$

This is consistent with the previously given solution (4.4). The minimum norm solutions for full and normal flow on the test sequence of Fig. 4.2 are given in Fig. 4.3b and c. Apart from some boundary effects they agree quite well with the correct flow field Fig. 4.2b.

4.1.3 Multichannel Parameter Estimation

The use of multichannel image sequences can greatly help in the estimation of velocity fields and associated parameter values. The TLS estimation technique put forward here can not only handle one data channel but is readily extended to incorporate multiple independent data sources. Especially for range flow estimation we treat depth and intensity as two independent channels. Other examples include colour images, or images taken in the thermal and visible waveband. Even sources like sonic data or mechanical measurements could be coupled. In the following it is assumed that the different data channels are sampled on the same grid.

In order to ensure no a priori bias between the channels we also require that all data channels have been scaled such that their values are in the same order of magnitude. This can be done by subtracting the mean and adjusting the data to have unit variance in all channels. This conversion to unit normal densities is well known from statistical analysis and is used whenever distinct data types have to be compared [Rice, 1995]. This obviously assumes standard distributions in all channels which is not very realistic. However for the purpose of adjusting the range of values it is sufficient.

For each individual channel c the basic model equation (4.5) is written as:

$$\boldsymbol{d}_c^T \boldsymbol{p}_c = 0 \; . \tag{4.33}$$

In general not all parameters may be estimated in all channels and there might also be different values, e.g. a decay constant might be different for different wavelengths. For ease of notation we rewrite each data and parameter vector such that they all are of equal size and that the parameters are consistently ordered. This is done by introducing zeros at the appropriate locations. For a combined estimation we then state the following minimisation problem, compare (4.7):

$$\boldsymbol{p} = \text{arg min} \int_{-\infty}^{\infty} w(\boldsymbol{x}-\boldsymbol{x}', t-t') \sum_c \beta_c (\boldsymbol{d}_c^T \boldsymbol{p})^2 \mathrm{d}\boldsymbol{x}' \mathrm{d}t' \quad \text{subject to} \quad \boldsymbol{p}^T \boldsymbol{p} = 1 \; . \tag{4.34}$$

Additional weights β_c can be used to favour some data channels against others. This is for instance useful when they have a different signal to noise ratio. Following the same procedure as before we conclude that the solution is determined by an eigenvalue analysis of the combined structure tensor:

$$\boldsymbol{J}' = \int_{-\infty}^{\infty} w(\boldsymbol{x}-\boldsymbol{x}', t-t') \; \sum_c \beta_c (\boldsymbol{d}_c^T \boldsymbol{d}_c) \; \mathrm{d}\boldsymbol{x}' \mathrm{d}t' = \sum_c \beta_c \boldsymbol{J}_c \; . \tag{4.35}$$

The combined tensor is directly given by the weighted sum of the tensors for the individual channels. For the case of range flow, see Sect. 4.3, we will see that the incorporation of intensity in addition to the depth data both enhances the accuracy and reduces the aperture problem [Spies et al., 2000a]. Let's assume that the noise on all individual tensors is of the same magnitude. How this can be achieved is the subject of the next section. Then the smallest eigenvalue will be in the order of $\lambda_n = \sum_{c=1}^{C} \sigma_c^2 = C\sigma^2$, where C gives the number of channels used. Clearly the threshold τ_2 has to be adjusted accordingly.

4.1.4 Tensor Equilibrium

For an unbiased and optimal TLS estimate the errors in all data terms need to be independent random variables with zero mean and equal variance [Van Huffel and Vandewalle, 1991; Mühlich and Mester, 1998]. If that is not the case we can compute an improved estimate using a technique called equilibrium. Here we only give a brief summary of this concept and refer to Mühlich and Mester [1999] for further details and examples. Usually the total least squares problem for N data points is written in the form:

$$\underbrace{\begin{bmatrix} \boldsymbol{d}_1^T \\ \vdots \\ \boldsymbol{d}_N^T \end{bmatrix}}_{\boldsymbol{D}} \boldsymbol{p} \doteq 0 \quad \text{subject to} \quad \boldsymbol{p}^T \boldsymbol{p} = 1 \; . \tag{4.36}$$

The solution is then given as the right singular vector to the smallest singular value of $\boldsymbol{D}$. As shown before this equals the eigenvector of the structure tensor $\boldsymbol{J} = \boldsymbol{D}^T\boldsymbol{D}$. Equilibration consists of transforming $\boldsymbol{D}$ into $\boldsymbol{W_l}\boldsymbol{D}\,\boldsymbol{W_r}$ with appropriate non-singular square weight matrices $\boldsymbol{W_l}$ and $\boldsymbol{W_r}$. Note that a multiplication with non-singular matrices does not change the rank of $\boldsymbol{D}$. It does however change the error metric in the minimisation problem.

First we treat the case of pure right hand equilibrium ($\boldsymbol{W_l} = \mathbb{1}$). Let the noise in the data be expressed in terms of an additive error matrix: $\boldsymbol{D} = \boldsymbol{D}_0 + \boldsymbol{N}$, where $\boldsymbol{D}_0$ is the correct data matrix. The requirement on the errors then reads as $< \boldsymbol{N}^T\boldsymbol{N} >= c\ \mathbb{1}$, where $< \cdot >$ denotes the expectation value and c is a constant. Applying $\boldsymbol{W_r}$ yields:

$$< \boldsymbol{N'}^T\boldsymbol{N'} >= \boldsymbol{W_r}^T < \boldsymbol{N}^T\boldsymbol{N} > \boldsymbol{W_r} \doteq c\ \mathbb{1}\,. \tag{4.37}$$

Thus we can meet the requirement by choosing $\boldsymbol{W_r}$ as the inverted root of the covariance matrix: $\boldsymbol{W_r}\boldsymbol{W_r}^T = < \boldsymbol{N}^T\boldsymbol{N} >^{-1}$. Although pure right hand equilibrium is sufficient to ensure an unbiased solution we still want to apply a left hand equilibrium to weight the rows of $\boldsymbol{D}$. Then $\boldsymbol{W_r}$ has to be chosen as the inverse root of $< \boldsymbol{N}^T\boldsymbol{W_l}^T\boldsymbol{W_l}\boldsymbol{N} >$. Note that the previously introduced weights in the structure tensor computation correspond to a diagonal left-hand equilibration matrix with $\boldsymbol{W_l} =$ diag$(\sqrt{w_i})$. Here the w_i are the discrete values of the weight function $w(\boldsymbol{x} - \boldsymbol{x}', t - t')$ used in (4.11).

In this thesis only diagonal weight matrices are used, i.e. the errors are assumed to be independent. Yet one has to be aware that because the derivative kernels act on the same noise source the intensity derivatives are correlated to some extend. This has also been pointed out by Nagel [1995]; Ng and Solo [1998]; Fermueller et al. [1999]. Applying diagonal equilibration matrices can be interpreted as row and column scaling of the data matrix $\boldsymbol{D}$. This procedure is well known and is often used to obtain numerically more stable results [Golub and van Loan, 1996]. In our special case we choose $\boldsymbol{W_l}$ to put an individual weight on the data points, this is usually Gaussian, see also Sect. 4.1.5. The right hand equilibration matrix is chosen as diag$[\frac{1}{\sigma_i}]$ where σ_i is the standard deviation of the data term d_i. Note the diagonal normalised weight matrix $\boldsymbol{W_l}^T\boldsymbol{W_l}$ does not change the expectation value. In order to retrieve the correct values the result p_i has then to be scaled with σ_i.

Clearly the thresholds introduced in Sect. 4.1.1 have to be adjusted when equilibration is used. As the standard deviation of all data values is now equal to unity we choose $\tau_2 = (3\sigma)^2 = 9$. In case of uniform data the trace will be $n\sigma^2$, where n is the dimension of the structure tensor. Because we only want to consider areas with significant variation we might choose $\tau_1 = 10\ n\sigma^2 = 10\ n$ as threshold.

Using a diagonal equilibration matrix with some very large values is an effective way to deal with error-free rows or columns in $\boldsymbol{D}$. However a more sophisticated and stable method to deal with such cases has been given by Demmel [1987].

Example: *Fitting a Straight Line.* The benefit gained from equilibration can be nicely illustrated for the simple example of fitting a straight line to some data points. Let the line be described by $y = m \cdot x + b$. For TLS this is recast as:

$$[x\ 1\ -y]^T \begin{bmatrix} m \\ b \\ 1 \end{bmatrix} = 0\,. \tag{4.38}$$

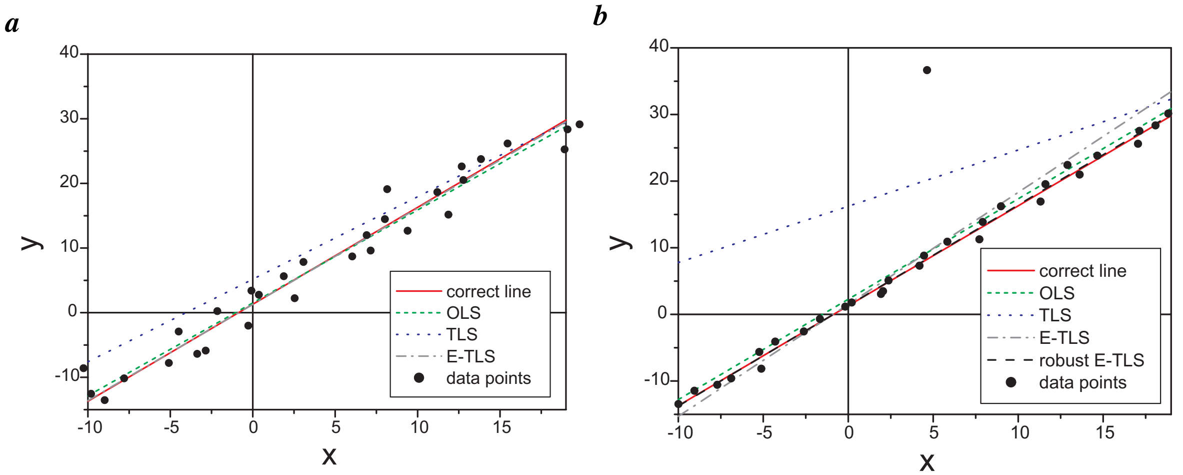


Figure 4.4: *Fitting a straight line:* ***a*** *A comparison of OLS, TLS and equilibrated TLS.* ***b*** *Comparison of the fits in the presence of an outlier.*

Clearly there is no error on the constant 1 in ***d***. As mentioned above such cases can be handled in practice by assuming a very low standard deviation. In a numerical experiment we add white noise with $\sigma_x = \sigma_y = 0.5$ and use a scaling factor of 1000 on the constant column. The correct parameters were set to $m = 1.5$ and $b = 1.3$ and 30 data points were used. After repeating the experiment for 100 times we get the following mean parameter estimates:

	m	b
OLS	1.493 ± 0.018	1.34 ± 0.19
TLS	1.483 ± 0.018	1.62 ± 0.20
E-TLS	1.498 ± 0.018	1.32 ± 0.19

We can observe that using equilibrium (E-TLS) gives significantly better results than the normal algorithm (TLS). We also note that in this case the use of total least squares does not give a noticeable advantage over ordinary least squares (OLS). This is somewhat surprising as both x and y contain errors and OLS only minimises the errors in y. Therefore we conduct another experiment with non-symmetric and higher noise, namely $\sigma_x = 1.5$ and $\sigma_y = 0.5$. The results for 100 tries then become:

	m	b
OLS	1.458 ± 0.044	1.49 ± 0.46
TLS	1.345 ± 0.219	4.08 ± 4.20
E-TLS	1.498 ± 0.046	1.30 ± 0.47

First we observe that for TLS without equilibrium the estimation of b becomes unstable. Also we can see that equilibrated TLS now performs somewhat better than ordinary least squares as expected. One example fit out of this experiment is shown in Fig. 4.4a, we see that OLS and E-TLS are nearly indistinguishable while TLS is clearly a lot worse. Actually this difference increases with higher noise levels. However one has to be aware that for low noise levels OLS is probably just as good in most applications, see also Sect. 4.2.2.

Another question is the algorithms robustness with respect to cross outliers. Figure 4.4b indicates that unscaled TLS becomes utterly useless in this case. The reason here is that the order of the eigenvalues changes. Here using the the eigenvector to the second smallest eigenvalue would give a far better result, but there is no way to detect such a case automatically. Both OLS and E-TLS are still able to give reasonable results. How the result can be further approved upon in the presence of outliers will be discussed in the next section.

4.1.5 Including Robust Statistics

The problem with outliers in a least squares estimation framework is that due to the error norm used they can have an arbitrarily high influence on the result. Recall that least squares, be it OLS or TLS, is only optimal when the underlying error distribution is Gaussian [Gelb, 1974]. The aim of robust statistics is to reduce or even eliminate the influence of the outliers and perform the regression only on the bulk of the data [Huber, 1981; Hampel et al., 1986]. These techniques have successfully been applied to a number of computer vision task [Meer et al., 1991]. The use of robust estimators for optical flow computation has been pioneered by Black and Anandan [1990, 1991]; Black [1992].

Stewart [1997] gives a review on the main robust techniques used in computer vision. There are a number of robust regression methods that can be used for motion analysis like least median of squares [Bab-Hadiashar and Suter, 1997; Sim and Park, 1998; Ong and Spann, 1999]. This is currently extended to the general parameter estimation framework presented here [Garbe, 2001]. Other methods include the Hough transform [Kälviäinen et al., 1992; Nesi et al., 1995] and random sampling [Torr, 1995]. Here we briefly discuss the approach based on influence functions also termed M-Estimators [Hampel et al., 1986]. This approach is chosen as it neatly fits into the presented TLS algorithm.

We rewrite our basic estimation problem (4.5) as:

$$\sum_i \rho\left(\boldsymbol{d}_i^T \boldsymbol{p}\right) \quad \rightarrow \quad \text{min} \;, \tag{4.39}$$

where ρ is the chosen estimator. The influence a particular data point has on the solution is determined by the derivative ψ of the estimator. Before we have used a quadratic estimator:

$$\rho\left(x\right) = x^2 \;, \qquad \psi\left(x\right) = 2\,x \;. \tag{4.40}$$

In this case the influence of outliers increases linearly. For a robust estimator this influence has to be limited. This is best achieved by so called redescending estimators for which the influence of outliers decreases and finally reaches zero for large values. There are a number of such estimators, see Black [1992] for a collection. In this work we use Tukey's biweight [Black, 1992]:

$$\rho(x,r) = \begin{cases} \frac{r^4x^2}{2} - \frac{r^2x^4}{2} + \frac{x^6}{6} & |x| < r \;, \\ \frac{r^6}{6} & else \;. \end{cases} \qquad \psi(x,r) = \begin{cases} x(r^2-x^2)^2 & |x| < r \;, \\ 0 & else \;. \end{cases} \tag{4.41}$$

Figure 4.5 shows the different behaviour of this estimator and the quadratic estimator. The additional parameter r determines for which residual the influence reaches zero and has to be chosen according to the problem at hand.

In order to solve the robust estimation problem it can be converted into an iteratively re-weighted least squares problem. Here the weights are used to reduce the influence of bad data points. A similar weighted TLS approach with weights based on the orthogonal distance from a least median

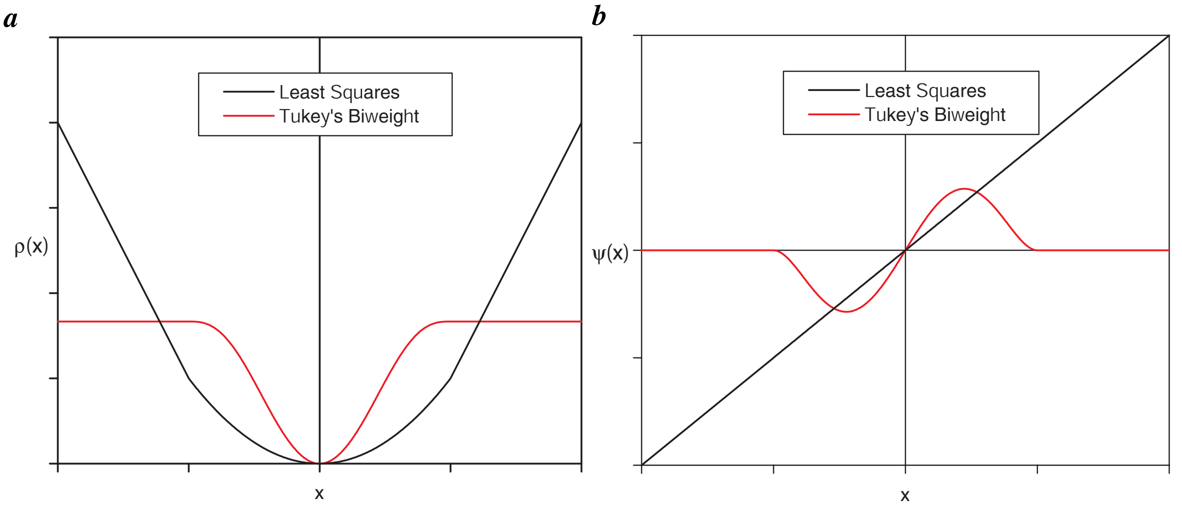


Figure 4.5: *Comparing LS and Tukey's biweight M-Estimator:* ***a*** *Estimator* $\rho(x)$ *and* ***b*** *influence function* $\psi(x)$.

of squares solution has been presented by Bab-Hadiashar and Suter [1998]. We start by writing the robust problem (4.39) as an equivalent weighted least squares problem:

$$\sum_i \rho\left(\boldsymbol{d}_i^T \boldsymbol{p}\right) = \sum_i \omega_i \left(\boldsymbol{d}_i^T \boldsymbol{p}\right)^2 \tag{4.42}$$

To get the minimal solution the derivatives are taken on both sides:

$$\sum_i \psi\left(\boldsymbol{d}_i^T \boldsymbol{p}\right) = 2\sum_i \omega_i \left(\boldsymbol{d}_i^T \boldsymbol{p}\right) = 0\ . \tag{4.43}$$

Thus the weights have to be chosen as:

$$\omega_i = \frac{\psi\left(\boldsymbol{d}_i^T \boldsymbol{p}\right)}{\left(\boldsymbol{d}_i^T \boldsymbol{p}\right)}\ . \tag{4.44}$$

If we want to keep the stronger weighting of the central pixels we may use the product of the weights $w_i' = \omega_i w_i$ to compute the structure tensor (4.11). Initially we simply employ the originally chosen weight function. From a thus obtained estimate the new weights are then compute from (4.44). This process is iterated until the residual (λ_n) drops below a set threshold, say τ_2, or a fixed number of iterations are reached. Note that on images we can significantly reduce the computational cost by only performing the robustification where the initial estimation did not give a satisfactory result, i.e. $\lambda_n > \tau_2$. However we may want to choose a slightly lower threshold to test wether the residual can be further reduced in such cases as well.

Applying the robust technique to fit the straight line to data containing one outlier returns very good parameters: $m = 1.509$ and $b = 1.33$, see Fig. 4.4b. In order to use this technique on image sequences we have to consider that it is quite possible that the robust error norm eliminates too many data points. As no sensible estimation is possible under these circumstances this has to be detected.

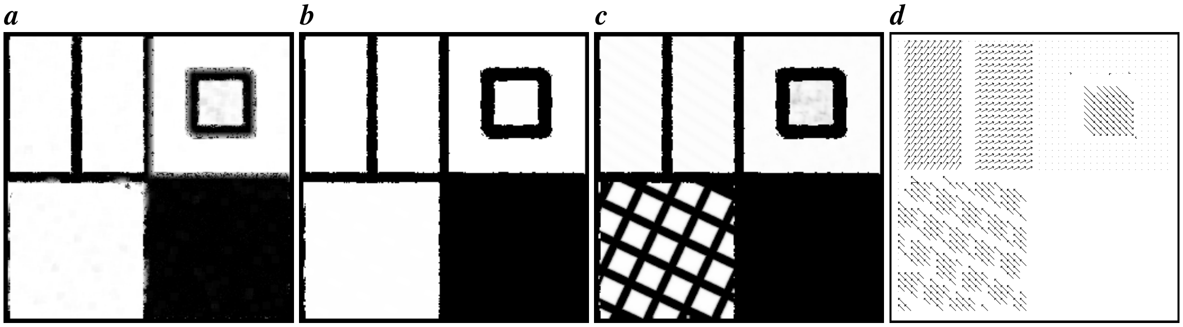


Figure 4.6: Results using robust statistics for the test sequence of Fig. 4.2: A count measure, ***b*** *confidence measure,* ***c*** *full flow type measure and* ***d*** *estimated full flow.*

A straightforward count measure for N data points is given by:

$$\bar{c} = \frac{1}{N}\sum_{i=1}^{N} c_i \quad \text{with} \quad c_i = \begin{cases} 1 & r_i \leq 0.5r \\ 2 - \frac{2}{r} r_i & 0.5r < r_i < r \\ 0 & r_i \geq r \end{cases} . \tag{4.45}$$

Here the r_i denote the individual residuals at each data point. As M-Estimators do not tolerate too many outliers we require $\bar{c} > 0.8$ in order to accept the result. Techniques that are more robust, i.e. have a higher breakpoint, are currently being investigated [Garbe, 2001]. The results of applying this scheme on the optical flow sequence given in Fig. 4.2 are shown in Fig. 4.6. We see how the count measure drops in the vicinity of discontinuities. Due to the relatively high threshold on this count measure there is only a minor increase in the overall density. Yet we see that the previously erroneous velocity estimations near some of the boundaries are now correct, compare Fig. 4.2b.

On the other hand we note that in the case of range flow estimation on plant leaves the preprocessing usually removes all outliers, see Sect. 2.2. Hence there is no need to use the robust method in this application. Yet in the examination of the sensor accuracy this robust fit is employed, see Sect. 7.3.

4.2 Example Applications

There exists a great number of potential applications for the presented parameter estimation framework. Here we will first describe how surface patches in range data can be estimated. Then we will recall the ordinary optical flow problem and illustrate how some parameters from dynamic physical processes may be incorporated. Finally we discuss the estimation of range flow in greater detail.

4.2.1 Surface Patches

The facet model for images expresses local neighbourhoods in terms of some basis functions [Haralick and Watson, 1981; Haralick, 1984]. Usually a polynomial is chosen and for instance the cubic facet model reads:

$$I(x,y) = a_1 + a_2 x + a_3 y + a_4 x^2 + a_5 xy + a_6 y^2 + a_7 x^3 + a_8 x^2 y + a_9 xy^2 + a_{10} y^3 . \tag{4.46}$$

The coefficients a_i are then found using a least squares fit. As Farnebäck [1999] points out the results can be greatly improved upon by using weighted least squares, with Gaussian weights. This is an example where the use of OLS is clearly appropriate as the locations of the pixels (x, y) are error-free. At least this is usually the premise made in computer vision. Every linear least squares fit can be cast as a convolution which makes the estimation of the facet parameters computationally efficient. This observation is at the heart of a technique called normalised convolution, where the image data is locally projected onto a set of basis functions taken the data's certainty into account [Knutsson and Westin, 1993].

But what about range data? Mc Ivor [1995] uses TLS in single range data sets to estimate the surface normals. In a similar approach we consider the Taylor series expansion of the depth values up to the first order, i.e. locally planar patches:

$$Z(X+\Delta X, Y+\Delta Y, t+\Delta t) = Z(X,Y,t) + \frac{\partial Z}{\partial X}\Delta X + \frac{\partial Z}{\partial Y}\Delta Y + \frac{\partial Z}{\partial t}\Delta t\,. \tag{4.47}$$

As opposed to the above given facet model example we extend the approximation to the temporal domain. Due to the data acquisition, see Chap. 2.1, at least X and Y are also error-prone and even the sampling rate might not be known accurately. Therefore TLS is the estimation technique of choice in this case. Equation (4.47) is recast for TLS as $\boldsymbol{d}^T\tilde{\boldsymbol{f}} = 0$ with:

$$\boldsymbol{d} = [\Delta X\ \Delta Y\ \Delta t\ \Delta Z]^T \quad \text{and} \quad \tilde{\boldsymbol{f}} = [Z_X\ Z_Y\ Z_t\ 1]^T\,, \tag{4.48}$$

where $\Delta Z = Z(X,Y,t) - Z(X+\Delta X, Y+\Delta Y, t+\Delta t)$. This can of course be easily extended to incorporate higher orders. Using such a local approximation of the surface gives a possibility to compute the derivatives needed in (3.19). An example has already been given in Sect. 3.1.2. However as outlined in Sect. 3.1 we rather use optimised derivative kernels and convolutions. In Sect. 7.3 we will fit a plane to complete data sets of planar targets in order to determine the accuracy of the sensor used.

4.2.2 Optical Flow

The proposed method to compute the structure tensor for optical flow estimation is by no means the only way to arrive at this tensor. Other approaches use quadrature filters [Bigün and Granlund, 1987; Granlund and Knutsson, 1995], a projection onto polynomial basis functions [Farnebäck, 2000a,b] or the autocovariance function of the signal [Mester, 2000]. Apart from an eigenvalue analysis the sought solutions can also be determined from the minors of the structure tensor [Barth, 2000].

We already used an optical flow example throughout this chapter to illustrate the presented concepts. Thus we won't discuss optical flow much further but rather show how it can be extended within our framework.

Before we treat possible extensions we want to explicitly give the formula for normal flow (4.31). In classical optical flow estimation we have $n = 3$. In case of an aperture problem only the largest eigenvalue is significant, $p = 1$. The minimum norm solution, i.e. the normal flow, then becomes:

$$\boldsymbol{n} = \frac{e_{13}}{1-e_{13}^2}\begin{bmatrix} e_{11} \\ e_{12} \end{bmatrix} = \frac{e_{13}}{e_{11}^2+e_{12}^2}\begin{bmatrix} e_{11} \\ e_{12} \end{bmatrix}\,. \tag{4.49}$$

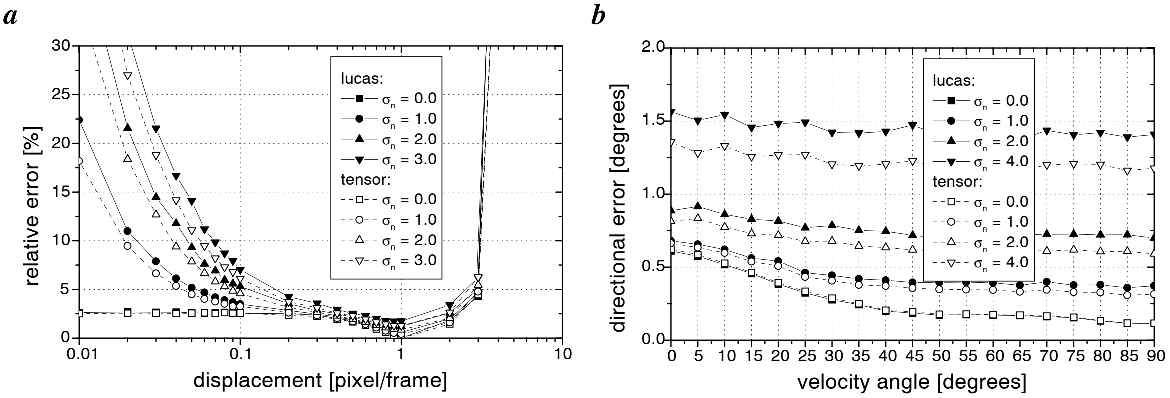


Figure 4.7: *Comparison of TLS and OLS on a random dot test sequence:* ***a*** *relative error dependent on speed and noise,* ***b*** *directional error dependent on direction and noise (displacement magnitude fixed at 0.4 pixel/frame).*

Note that often this normal flow is given with the opposite sign [Granlund and Knutsson, 1995; Haußecker and Spies, 1999]. There the formula is heuristically derived from the argument that in this case we encounter a plane in the spatio-temporal image data. The eigenvector then defines the normal of this plane. However the sign of this plane normal is a matter of choosing coordinate systems. Here a right handed system is chosen, placing the origin in the upper left corner of the image stack. Using the lower left corner, as usual in graphics, we obtain a left handed coordinate system. This changes the sign of the plane orientation. In other words the normal points in the opposite direction.

OLS versus TLS.

In the context of optical flow estimation it is also interesting to compare the performance of OLS and TLS against each other. Towards this end we use an artificial test image that is generated from a random dot pattern. Sequences with movement are then computed from the Fourier spectra via the shift theorem as suggested by Bainbridge-Smith and Lane [1997]. Figure 4.7 shows a comparison of an ordinary least squares approach according to Lucas and Kanade [1981] and the structure tensor technique operating on such a sequence. Without additional noise both perform equally well. When adding noise of increasing standard deviation to the images the total least squares method turns out to be more accurate. Mainly for small displacements (< 0.1 pixel/frame) the relative error increases slightly less for the total least squares approach. However, the benefit of using total least squares becomes more evident for the error in the estimated flow direction as shown in Fig. 4.7b, where a displacement of 0.4 pixel/frame is used.

Parametric Flow Models.

A first extension of the standard structure tensor is to drop the requirement of locally constant flow. It is common to model the velocity field in a local neighbourhood by an affine model [Fleet, 1992;

Farnebäck, 2000a]:

$$\boldsymbol{f} = \boldsymbol{t} + \boldsymbol{A}\boldsymbol{x} = \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} + \begin{bmatrix} a_3 & a_4 \\ a_5 & a_6 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} . \tag{4.50}$$

We can then insert this model into the brightness change constraint equation $\boldsymbol{\nabla g f} = 0$. The resulting parameter and data vectors in our estimation framework are:

$$\boldsymbol{d} = [g_x \; g_y \; xg_x \; yg_x \; xg_y \; yg_y \; g_t]^T \; ; \qquad \tilde{\boldsymbol{f}} = [a_1 \; a_2 \; a_3 \; a_4 \; a_5 \; a_6 \; 1]^T \; . \tag{4.51}$$

Obviously one can also choose higher order models for the local flow field [Klette et al., 1996]. Also it is quite possible to choose different models according to the local structure of the data [Black and Jepson, 1996]. Clearly we can develop the flow field also in another basis other than polynomial, if the application motivates that. If we are mainly interested in a particular aspect of the flow field, such as it's divergence or rotation, we may want to only include this in our estimation. For example we can measure growth by the divergence d of the optical flow field. One-dimensional growth in the x-direction, as found in roots, result in:

$$\boldsymbol{d} = [g_x \; g_y \; xg_x \; g_t]^T \; ; \qquad \tilde{\boldsymbol{f}} = [u \; v \; d \; 1]^T \; . \tag{4.52}$$

In the same way we can determine the two-dimensional growth rate from the divergence, i.e. the trace of $d = \mathrm{tr}\boldsymbol{A}$, directly using the following data and parameter vectors:

$$\boldsymbol{d} = [g_x \; g_y \; \frac{xg_x + yg_y}{2} \; g_t]^T \; ; \qquad \tilde{\boldsymbol{f}} = [u \; v \; d \; 1]^T \; . \tag{4.53}$$

4.2.3 Optical Flow with Brightness Changes

What if there are brightness changes other than those caused by movement? In Fig. 4.8 two examples of a moving Gaussian bell with first an additional exponential decay and second a diffusion are shown. The correct movement is to the lower right, but we get the optical flow shown in Fig. 4.8c and g. Obviously we need to extend the optical flow constraint if we want to be able to deal with such data [Haußecker et al., 1999; Haußecker and Fleet, 2000; Spies and Jähne, 2001].

Towards this end the intensity is allowed to vary along the trajectories we are estimating. If this variation can be expressed in terms of a model, the brightness change equation becomes:

$$g_x u + g_y v + g_t = f(g, t, \boldsymbol{a}) \; . \tag{4.54}$$

Here f captures additional changes that may depend on the greyvalue, time and a set of model parameters $\boldsymbol{a}$. Prominent physical examples are:

Source terms. Source terms cause first-order temporal changes in the image sequence proportional to the source strength q:

$$f(g, t, \boldsymbol{a}) = q \quad \rightarrow \quad \boldsymbol{d} = [g_x \, g_y \, 1 \, g_t]^T \quad \text{and} \quad \tilde{\boldsymbol{f}} = [u \, v \; - q \, 1]^T \; . \tag{4.55}$$

Source terms occur for example in infrared images if an object is heated or cooled or if the global illumination of a scene changes [Garbe et al., 2000; Garbe, 2001].

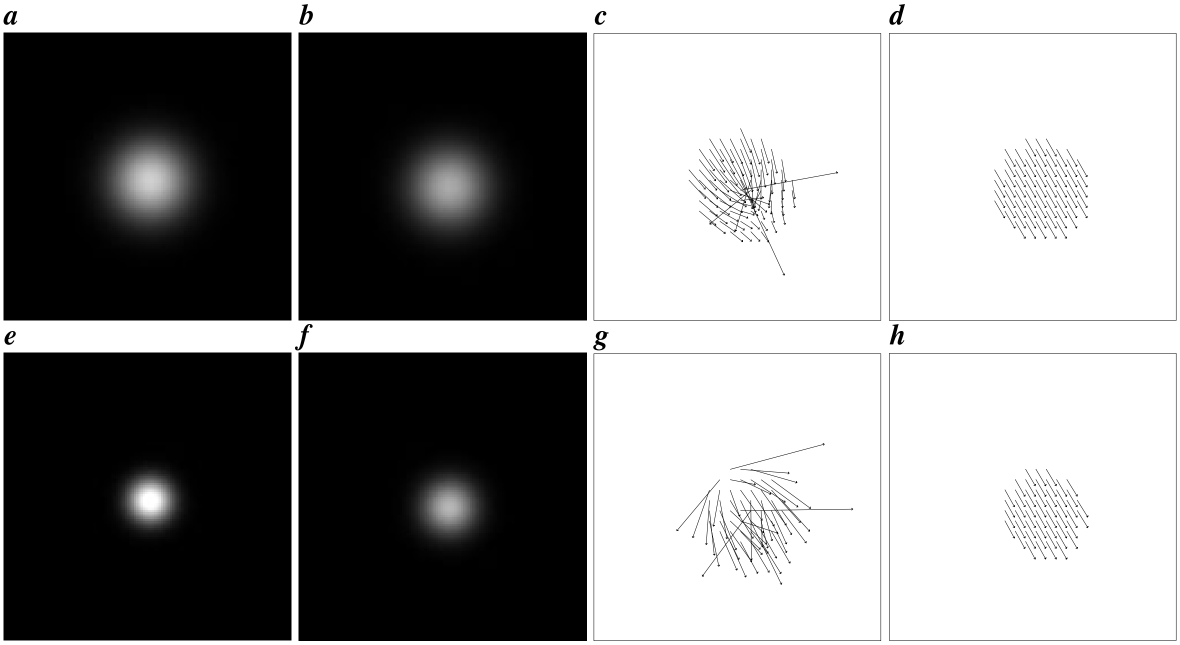


Figure 4.8: *Test sequence with a moving Gaussian bell and additional brightness changes. Exponentially decay:* ***a*** *first and* ***b*** *last frame,* ***c*** *standard optical flow and* ***d*** *extended flow. Diffusion:* ***e*** *first and* ***f*** *last frame,* ***g*** *standard optical flow and* ***h*** *extended flow.*

Relaxation processes. A relaxation also causes a first-order temporal change. This time the change is proportional to the relaxation constant κ and the gray value g:

$$f(g,t,\boldsymbol{a}) = -\kappa\, g \quad \rightarrow \quad \boldsymbol{d} = [g_x\, g_y\, g\, g_t]^T \quad \text{and} \quad \tilde{\boldsymbol{f}} = [u\, v\, \kappa\, 1]^T \ . \tag{4.56}$$

Examples for relaxation processes include decay processes of fluorescence or temperature.

Diffusion processes. If a transport process is captured by image sequences, we usually encounter diffusion as well as translation. According to Fick's second law, this is a second-order process with a diffusion coefficient D that describes the tendency of an object to spread out by this process:

$$f(g,t,\boldsymbol{a}) = D\Delta g \quad \rightarrow \quad \boldsymbol{d} = [g_x\, g_y\, \Delta g\, g_t]^T \quad \text{and} \quad \tilde{\boldsymbol{f}} = [u\, v\, -D\, 1]^T \ . \tag{4.57}$$

As can be seen in Fig. 4.8d,h we are able to estimate the correct velocity field even under such circumstances when the appropriate model is used. On top of that the physical parameters of interest such as decay constants are estimated simultaneously. If there is no such physical motivated model available one can of course simply take a Taylor series expansion up to the desired order. This has for example been shown to be useful for moving illumination envelopes [Haußecker and Fleet, 2000].

4.3 Range Flow

As range flow estimation is the main concern in this work we will treat this special case in slightly more detailed. A least squares solution to the range flow problem, including formulas for the normal flows, has been presented in [Barron and Spies, 2000; Barron et al., 2000] and is described in Appendix A.

Depending on the constraint equation used the data and parameter vectors for range flow estimation become [Spies et al., 1999]:

$$\boldsymbol{d} = [Z_X \, Z_Y \, 1 \, Z_t]^T \quad \text{and} \quad \tilde{\boldsymbol{f}} = [U \; V \; -W \; 1]^T \, , \tag{4.58}$$

when based on (3.19). As the results presented here are not obtained using this equation we will not further discuss this case. But note that here we have a constant term in the data vector. Thus some scaling is appropriate even when we assume that the other derivatives show the same noise level, see Sect. 4.1.4.

In the experiments reported we are using (3.24) as constraint equation to obtain:

$$\boldsymbol{d} = \left[\frac{\partial(Z,Y)}{\partial(x,y)} \, \frac{\partial(X,Z)}{\partial(x,y)} \, \frac{\partial(Y,X)}{\partial(x,y)} \, \frac{\partial(X,Y,Z)}{\partial(x,y,t)} \right]^T \quad \text{and} \quad \tilde{\boldsymbol{f}} = [U \, V \, W \, 1]^T \, . \tag{4.59}$$

However in order to establish suitable values for the equilibration matrix we look at (3.26) for simplicity. We assume the error in the X,Y,Z-coordinates to be given by $\sigma_x \approx \sigma_y \ll \sigma_z$, compare Sect. 2.1 and Sect. 7.3. As all derivatives are computed using the same filter kernels we can use the same scaling factor a to get the variance of the error in the derivatives, see (4.20). For the error variance of the data vector we then find:

$$\begin{aligned}
\sigma_{d_1}^2 &= (\partial_x Z)^2 a \sigma_y^2 + (\partial_y Y)^2 a \sigma_z^2 \approx (\partial_y Y)^2 a \sigma_z^2 \approx (\Delta y)^2 a \sigma_z^2 \, , && (4.60)\\
\sigma_{d_2}^2 &= (\partial_y Z)^2 a \sigma_x^2 + (\partial_x X)^2 a \sigma_z^2 \approx (\partial_x X)^2 a \sigma_z^2 \approx (\Delta x)^2 a \sigma_z^2 \, , && (4.61)\\
\sigma_{d_3}^2 &= (\partial_y Y)^2 a \sigma_x^2 + (\partial_x X)^2 a \sigma_y^2 \approx (\Delta y)^2 a \sigma_x^2 + (\Delta x)^2 a \sigma_y^2 \, , && (4.62)\\
\sigma_{d_4}^2 &\approx (\partial_x X)^2 (\partial_y Y)^2 a \sigma_z^2 \approx (\Delta x)^2 (\Delta y)^2 a \sigma_z^2 \, . && (4.63)
\end{aligned}$$

In order to obtain global scaling values we also use the approximation $\partial_x X \approx \Delta x$ and $\partial_y Y \approx \Delta y$. This assumptions states that the world coordinates of the sampled points are evenly sampled. For d_4 we further assumed $\partial_t X \approx \partial_t Y \approx 0$ stating that the X- and Y-coordinates seen at a particular pixel do only change by a minimal amount compared to Δx and Δy. While these assumptions are not entirely correct they are a reasonable for most scenes. As stated above each data element is then scaled by the inverse of its standard error deviation. Typical values are: $\sigma_x \approx \sigma_y \approx 0.01mm$, $\sigma_z \approx 0.1mm$ and $\Delta x \approx \Delta y \approx 0.2mm$.

4.3.1 Flow Types

As discussed in Sect. 3.3 we encounter two types of normal flows for range flow. On linear structures the motion along the line can not be resolved. This results in two vanishing eigenvalues of the structure tensor. We name the corresponding minimum norm velocity *line flow* which is computed from (4.31) as follows:

$$\boldsymbol{f}_l = \frac{1}{1 - e_{14}^2 - e_{24}^2} \left[e_{14} \begin{bmatrix} e_{11} \\ e_{12} \\ e_{13} \end{bmatrix} + e_{24} \begin{bmatrix} e_{21} \\ e_{22} \\ e_{23} \end{bmatrix} \right] \, . \tag{4.64}$$

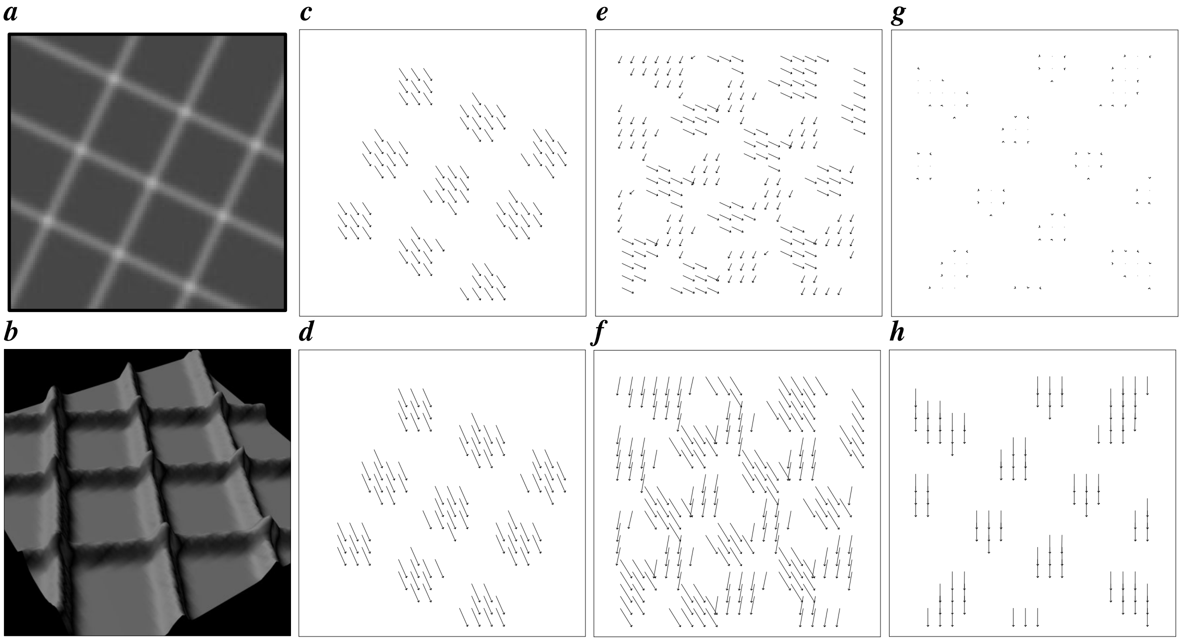


Figure 4.9: Example flow types for range flow: ***a*** *synthetic depth map,* ***b*** *rendered.* $X-Y$ *components of the estimated flow fields:* ***c*** *full flow,* ***e*** *line flow and* ***g*** *plane flow and* $X-Z$ *components of the estimated flow fields:* ***d*** *full flow,* ***f*** *line flow and* ***h*** *plane flow.*

When we encounter planar structures only the movement perpendicular to the plane can be resolved. Hence all but one eigenvalue falls below τ_2 and the *plane flow* becomes:

$$\boldsymbol{f}_p = \frac{e_{14}}{1-e_{14}^2}\begin{bmatrix} e_{11} \\ e_{12} \\ e_{13} \end{bmatrix} = \frac{e_{14}}{e_{11}^2+e_{12}^2+e_{13}^2}\begin{bmatrix} e_{11} \\ e_{12} \\ e_{13} \end{bmatrix} . \tag{4.65}$$

On point like structures motion in all directions can be measured and the resulting *full flow* is computed as usual from the eigenvector to the single vanishing eigenvalue λ_4:

$$\boldsymbol{f}_f = \frac{1}{e_{44}}\begin{bmatrix} e_{41} \\ e_{42} \\ e_{43} \end{bmatrix} . \tag{4.66}$$

Figure 4.9 shows a synthetic example that nicely illustrates all three kinds of encountered flow types.

4.3.2 Incorporating Intensity

As discussed in Sect. 3.4 the intensity yields another constraint on the X- and Y-components of the sought range flow. In order to incorporate this constraint we have to insert zero at the appropriate place, see Sect. 4.1.3. Then (3.35) translates into:

$$\boldsymbol{d} = \left[\frac{\partial(I,Y)}{\partial(x,y)}\,\frac{\partial(X,I)}{\partial(x,y)}\;0\;\frac{\partial(X,Y,I)}{\partial(x,y,t)}\right]^T \quad \text{and} \quad \tilde{\boldsymbol{f}} = [U\;V\;W\;1]^T \tag{4.67}$$

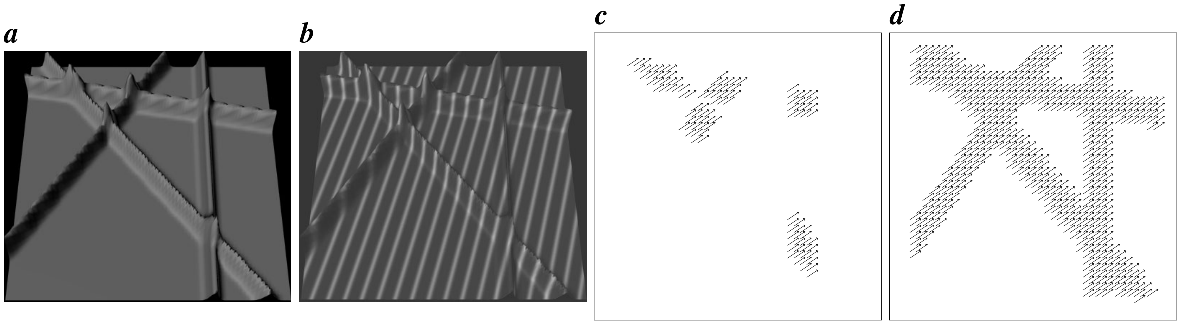


Figure 4.10: Rendered synthetic data: ***a*** *depth only and* ***b*** *with intensity texture. X-Y-range flow:* ***c*** *depth only and* ***d*** *using the intensity as well.*

As mentioned above we scale the intensity to have the same mean and variance as the depth data before combining the two channels. Upon replacing σ_z with σ_i we can use the same scaling strategy as before. The use of the intensity has two advantages. First more data points are used for the local estimation which increases the accuracy. Secondly and more importantly through the use of the intensity as well it is often possible to resolve the aperture problem. For example does the presence of a not-oriented texture on a planar surface enable the computation of full flow where otherwise only plane flow can be estimated.

To illustrate the effect of the additional intensity information we use the synthetic sequence shown in Fig. 4.10, the correct movement is $[0.66, -0.46, 0.34]^T$. Using depth maps alone local TLS estimation only gives sparse full flow, Fig. 4.10c. The flat surface only allows the computation of plane flow perpendicular to it. On the lines any movement in the direction of the lines can not be resolved. When the linear intensity pattern (Fig. 4.10b) is taken into consideration the displacements along the lines can now be computed. Hence the full flow density increases from 7% to 38%, see Fig. 4.10d. Clearly this is not the case if intensity and depth lines coincide. On the planar surface the intensity allows for the computation normal to the intensity lines, yet one direction remains ambiguous. Both the relative error and the directional error remain very low on this ideal test case ($< 1\%$ and $< 1°$).

Finally we want to show the effect of the intensity on the confidence and type measures introduced in Sect. 4.1.1 on a real castor oil leaf sequence in Fig. 4.11. When only the 3D structure is used the confidence is very high (Fig. 4.11e) but hardly any full velocity can be recovered (Fig. 4.11f). This is due to the relatively planar structure of the leaf yielding mainly plane flow (Fig. 4.11h). The veins are examples for linear structures and consequently the line flow measure (Fig. 4.11g) is high on the main veins. When the intensity data is used as well ($\beta = 1$) the situation changes dramatically. First we note that the confidence remains high almost everywhere (Fig. 4.11i). Yet the additional information makes a full flow estimate (Fig. 4.11j) on large portions of the leaf possible. Only at some locations where the intensity is quite uniform we detect line or even plane flow (Fig. 4.11k,l). Thus we find the incorporation of the intensity data to be vital for a complete velocity estimation on this kind of data.

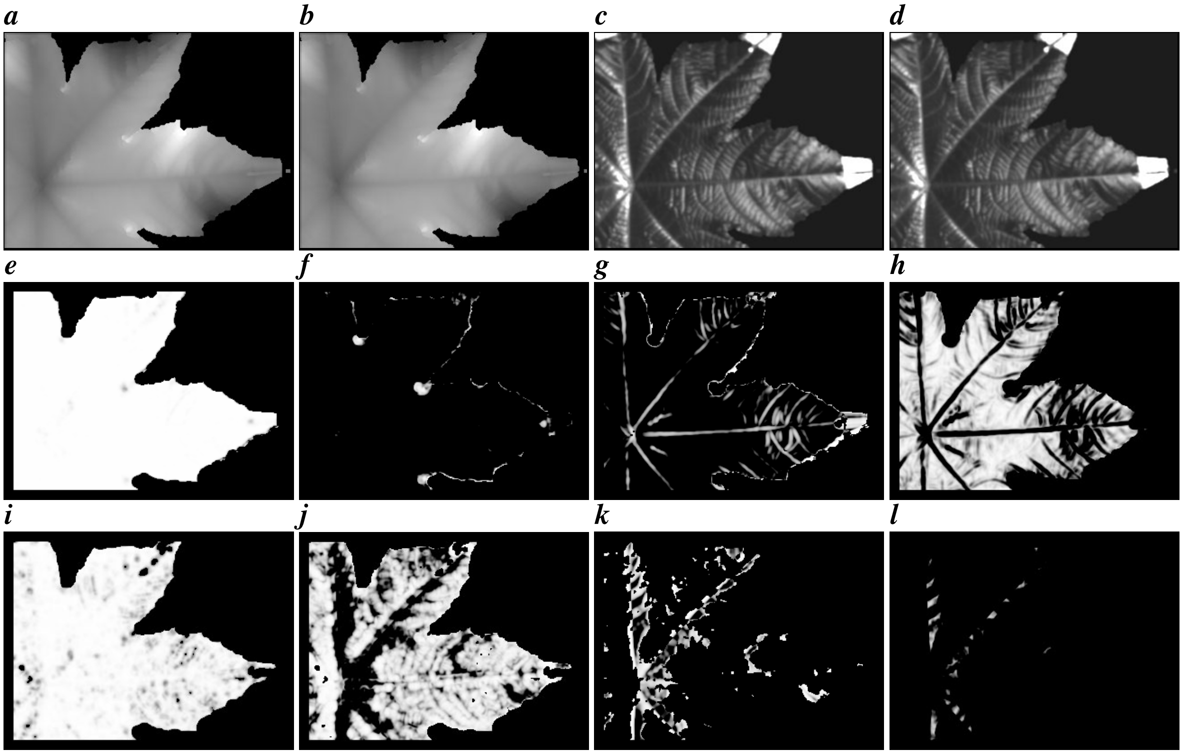


Figure 4.11: *Confidence and type measures on a real range sequence with and without intensity. Depth data* ***a*** *and* ***b*** *, Intensity data* ***c*** *and* ***d*** *. Using only the 3D data:* ***e*** *confidence,* ***f*** *full flow,* ***g*** *line flow and* ***h*** *plane flow type measures. Using both intensity and 3D data:* ***i*** *confidence,* ***j*** *full flow,* ***k*** *line flow and* ***l*** *plane flow type measures.*

4.4 Summary

This chapter introduced a general total least squares framework for the estimation of model parameters in a spatio-temporal data array. Optical and range flow have been shown to be special cases that can be nicely estimated within this framework. We introduced a normalised confidence measure and showed how linear dependencies in the data are resolved. Then the method has been extended to incorporate multiple data channels. Furthermore it has been shown how robust estimation can be embedded in this framework. Non-uniform noise levels in the data can be accounted for by the method of equilibration. On a few examples, including optical flow with brightness changes and range flow, the strength of the proposed algorithm has been demonstrated.

Chapter 5

TLS Parameter Regularisation

The TLS estimation method described in Chap. 4 usually produces sparse parameter vector fields. This chapter presents a general variational method to fill the remaining gaps and extend normal flows to full flows. The result is a dense and full parameter field. The same method can also be used for the restoration of corrupted images or image sequences. It is illustrated how the variational approach naturally extends to an anisotropic diffusion process. Towards this end we introduce a special choice of diffusion tensor that is motivated by the previous analysis of the structure tensor.

5.1 Variational Framework

It follows from basic variational calculus that for an energy functional over a Lagrange function L:

$$E(\boldsymbol{p}) = \int_{\Omega} L(\boldsymbol{p}, \nabla \boldsymbol{p}, \boldsymbol{d}, \nabla \boldsymbol{d}) \, \mathrm{d}x \mathrm{d}y \mathrm{d}t \; , \tag{5.1}$$

the solution vector $\boldsymbol{p}$ that minimises E has to fulfill the Euler-Lagrange equations:

$$\frac{\partial L}{\partial p_i} - \frac{\mathrm{d}}{\mathrm{d}x} \frac{\partial L}{\partial (p_i)_x} - \frac{\mathrm{d}}{\mathrm{d}y} \frac{\partial L}{\partial (p_i)_y} - \frac{\mathrm{d}}{\mathrm{d}t} \frac{\partial L}{\partial (p_i)_t} = 0 \; ; \quad i = 1 \ldots n \; . \tag{5.2}$$

Here the indices x, y, t denote partial differentiation with respect to x, y, t. Apart from the sought parameters L can also depend on a set of data values $\boldsymbol{d}$ that may include the coordinates or functions of them. Equation (5.2) gives n partial differential equations in the n parameters to be estimated. For every pixel such a system of equations exists and in special cases we can solve the resulting sparse matrix equation $\boldsymbol{A}\boldsymbol{p} = \boldsymbol{b}$ by the use of standard iterations $\boldsymbol{M}\boldsymbol{p}^{k+1} = \boldsymbol{N}\boldsymbol{p}^{k} + \boldsymbol{b}$ where $\boldsymbol{A} = \boldsymbol{M} - \boldsymbol{N}$ is a suitable splitting of the matrix $\boldsymbol{A}$ [Golub and van Loan, 1996]. Another numerical approach considers these equations as the steady state of a diffusion reaction system [Schnörr and Weickert, 2000]:

$$\frac{\partial p_i}{\partial s} = \frac{\partial L}{\partial p_i} - \frac{\mathrm{d}}{\mathrm{d}x} \frac{\partial L}{\partial (p_i)_x} - \frac{\mathrm{d}}{\mathrm{d}y} \frac{\partial L}{\partial (p_i)_y} - \frac{\mathrm{d}}{\mathrm{d}t} \frac{\partial L}{\partial (p_i)_t} \; ; \quad i = 1 \ldots n \; . \tag{5.3}$$

Where s denotes diffusion time or scale parameter, discretisation of $\partial_s p_i$ by finite differences results in the standard explicit Euler forward evaluation scheme.

The function L defines the problem at hand. For example in mechanics this Lagrange function is given in its natural form as $L = T - U$ with a kinetic Energy T and a potential energy U [Scheck, 1990]. In computer vision we often express the Lagrange function in terms of a data (D) and smoothness (V) term: $L = D + \alpha V$. The constant α is a regularisation parameter that controls the relative influence of the two terms. There exists a unique minimiser $\boldsymbol{p}$ for $E(\boldsymbol{p})$ when E is a convex functional [Schnörr, 1999]. Assume the energy functional can be written as a quadratic form:

$$E(\boldsymbol{p}) = \frac{1}{2}a(\boldsymbol{p},\boldsymbol{p}) - f(\boldsymbol{p}) + c \, , \tag{5.4}$$

with a bilinear form $a(\cdot,\cdot) : \mathrm{I\!R}^m \times \mathrm{I\!R}^m \to \mathrm{I\!R}$, a linear form $f(\cdot) : \mathrm{I\!R}^m \to \mathrm{I\!R}$ and a constant term c. Then $E(\boldsymbol{p})$ is strictly convex and a unique solution exits if [Hackbusch, 1986; Schnörr, 1991]:

$$a(\boldsymbol{p},\boldsymbol{p}) \geq C|\boldsymbol{p}| \quad \forall \boldsymbol{p} \in \mathrm{I\!R}^m \, , \tag{5.5}$$

with a positive constant $C > 0$. All the smoothness terms considered here are positive definite bilinear forms in the derivatives of $\boldsymbol{d}$. Hence it is sufficient to show that (5.5) holds for the data term alone in order to prove the existence of an unique solution.

5.1.1 Data Constraints

We discuss some possible data terms. After stating the standard versions we introduce a novel constraint that utilises the nature of the previously determined TLS solution. In the following we want to estimate the parameter field $\boldsymbol{p} = [p_1, \ldots, p_{n-1}]^T$, please note that this has previously been called $\boldsymbol{m}$. For numerical reasons the parameters should all lie in roughly the same range, here it is assumed that a normalisation towards this end has already been performed. In places where the parameters have been estimated from the above TLS procedure these estimations are denoted $\boldsymbol{q}$, obtained from (4.31) or directly from (4.4) where appropriate.

Parameter Constraint Equation.

Obviously one can use the original constraint equation (4.5) for the parameter estimation problem:

$$D = (\boldsymbol{d}^T \tilde{\boldsymbol{f}})^2 \, . \tag{5.6}$$

Note that here we will always use the L^2 norm but this is of course not the only possible choice. Such a data constraint is what is typically chosen in variational optical flow algorithms [Horn and Schunk, 1981; Nagel and Enkelmann, 1986; Schnörr, 1993; Alvarez et al., 2000b]. A simple extension of such optical flow algorithms to the parameter estimation framework considered in this work is given in Appendix B.

However, while some of these approaches make use of elaborate smoothness constraints the data constraint remains the same. We are concerned with the computation of dense full parameter fields from the previous TLS solution, Chap. 4. It then makes sense to exploit the information about the local data structure obtained from the structure tensor.

Data Restoration.

Section 2.2.2 already introduced a simple variational approach that can be used to interpolate missing data points and smooth the data. In the context discussed here this corresponds to a data term that directly penalises the deviation between a regularised solution $\boldsymbol{p}$ and the original solution $\boldsymbol{q}$ where available:

$$D = c(\boldsymbol{p} - \boldsymbol{q})^2 \; . \tag{5.7}$$

Here c gives the confidence of the original solution, for example the TLS confidence measure ω_c (4.16). Such a data term is for instance useful when $\boldsymbol{q}$ is simply a scalar data value and a solution is required that is close but also fulfills some smoothness properties. As seen in Sect. 2.2.2, missing data can also be interpolated using this data term. For this data term the bilinear form in (5.4) is given by:

$$a(\boldsymbol{p}, \boldsymbol{p}) = 2 \int c \, \boldsymbol{p}^T \boldsymbol{p} \; . \tag{5.8}$$

Hence we conclude with (5.5) that a unique solution exists if c does not vanish everywhere, i.e. some original data $\boldsymbol{q}$ exists. However when we want to apply such a data term in our parameter estimation framework we can only use the TLS solutions $\boldsymbol{q}$ where a full parameter estimation was possible. As this would imply to discard the information given by the minimal norm solutions such a data constraint is not the best choice in this case. We will now discuss how all information available from the TLS estimate can be incorporated in our regularisation framework.

TLS Subspace Constraint.

As we are now working with $m = n - 1$ dimensions and from the structure of the TLS solution given by (4.31) we can use the reduced eigenvectors as a, not necessarily orthogonal, basis for the desired solution:

$$\hat{\boldsymbol{b}}_i = \frac{1}{\sum_{k=1}^{m} e_{ik}^2} \begin{bmatrix} e_{i1} \\ \vdots \\ e_{im} \end{bmatrix} \; . \tag{5.9}$$

Using this notation we define a projection matrix which projects onto the subspace that was determined by the TLS algorithm. The basis of this subspace are the reduced eigenvectors to the non-vanishing eigenvalues:

$$\begin{aligned} \boldsymbol{P} &= \boldsymbol{B}_p \boldsymbol{B}_p^T \quad \text{where} \quad \lambda_1 \geq \ldots \geq \lambda_p > \lambda_{p+1} \approx \ldots \approx \lambda_n \approx 0 \; , & (5.10) \\ \boldsymbol{B}_p &= [\hat{\boldsymbol{b}}_1 \; \ldots \; \hat{\boldsymbol{b}}_p] \; . & (5.11) \end{aligned}$$

Each estimated parameter vector $\boldsymbol{q}$ restricts the solution within this subspace. We therefore require the regularised solution to be close in a least squares sense:

$$D = \omega_c \left(\boldsymbol{P}\boldsymbol{p} - \boldsymbol{q} \right)^2 \; . \tag{5.12}$$

At locations where no solution has been computed obviously no such data term exists, this is readily taken into account by using the TLS confidence measure ω_c (4.16) as weight. It is sometimes desirable to use an orthogonal basis as then the variation of $\boldsymbol{p}$ is restricted to the orthogonal complement of the TLS subspace. Such a basis system is easily computed by successive orthogonalisation [Bronstein and Semendjajew, 1989]:

$$\hat{\boldsymbol{a}}_1 = \hat{\boldsymbol{b}}_1 \quad \text{and} \quad \hat{\boldsymbol{a}}_k = \frac{\boldsymbol{a}_k}{|\boldsymbol{a}_k|} \quad \text{with} \quad \boldsymbol{a}_k = \hat{\boldsymbol{b}}_k - \sum_{i=1}^{k-1} \frac{\hat{\boldsymbol{b}}_k^T \hat{\boldsymbol{a}}_i}{\hat{\boldsymbol{a}}_i^T \hat{\boldsymbol{a}}_i} \hat{\boldsymbol{a}}_i \,. \tag{5.13}$$

A conceptually similar data term using the orthogonal projection on a subspace of an OLS solution based on multiple constraints at the same pixel has been given by Schnörr [1993]. The considered data term (5.12) can be written in the form of (5.4) with:

$$a(\boldsymbol{p}, \boldsymbol{p}) = 2 \int \omega_c \, \boldsymbol{p}^T \boldsymbol{P}^T \boldsymbol{P} \boldsymbol{p} \,, \quad f(\boldsymbol{p}) = 2 \int \omega_c \, \boldsymbol{q}^T \boldsymbol{P} \boldsymbol{p} \quad \text{and} \quad c = \int \omega_c \, \boldsymbol{q}^T \boldsymbol{q} \,. \tag{5.14}$$

Let's assume that we only have one full solution at one place x_0 and no solution everywhere else. At this location we then have $\boldsymbol{P} = \mathbb{1}$ and the bilinear form becomes: $a(\boldsymbol{p}, \boldsymbol{p}) = 2\omega_{c0} \, \boldsymbol{p}^T \boldsymbol{p}$. As a full flow solution implies $\omega_{c0} > 0$ the condition (5.5) is satisfied with $C = \frac{1}{2\omega_{c0}}$. Because $a(\boldsymbol{p}, \boldsymbol{p})$ is positive definite the argument still holds when there are further places where a TLS solution $\boldsymbol{q}$ is available. Thus we conclude that the existence of a single full flow estimate is sufficient for a unique regularised solution to exist. Of course a dense solution based on just a single full flow estimate does in most cases not capture the real parameter field.

5.1.2 Smoothness Constraints

There is a large variety of smoothness constraints or regularisers that have been proposed for optical flow estimation. In a recent summary Schnörr and Weickert [2000] showed the connection between such variational approaches and a corresponding diffusion process. In the context of tensor-driven diffusion it is worth mentioning that Alvarez et al. [1999] showed that the oriented smoothness approach [Nagel and Enkelmann, 1986] can be interpreted as a tensor-driven anisotropic diffusion process. Here we will explicitly state three different types of smoothness constraints.

The Membrane Model.

In their pioneering work Horn and Schunk [1981] introduced a homogenous regulariser for optical flow computation. Translated into the m parameter estimation framework this results in the following smoothness term:

$$V = \sum_{i=1}^{m} (\nabla p_i)^2 \,. \tag{5.15}$$

Apart from this membrane model higher order smoothness terms such as a thin-plate could be employed as well [Terzopoulos, 1986]. 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 ($\nabla = [\partial x, \partial y, \partial t]^T$).

Clearly such a regulariser produces undesirable results at discontinuities which are smoothed across. However if the considered application produces smooth and continuous parameters a membrane model is admissible. This is the case for range flow estimation on leaves, compare also Sect. 2.2.2. Therefore we will discuss this case in more detail in Sect. 5.1.3.

Projected Smoothness.

In the above formulation we obtain a smooth parameter field whose overall smoothness is regulated by the smoothness parameter α. While a large smoothness parameter guarantees fast convergence we would rather use a slightly smaller value in order to give more influence to the data term. A practical approach might be to first run the algorithm for a while with a large α to close any gaps. Afterwards one can then obtain the final result through additional iterations with a smaller value. The same could be obtained in a more automated manner by reducing α after each time step until it reaches the desired small value. This obviously implies the use of a fixed minimal number of iterations.

As we have already determined the subspace in which the TLS solution lies it makes sense to smooth this subspace differently from its orthogonal complement. While we want strong smoothing in the orthogonal space in order to spread the missing information we only require a little bit of smoothing of the already available solution for stabilisation. These ideas motivate the following smoothness term:

$$V = \psi \sum_{i=1}^{m} (\nabla(\boldsymbol{P}\boldsymbol{p})_i)^2 + (1-\psi) \sum_{i=1}^{m} (\nabla(\boldsymbol{P}^{\perp}\boldsymbol{p})_i)^2 \, . \tag{5.16}$$

Where the orthogonal projection matrix is given by $\boldsymbol{P}^{\perp} = \mathbb{1} - \boldsymbol{P}$. The relative smoothness between the two orthogonal subspaces is regulated by the parameter ψ. Because we require an orthogonal projection in this case we use the orthogonal and normalised basis$\hat{\boldsymbol{a}}_i$ (5.13) to construct the projection matrix.

Note that it makes no sense to use such a smoothness constraint when we use the TLS subspace data constraint. Because in this case there is no data constraint in the orthogonal subspace any smoothing parameter cancels out. Hence the membrane model does in this combination already ensure a different smoothing of the two subspaces which is readily modelled by α, see Sect. 5.1.3.

Anisotropic Diffusion.

So far we only considered linear isotropic smoothness terms. This can be made nonlinear by varying the smoothness strength locally. For example we can replace the global constants α and ψ by a function that depends on the certainty ω_t (4.21) that a certain normal flow type has been estimated. This is directly related to a nonlinear image diffusion approach where the smoothing strength is based on the gradient magnitude, which has originally been termed somewhat misleadingly anisotropic diffusion [Perona and Malik, 1990]. We are not exploiting this any further but rather investigate a truly anisotropic approach.

We do not only want to vary the smoothing strength locally but also smooth differently in different directions (on the data array). To do so we first have to automatically establish these directions and the associated smoothing strength, if they are not available a priori. Towards this end we examine the standard structure tensor $\boldsymbol{J}$ (4.11) for spatio-temporal data sets in greater detail. As shown in Chap. 4, an eigenvalue analysis of the structure tensor corresponds to a total least squares fit of a locally constant displacement vector (= orientation) field to the data. There are four types of neighbourhoods that have to be considered, compare Sect. 4.1.1:

1. **No structure** $(\lambda_{1,2,3} \approx 0)$: All eigenvalues vanish and no motion can be estimated.
2. **1D-structure** $(\lambda_3 \approx 0;\ \lambda_1, \lambda_2 > \tau_2)$: Moving corner- or point-like structures result in one-dimensional trajectories in direction of $\hat{\boldsymbol{e}}_3$.
3. **2D-structure** $(\lambda_{2,3} \approx 0;\ \lambda_1 > \tau_2)$: This is the well known aperture problem caused by a moving line resulting in a plane in the data volume. The normal of that plane is given by $\hat{\boldsymbol{e}}_1$.
4. **No coherent structure** $(\lambda_{1,2,3} > \tau_2)$: In this case the local fit of the flow model (here constant) failed and no sensible orientation may be estimated.

Hence we conclude that the eigenvectors of the structure tensor are a good choice for the sought smoothing directions. For the smoothing strength along these directions we use a normalised measure based on the eigenvalues which is similar to ω_t:

$$\mu_i := \begin{cases} 1 & \text{if } \mu_i \leq \tau_2, \\ 1 - \exp(\frac{-c}{(\lambda_i - \tau_2)^2}) & \text{else,} \end{cases} \qquad i = 1, 2, 3\ , \tag{5.17}$$

where $c > 0$. Because the λ_i are bounded the unstable case of a zero smoothing strength does not occur. There is nothing magic about the choice of an exponential function other than its previous successful use for nonlinear diffusion [Perona and Malik, 1990; Weickert, 1998]. The above choice of eigenvalues explains itself through the four previously considered cases:

1. **No structure**: All μ_i are one and isotropic smoothing is employed.
2. **1D-structure**: Only λ_3 falls below the threshold. Thus the only smoothing is along $\hat{\boldsymbol{e}}_3$ which points in the direction of the displacement.
3. **2D-structures**: Both λ_2 and λ_3 are below τ_2, this leads to smoothing in both directions of the associated eigenvectors $\hat{\boldsymbol{e}}_2$ and $\hat{\boldsymbol{e}}_3$. As we are dealing with a moving line this corresponds to smoothing both along the line and in the direction of the movement.
4. **No coherent structure**: All eigenvalues are big ($> \tau_2$) and no or very little smoothing is applied. In particular there is no smoothing across motion discontinuities.

It is clear from (5.17) that the above distinctions are not hard breaks but are continuously approached as the eigenvalues reach the threshold. From the previous considerations we can define a nonlinear anisotropic regulariser as follows:

$$V = \sum_{i=1}^{3} \mu_i \sum_{k=1}^{m} (\hat{\boldsymbol{e}}_i^T \nabla p_k)^2\ . \tag{5.18}$$

This smoothness term restricts the variation of the parameter vector along the spatio-temporal directions given by the $\hat{\mathbf{e}}_i$ according to the μ_i. Because the described regulariser enhances the oriented structures present in the data the related diffusion process, has been termed *orientation enhancing diffusion* [Spies and Scharr, 2001]. Additionally we could define such a smoothness term for both of the TLS subspaces similar to (5.16).

5.1.3 Example variational schemes

We now discuss example data and smoothness term combinations and describe a possible numerical realisation. The key application considered here, namely range flow estimation on moving leaf surfaces, guarantees smooth and continuous velocity fields. Therefore we do not need to consider an anisotropic regulariser in this special case. As an example showing the benefit of such a regulariser we consider the use of anisotropic diffusion to enable optical flow estimation in noisy image data.

Subspace Data and Membrane Model.

As this is the approach we are using for range flow estimation we will discuss this combination in more detail. Combining the data (5.12) and smoothness (5.15) constraints in the considered area A yields the following minimisation problem (we set $\omega = \omega_c$ in the following):

$$\int_A \underbrace{\left\{ \omega \left(\boldsymbol{P}\boldsymbol{p} - \boldsymbol{q} \right)^2 + \alpha \sum_{i=1}^{m} (\nabla p_i)^2 \right\}}_{L(\boldsymbol{p}, \nabla \boldsymbol{p})} \mathrm{d}x \mathrm{d}y \to \min . \tag{5.19}$$

Here α is a regularisation parameter that controls the influence of the smoothness term. Using vector notation we write the Euler-Lagrange equations as follows:

$$\frac{\partial L}{\partial \boldsymbol{p}} - \frac{\mathrm{d}}{\mathrm{d}x} \frac{\partial L}{\partial (\boldsymbol{p}_x)} - \frac{\mathrm{d}}{\mathrm{d}y} \frac{\partial L}{\partial (\boldsymbol{p}_y)} = 0 . \tag{5.20}$$

Computing the derivatives yields:

$$2\omega \boldsymbol{P}(\boldsymbol{P}\boldsymbol{p} - \boldsymbol{q}) - 2\alpha \left[\frac{\mathrm{d}}{\mathrm{d}x}(\boldsymbol{p}_x) + \frac{\mathrm{d}}{\mathrm{d}y}(\boldsymbol{p}_y) \right] = 0 . \tag{5.21}$$

Introducing the Laplacian $\Delta \boldsymbol{p} = \boldsymbol{p}_{xx} + \boldsymbol{p}_{yy}$ this becomes:

$$\omega \boldsymbol{P}\boldsymbol{p} - \omega \boldsymbol{P}\boldsymbol{q} - \alpha \Delta \boldsymbol{p} = 0 . \tag{5.22}$$

Where the idempotence of the projection matrix $\boldsymbol{P}\boldsymbol{P} = \boldsymbol{P}$ is used. The Laplacian can be approximated as $\Delta \boldsymbol{p} = \bar{\boldsymbol{p}} - \boldsymbol{p}$, where $\bar{\boldsymbol{p}}$ denotes a local average [Jähne, 1995]. Using this approximation we arrive at:

$$\left(\omega \boldsymbol{P} + \alpha \mathbb{1} \right) \boldsymbol{p} = \alpha \bar{\boldsymbol{p}} + \omega \boldsymbol{P}\boldsymbol{q} . \tag{5.23}$$

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

$$\boldsymbol{p}^{k+1} = \alpha \boldsymbol{A}^{-1} \bar{\boldsymbol{p}}^k + \omega \boldsymbol{A}^{-1} \boldsymbol{P} \boldsymbol{q} \, . \tag{5.24}$$

Initialisation can for example be done by using the TLS solution $\boldsymbol{p}^0 = \boldsymbol{q}$ where available. Note that (5.24) converges ($\boldsymbol{p}^{k+1} = \boldsymbol{p}^k$) when (5.23) is satisfied, i.e. when (5.19) is minimised. The matrix $\boldsymbol{A}^{-1}$ only has to be computed once, existence of the inverse is guaranteed as shown next.

> To show that $\boldsymbol{A}$ is always regular we use the Sherman-Morrison-Woodbury Lemma [Großmann and Terno, 1997]. 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:
>
> $$\bar{\boldsymbol{Q}} = \boldsymbol{Q} + \boldsymbol{R}\boldsymbol{S}\boldsymbol{T}^T \, , \tag{5.25}$$
>
> 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:
>
> $$\boldsymbol{A} = \alpha \mathbb{1}_m + \omega \boldsymbol{B}_p \mathbb{1}_p \boldsymbol{B}_p^T \, . \tag{5.26}$$
>
> Thus we have to examine:
>
> $$\boldsymbol{U} = \frac{1}{\omega} \mathbb{1}_p + \bar{\boldsymbol{B}}_p^T (\frac{1}{\alpha} \mathbb{1}_m) \bar{\boldsymbol{B}}_p \, . \tag{5.27}$$
>
> $\boldsymbol{U}$ has diagonal elements $\alpha^{-1} + \omega^{-1}$ and the off-diagonal elements of $\boldsymbol{U}$ are given by $\alpha^{-1} \boldsymbol{b}_i^T \boldsymbol{b}_j = \alpha^{-1} \cos(\beta_{ij}) \leq \alpha^{-1}$, where β_{ij} is the angle between $\boldsymbol{b}_i$ and $\boldsymbol{b}_j$. Thus $\boldsymbol{U}$ is diagonal dominant with all diagonal elements strictly positive and hence a symmetric positive definite matrix [Golub and van Loan, 1996]. This implies $\boldsymbol{U}$ is regular. Hence we conclude that for $\alpha > 0$ the matrix $\boldsymbol{A}$ can always be inverted. When an orthogonal projection is used we can even give the inverse analytically. From
>
> $$\bar{\boldsymbol{Q}}^{-1} = \boldsymbol{Q}^{-1} - \boldsymbol{Q}^{-1} \boldsymbol{R} \boldsymbol{U}^{-1} \boldsymbol{T}^T \boldsymbol{Q}^{-1} \, , \tag{5.28}$$
>
> and $\boldsymbol{U} = (\omega^{-1} + \alpha^{-1}) \mathbb{1}_p = \frac{\alpha + \omega}{\alpha \omega} \mathbb{1}_p$ we find:
>
> $$\boldsymbol{A}^{-1} = \alpha^{-1} \left(\mathbb{1}_m - \frac{\omega}{\alpha + \omega} \boldsymbol{P} \right) \, . \tag{5.29}$$
>
> In order to verified that $\boldsymbol{A}^{-1} \boldsymbol{A} = \mathbb{1}$ only the idempotence of the projection matrix is needed, hence (5.29) also holds when the basis vectors (5.9) are not orthogonalised.

We can thus examine the iteration process in more detail. Inserting (5.29) into (5.24) yields:

$$\boldsymbol{p}^{k+1} = \bar{\boldsymbol{p}}^k - \frac{\omega}{\alpha + \omega} \boldsymbol{P} \bar{\boldsymbol{p}}^k + \frac{\omega}{\alpha + \omega} \boldsymbol{P} \boldsymbol{q} \, . \tag{5.30}$$

where we also used the idempotence of $\boldsymbol{P}$. Using $\mathbb{1} = \boldsymbol{P} + \boldsymbol{P}^{\perp}$ we can separate the two subspaces:

$$\boldsymbol{p}^{k+1} = \boldsymbol{P}^{\perp} \bar{\boldsymbol{p}}^k + \frac{1}{\alpha + \omega} \boldsymbol{P} (\alpha \bar{\boldsymbol{p}}^k + \omega \boldsymbol{q}) \, . \tag{5.31}$$

In the orthogonal subspace, where no initial TLS solution is available, the local average is used. In the subspace determined by the TLS solution the iterative update is given by a weighted mean of the

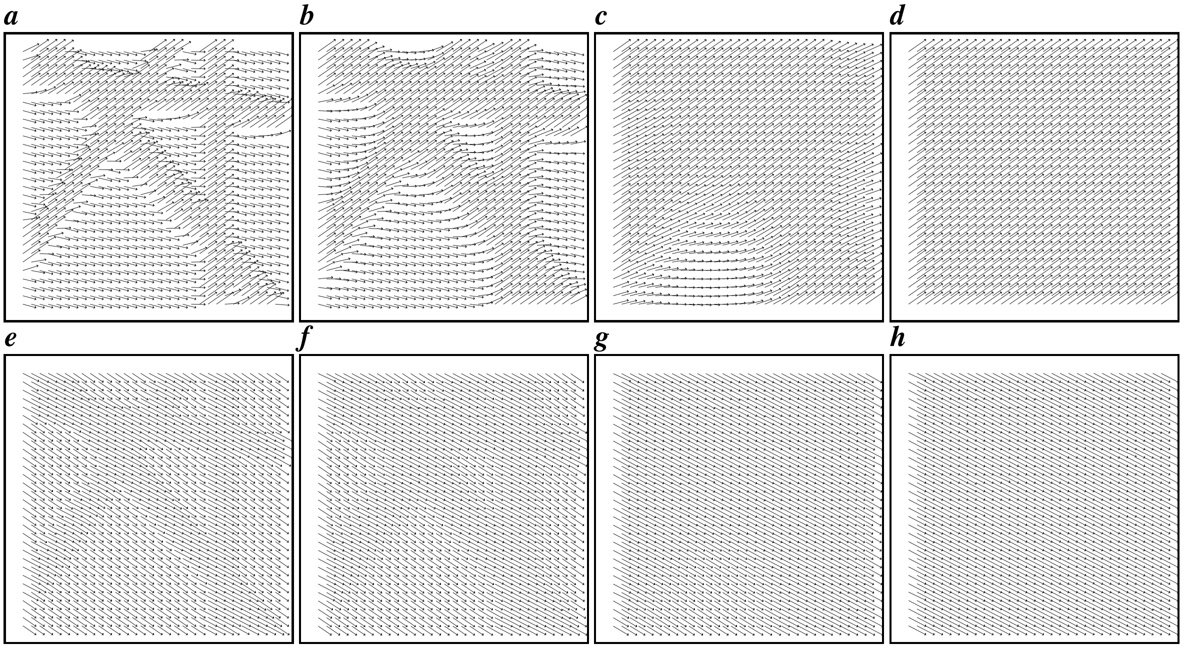


Figure 5.1: *Example regularisation for the range flow field computed on the data given in Fig. 4.10. XY-flow filed after **a** 10, **b** 100, **c** 1000 and **d** 10000 iterations. XZ-flow filed after **e** 10, **f** 100, **g** 1000 and **h** 10000 iterations.*

local average and the originally available solution. The weights are determined by the regularisation constant α and the confidence measure ω respectively. Notice that the choice of α only regulates the amount of smoothing in the TLS subspace and has no effect in the orthogonal subspace.

The effect of this regularisation for a range flow field is illustrated in Fig. 5.1 where the algorithm is applied to the data given in Fig. 4.10. Here we used a regularisation parameter $\alpha = 1$. In the beginning the flow field is dominated by the normal flows. With increasing iterations the full flow information spreads out. However due to the large regions without initial full flow estimation it takes a very long time until convergence is reached. In this example a 5×5 Box filter is used to compute the average.

Anisotropic Diffusion.

We first discuss the case of the anisotropic diffusion smoother without any data term. We will see that the Euler-Lagrange equations lead to a diffusion system driven by a diffusion tensor. Then the data is evolved, as seen below, with the diffusion time. The complete evolution is called *scale space* [Lindeberg, 1994; Weickert, 1998]. If there is no data term in this case the result has to be examined at a certain scale. Otherwise the diffusion terminates at the steady state of the smoothness term which is simply a homogeneous data set. We can rewrite the anisotropic smoothness term (5.18) as:

$$V = \alpha \sum_{i=1}^{3} \mu_i \sum_{k=1}^{m} (\hat{\boldsymbol{e}}_i^T \nabla p_k)^2 = \alpha \sum_{k=1}^{m} (\nabla p_k)^T \left(\sum_{i=1}^{3} \mu_i \hat{\boldsymbol{e}}_i \hat{\boldsymbol{e}}_i^T \right) \nabla p_k = \alpha \sum_{k=1}^{m} (\nabla p_k)^T \boldsymbol{D} \, \nabla p_k \; . \quad (5.32)$$

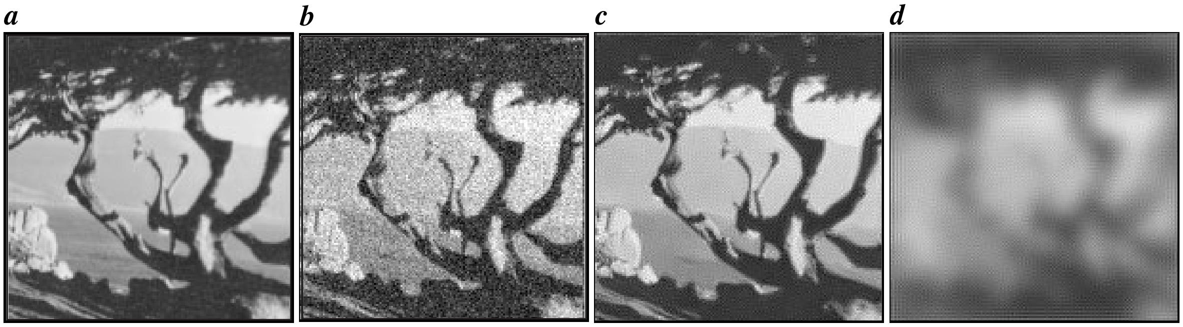


Figure 5.2: Denoising the diverging tree sequences, central frame: ***a*** *without noise,* ***b*** $\sigma_n = 20$*,* ***c*** *after 20 anisotropic diffusion iterations and* ***d*** *after 20 isotropic diffusion iterations.*

Where we introduced the diffusion tensor:

$$\boldsymbol{D} = \sum_{i=1}^{3} \mu_i \hat{\boldsymbol{e}}_i \hat{\boldsymbol{e}}_i^T = [\hat{\boldsymbol{e}}_1\ \hat{\boldsymbol{e}}_2\ \hat{\boldsymbol{e}}_3] \begin{bmatrix} \mu_1 & & \\ & \mu_2 & \\ & & \mu_3 \end{bmatrix} [\hat{\boldsymbol{e}}_1\ \hat{\boldsymbol{e}}_2\ \hat{\boldsymbol{e}}_3]^T \ . \tag{5.33}$$

The μ_i are the eigenvalues of $\boldsymbol{D}$ which is diagonal in the coordinate system given by the smoothing directions $\hat{\boldsymbol{e}}_i$. As the μ_i are strictly positive $\boldsymbol{D}$ is a symmetric positive definite matrix. There is one Euler-Lagrange equation (5.2) for every k, because of the distinct sum in (5.32) we can consider them separately. We drop the index k at the moment to keep the notation simple. Then we encounter Euler-Lagrange equations:

$$\begin{bmatrix} \frac{\mathrm{d}}{\mathrm{d}x} & \frac{\mathrm{d}}{\mathrm{d}y} & \frac{\mathrm{d}}{\mathrm{d}z} \end{bmatrix} \begin{bmatrix} \frac{\partial V}{\partial p_x} \\ \frac{\partial V}{\partial p_y} \\ \frac{\partial V}{\partial p_z} \end{bmatrix} = 2\,\nabla \cdot \begin{bmatrix} \boldsymbol{d}_1^T \nabla p \\ \boldsymbol{d}_2^T \nabla p \\ \boldsymbol{d}_3^T \nabla p \end{bmatrix} = 2\,\nabla \cdot (\boldsymbol{D}\,\nabla p) = 0 \ . \tag{5.34}$$

If there is no data term present we can drop the factor 2 and using (5.3) we end up with the diffusion equations:

$$\frac{\partial p_k}{\partial s} = \nabla \cdot (\boldsymbol{D}\,\nabla p_k)\ ; \quad k = 1 \ldots m \ . \tag{5.35}$$

Note that all elements of $\boldsymbol{p}$ are smoothed using the same diffusion tensor. The use of this anisotropic diffusion process to denoise an image sequence is illustrated in Fig. 5.2. In this simple case there is only one component $m = 1$ and hence only one diffusion system. We see that after applying 20 iterations the denoised data (Fig. 5.2 c) is very close to the original noise free data (Fig. 5.2 a). However when an isotropic diffusion is chosen, i.e. $\mu_i = 1;\ i = 1, 2, 3$, the same number of iterations not only removes the noise but also strongly smoothes the data (Fig. 5.2 d). As this destroys the image structure it is often undesirable, for example for an optical flow estimation, see Sect. 5.2.3.

In the presence of a data term the Euler-Lagrange equations result in a diffusion reaction system. For example when using the data restoration term (5.7) the corresponding system becomes:

$$\frac{\partial p_k}{\partial s} = \nabla \cdot (\boldsymbol{D}\,\nabla p_k) - \frac{c}{\alpha}(p_k - q_k)\ ; \quad k = 1 \ldots m \ . \tag{5.36}$$

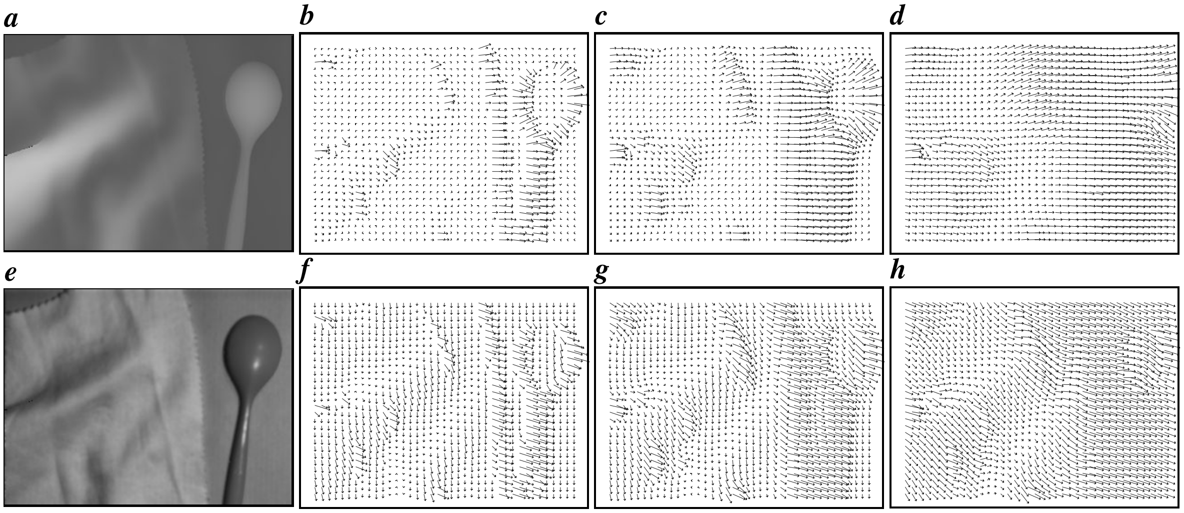


Figure 5.3: Example regularised range flow: ***a*** *depth data and XY-components of the range flow field after* ***b*** *0,* ***c*** *100 and* ***d*** *1000 iterations.* ***e*** *depth data and XZ-components of the range flow field after* ***f*** *0,* ***g*** *100 and* ***h*** *1000 iterations.*

Note that here it is convenient to change the sign in the Euler Lagrange equations (5.2). With the TLS subspace data constraint the resulting diffusion system is given by:

$$\frac{\partial p_k}{\partial s} = \nabla \cdot (\boldsymbol{D} \, \nabla p_k) - \frac{\omega}{\alpha} \boldsymbol{P}(p_k - q_k) \; ; \quad k = 1 \ldots m \; . \tag{5.37}$$

A highly accurate and efficient discretisation for this equation has recently been derived for both 2D [Scharr, 2000; Scharr and Weickert, 2000] and 3D data [Spies and Scharr, 2001]. For completeness this scheme is given in Appendix C.

5.2 Examples

In order to illustrate the strength of the presented method to compute dense range and optical flow fields from a previous sparse TLS estimation we show some results on real data. We also illustrate how the anisotropic diffusion system can be used to enable accurate optical flow estimation in very noisy image sequences.

5.2.1 Range Flow

As stated above, dense range flow fields may be obtained using the TLS subspace data constraint and the membrane regulariser. A first real data range flow example that shows how the information spreads out is given in Fig. 5.3 where we used $\alpha = 1$. Because here only the depth data without the intensity is used, there are very few places where a full flow estimate is possible. Therefore the initial flow (Fig. 5.3 b,f) clearly shows the linear and planar regions. With continuing regularisation this information spreads out but the initial flow field components are preserved. Because of the very sparse full flow estimates the final result is not very trustworthy in this special case.

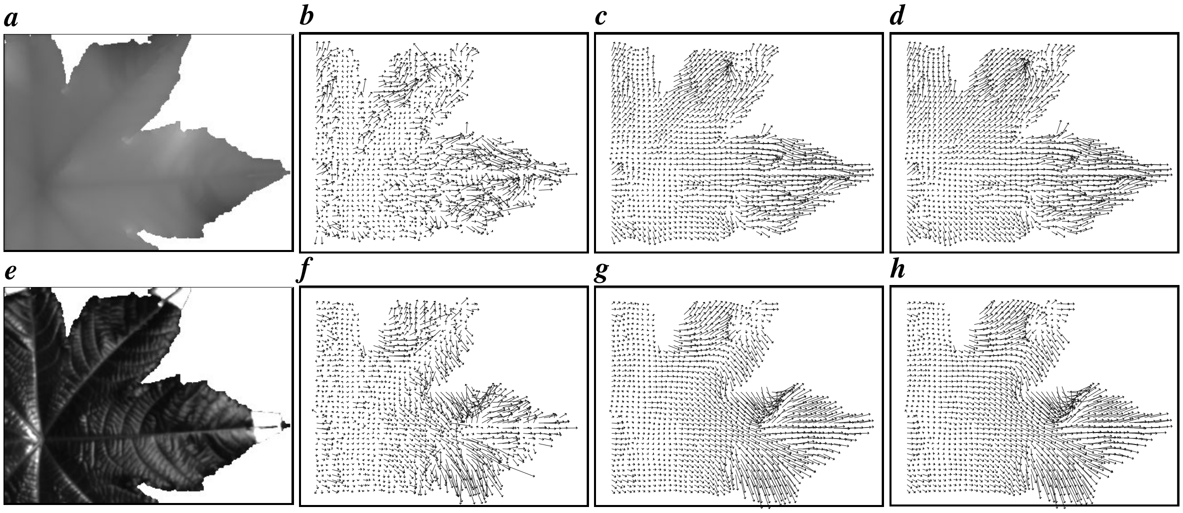


Figure 5.4: *Example regularised range flow on a castor oil leaf:* ***a*** *depth data and XY-components of the range flow field after* ***b*** *0,* ***c*** *100 and* ***d*** *1000 iterations.* ***e*** *depth data and XZ-components of the range flow field after* ***f*** *0,* ***g*** *100 and* ***h*** *1000 iterations.*

Another example on a rather noisy sequence of a castor oil leaf is shown in Fig. 5.4. Here we see that a larger smoothing parameter of $\alpha = 10$ can be used to regularise the estimated range flow. We also see that due to the availability of initial full flow estimates there is little change between 100 and 1000 iterations, Fig. 5.4 c,g and Fig. 5.4 d,h. In general we find that for range flow on castor oil leaves we usually reach convergence from 200 to a maximum of 500 iterations.

5.2.2 Dense Optical Flow

In the same way as for range flow we can use the TLS subspace data constraint in conjunction with the membrane regulariser to obtain dense and smooth optical flow fields. On continuous surfaces such a regularisation greatly enhances the optical flow estimation. This is illustrated on an example stereo application. Instead of the classic two camera system we employ a special experimental set-up with only one camera which is continuously moved from one stereo position to the other. The correspondence problem is solved by tracking each pixel over the acquired image sequence. Such a tracking of each individual pixel across the image sequence requires a dense and accurate velocity estimation. More details on this system can be found in Spies et al. [2000b].

Instead of the previously used global regularisation parameter α we can also adjust the influence of the smoothness term locally. This is again based on the confidence:

$$\alpha = \alpha_{min} + \omega_c(\alpha_{max} - \alpha_{min}) \; . \tag{5.38}$$

As ω_c varies between 0 and 1 the resulting smoothness factor α lies between α_{min} for low confidence and α_{max} for high confidence values. An example from this application is shown in Fig. 5.5. Here we used 39 image frames which were taken by a camera mounted on an arm that rotated by 34.16 degrees. The parameters used are $\alpha_{min} = 1$ and $\alpha_{max} = 10$. The accumulated flow field between the first and last image is shown in Fig. 5.5 c. This can then be used to recover the depth as shown in

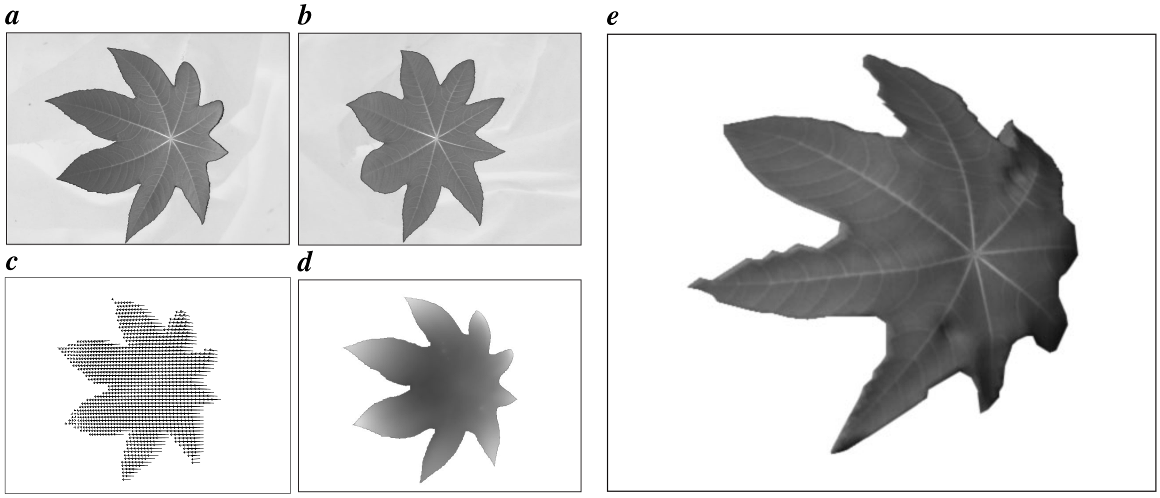


*Figure 5.5: Regularised optical flow to recover the structure of a leaf: **a** first image, **b** last image, **c** accumulated displacement between the two frames, **d** computed depth map and **e** texture mapped rendered structure.*

Fig. 5.5 d,e The resulting depth map seems to be qualitatively correct. However there are some dents in the leaf surface which are artifacts resulting from the unstabilised summation of the displacement vector fields. One way to improve the results is to use the accumulated displacement as starting value for a more sophisticated stereo matching scheme [e.g. Alvarez et al., 2000a].

5.2.3 Anisotropic Diffusion for Optical Flow in Noisy Image Data

This section demonstrates how the anisotropic diffusion method can be used to improve the optical flow estimation, in particular on noisy image sequences. To do so we present some results on the diverging and translating tree sequences. Figure 5.2 shows an example image of the tree sequences. We only examine the computed full flow, results for normal flow are practically identical. It turns out that the performance is not very sensible to the choice of parameters. In the given example we use $c = 5$, a time step size $\tau = 1$ and threshold $\tau_2 = 10$. The averaging in the structure tensor calculation is implemented using a 7-tap Binomial.

In order to quantify the performance of our algorithm we employ the angular error measure [Barron et al., 1994]. For this measure the 2D optical flow quantities are extended to 3D as $\boldsymbol{r} = (f_1\ f_2\ 1)^T$. The deviation between a correct flow field $\boldsymbol{r}_c$ and an estimated flow $\boldsymbol{r}_e$ is then described by:

$$E = \arccos(\boldsymbol{r}_c^T \boldsymbol{r}_e / \|\boldsymbol{r}_c\| \|\boldsymbol{r}_e\|)\ [^\circ]\ . \tag{5.39}$$

This error measure combines the errors in magnitude and direction into a single number and allows to compare different algorithms because of its wide spread use. It has, however, been criticised because symmetric deviations do not lead to the same error values [Otte and Nagel, 1994; Haußecker and Spies, 1999].

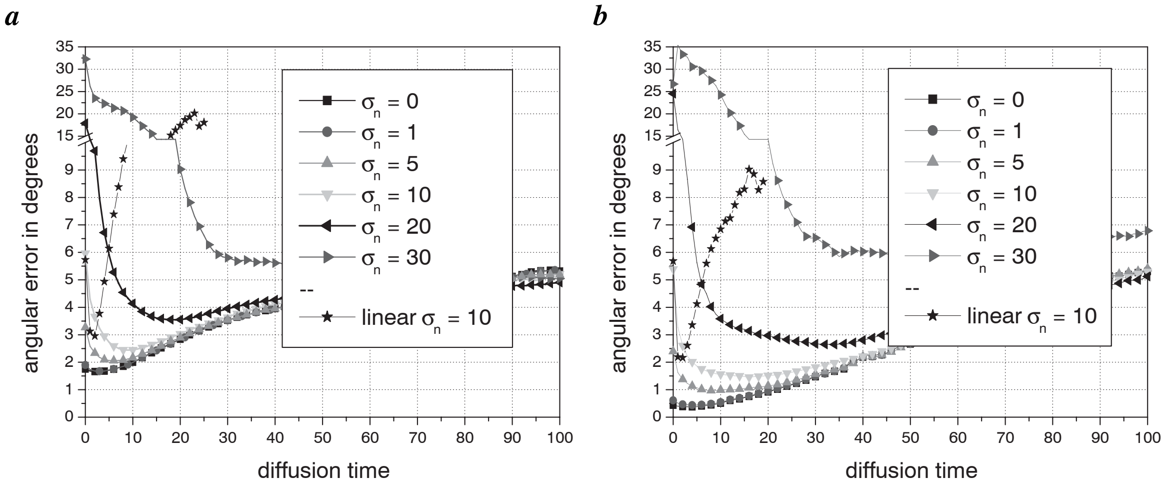


Figure 5.6: *Angular error on the tree sequences at various noise levels:* ***a*** *diverging tree,* ***b*** *translating tree.*

Apart from using the original sequence we introduce artificial noise with a normal distribution and standard deviations of $\sigma_n = 1, 5, 10, 20, 30$. The resulting error values are plotted as a function of the diffusion time in Fig. 5.6. Even though there is an improvement on the original sequences (diverging: $1.98° \rightarrow 1.71°$; translating: $0.72° \rightarrow 0.44°$) the real strength of the method comes into play for very high noise levels. Here the improvement in the computed optical flow field is quite dramatic. Additional noise with standard deviation of $\sigma_n = 10$ can almost completely be compensated for. Bear in mind that typical CCD cameras have noise levels around $\sigma_n \approx 1$. The improvement for higher noise levels is also quite dramatic. In Fig. 5.6 the angular error obtained when isotropic smoothing is applied for $\sigma_n = 10$ is shown as well. As expected the error initially decreases but then quickly rises again. Not surprisingly we find that orientation enhancing diffusion does yield better results.

It can be observed that with increasing diffusion time it eventually comes to some oversmoothing of the sequence. This in turn results in a decreasing estimation accuracy. Thus we need to determine a suitable stopping criterion. Figure 5.7 shows the mean confidence value, see Eq. (4.16), depending on the diffusion time. It can be seen that the confidence ceases to increase after a certain time. Hence we stop the diffusion once the increase is less than a threshold ϵ. The choice for this threshold depends on the value of σ, for $\sigma = 10$ we find $\epsilon = 0.003$ to be a good choice. The following table compares the thus obtained results (stop) to the best results achieved (min) on the diverging tree sequence:

σ_n	min E	diff. time	stop E	diff. time
0	1.66	2	1.68	4
1	1.67	3	1.68	4
5	2.06	6	2.06	6
10	2.44	9	2.45	8
20	3.55	19	3.59	16
30	4.99	64	5.72	34

Using this automated stopping procedure we get very close to optimal results. Also the stopping time is very close to the time where the best result is obtained. Only for extremely high noise levels a lower threshold leads to slightly better results.

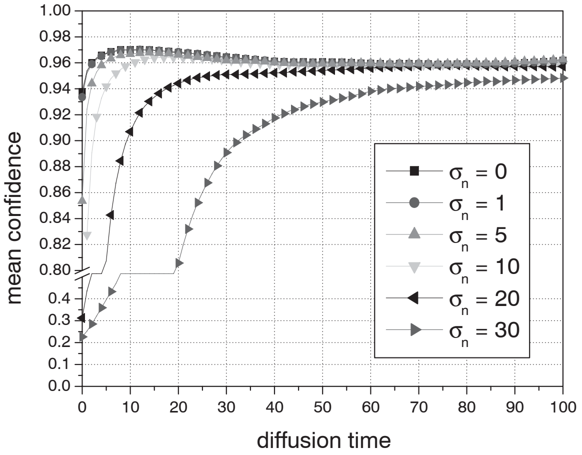


Figure 5.7: Mean confidence value for the diverging tree sequence.

5.3 Summary

This chapter introduced a variational framework that can be used to estimate dense full parameter fields. Towards this end a novel data term that restricts the regularised solution only in the subspace determined by the TLS algorithm has been motivated. Because range flow in leaves is continuous and smooth the membrane model can be used as a suitable smoothness term in this case. It has been shown that the combination of the subspace data constraint and the membrane smoothness model leads to a different smoothing in the estimated TLS subspace and its orthogonal complement. This scheme has been shown to yield smooth and dense range flow fields.

Additionally it has been shown how the variational framework can be extended to a truly anisotropic regularisation. The smoothing directions and appropriate smoothness strengths follow naturally from the structure tensor. The resulting diffusion process has been used very successfully to enable accurate optical flow estimation in noisy image sequences.

Chapter 6

Expansion (Growth) Estimation

There are a number of applications where one wants to determine whether the movement and deformation of an observed surface changes its surface area. Examples are the study of the effects of temperature changes or mechanical stress on materials and the observation of growth rates in biological systems. We focus on the latter for leaves and refer to Chap. 7 and Chap. 9 for practical aspects of this application. This chapter discusses the measurement of expansion or growth rates from velocity fields. First the usual definition of expansion as divergence in one to three dimensions is discussed and the limitations of the 2D approach for surfaces in 3D are illustrated. Then the area growth of a 2D surface embedded in 3D coordinates is evaluated and an error analysis made.

6.1 Growth as Divergence

In order to determine expansion rates we consider the observed object as a compressible fluid. The movement of this fluid is described by the measured displacement vector field $\boldsymbol{f}$. Note that for these considerations a full and differentiable velocity field is needed. Thus one can not use normal flows (Sect. 4.1.2) but need to employ a regularised flow field as obtained using the methods given in Chap. 5. Let us examine an element as depicted in Fig. 6.1 with a volume of $V = \mathrm{d}x\,\mathrm{d}y\,\mathrm{d}z$. The left and right faces move with velocity U and $U + U_x\,\mathrm{d}x$ respectively. Hence the change of the length $\mathrm{d}x$ becomes $\Delta\mathrm{d}x = U_x\mathrm{d}x$. Using the same argument for the two other directions we conclude that the overall change in the volume is given by:

$$\Delta V = U_x\,\mathrm{d}x\,\mathrm{d}y\,\mathrm{d}z + \mathrm{d}x\,V_y\,\mathrm{d}y\,\mathrm{d}z + \mathrm{d}x\,\mathrm{d}y\,W_z\,\mathrm{d}z = \mathrm{d}x\,\mathrm{d}y\,\mathrm{d}z\,(U_x + V_y + W_z)\ . \tag{6.1}$$

The rate of change in the volume follows to:

$$r = \frac{\Delta V}{V} = U_x + V_y + W_z = \mathrm{div}\ \boldsymbol{f} = \nabla^T\boldsymbol{f}\ . \tag{6.2}$$

In other words the rate of volume change is given by the divergence of the velocity field. The relative expansion (growth) rate is often expressed in % per time interval:

$$e = \mathrm{div}\ \boldsymbol{f}\cdot 100\ [\%/t]\ . \tag{6.3}$$

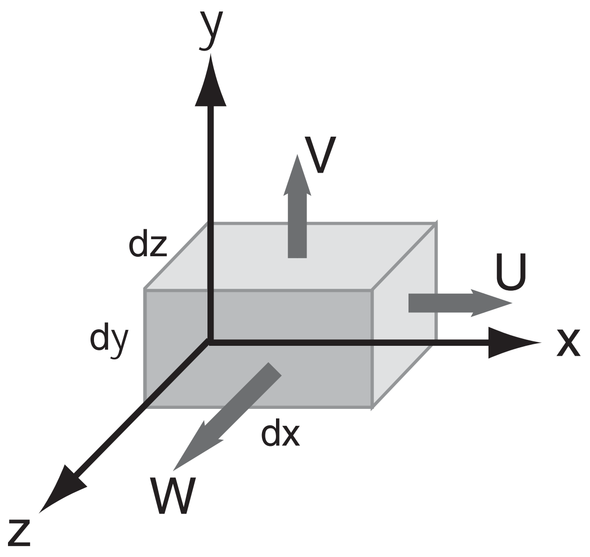


Figure 6.1: *Illustration of the volume change caused by a diverging velocity field.*

This equation could be used to measure the volume changes in sequences of volumetric data sets. Examples are PET [Klein and Huesman, 1997] or CT [Song et al., 1994] images. In computer vision, however, one does not often have volumetric data. Therefore the one and two dimensional cases are examined next.

6.1.1 Growth in 1D and 2D

Linear or one-dimensional expansion occurs for example in plant roots. If the coordinate system is aligned such that the movement is along the x-direction the relative growth rate is directly given by:

$$e = \partial_x u \cdot 100 \; [\%/t] \; . \tag{6.4}$$

This has been applied to study the growth patterns present in roots [Walter et al., 2001; Kirchgeßner et al., 2001]. On an example of a corn root the growth zone, located shortly after the tip, is clearly visible (Fig. 6.2). This zone retains its position relative to the tip and is moving with the root.

From the optical flow in an image sequence we can determine the areal expansion of the observed object by the 2D divergence:

$$e = (\partial_x u + \partial_y v) \cdot 100 \; [\%/t] \; . \tag{6.5}$$

Clearly the divergence only measures the actual growth if there is no movement towards the camera. The problems arising from this restriction when analysing leaves will be examined in the next section. Nevertheless can such a two-dimensional technique be used very successfully to study areal growth patterns in plant leaves [Schmundt et al., 1998; Schmundt, 1999; Walter, 2001].

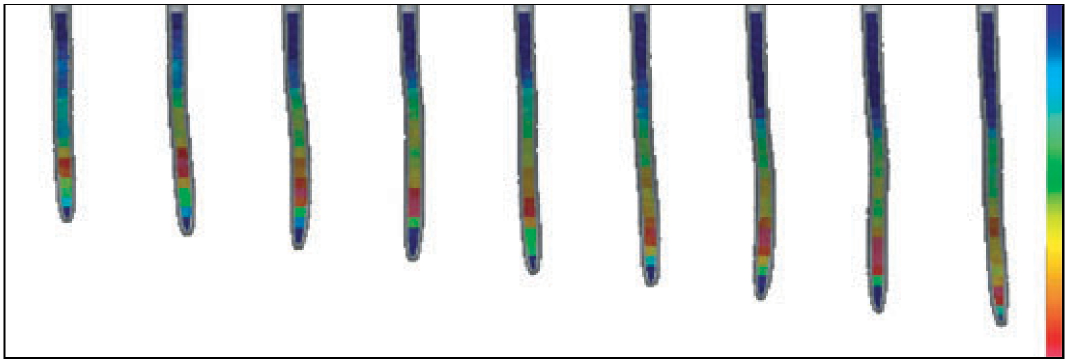


Figure 6.2: One-dimensional growth rate [$0, 30\%/h$] found on a corn root, the time between the images is 15 minutes and the observed length is $12\,mm$.

6.1.2 Limitations

The prerequisite of the above growth estimation technique is that the observed object lies in a fronto-parallel plane and stays in that plane at all times. But what is the effect on the estimated 2D growth rate if this assumption is violated. Under central projection an area $A = \Delta X \Delta Y$ gets projected on the image plane to an area $a = \Delta x \Delta y = \frac{f^2}{Z^2} A$. This projected area depends on the object distance as $\frac{\partial a}{\partial Z} = -2\frac{f^2}{Z^3} A$. Hence a change in the distance results in a change in the projected area even when the actual area A remains the same. A depth change of δZ results in a relative area change:

$$\frac{\delta a}{a} = -2\frac{\delta Z}{Z} \cdot 100\ [\%]\ . \tag{6.6}$$

For an example viewing distance of $Z = 300\,mm$ the error in the estimated growth ($e = \frac{\delta a}{a}$) becomes $1\%/h$ for a movement of $\delta Z = 1.5\,mm/h$. As will be seen in Chap. 9 vertical movements in this magnitude can occasionally occur even on leaves that are fixed in a plane.

So far we only considered a fronto-parallel plane moving in Z direction, but often the surface is tilted as well. The dependence of the observed world coordinate X on the sensor coordinate x for a locally planar surface has already been given in (3.11). Let the surface be oriented such that its slope k lies in X-direction, i.e. $k = Z_x/X_x$ and $Z_y = 0$. Then an area element A is projected on the image plane on an area element a:

$$a = \mathrm{d}x\ \mathrm{d}y = A\frac{f^2}{Z^2}(1 + k\,\tilde{x}) = A\frac{f^2}{Z^2}\left(1 + k\frac{X}{Z}\right)\ , \tag{6.7}$$

where we used $\tilde{x} = \frac{x}{f} = \frac{X}{Z}$. A change in depth δZ results in a relative area change of:

$$\left(\frac{\delta a}{a}\right)_Z = \frac{(\partial_Z a)\ \delta Z}{a} = -2\frac{\delta Z}{Z}\ \frac{Z + 1.5\,k\,X}{Z + k\,X} \cdot 100\ [\%]\ . \tag{6.8}$$

In comparison to (6.6) there is an additional factor that depends on the slope k, the position X and the depth Z. Only for small k and near the centre of the image this term does not contribute to the area change. Hence for a tilted surface a change in depth does not only result in an expansion on the image plane but this expansion is not even uniform.

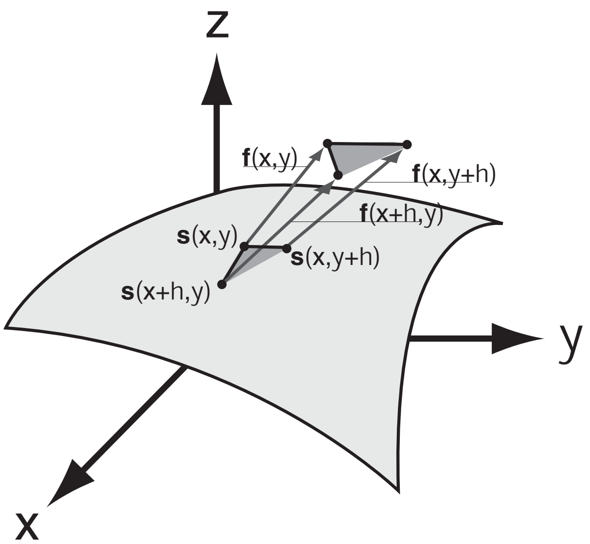


***Figure 6.3:** Computing the local surface area and its change caused by a non-uniform velocity.*

The projected area (6.7) also depends on the slope and the position. Thus a change in any of these will result in a perceived area change. Taking the derivatives with respect to these two variables we find the following relative area changes:

$$\left(\frac{\delta a}{a}\right)_k = \frac{X}{Z + k\,X}\,\delta k \cdot 100\ [\%]\ , \tag{6.9}$$

$$\left(\frac{\delta a}{a}\right)_X = \frac{k}{Z + k\,X}\,\delta X \cdot 100\ [\%]\ . \tag{6.10}$$

The further the distance in X from the origin (the centre of projection) the greater is the influence of a surface tilt. For a position of $Z = 300\,mm$ and $X = 30\,mm$ and an initially flat surface ($k = 0$) we get an expansion rate of 1% for a change in the slope by $\delta k = 0.1$. With $k = \tan\vartheta$ this corresponds to a tilt from $0°$ to $5.7°$.

The dependence of the growth error on a movement in X increases with the slope. If the patch is fronto-parallel a horizontal movement does not result in a perceived area change. But a horizontal movement of a tilted plane induces an area change in the image plane. On a surface tilted by $\vartheta = 15°$ ($k = 0.27$) at $Z = 300\,mm$ and $X = 50\,mm$ a growth rate of $1\%/h$ is created by a horizontal displacement of $11.6\,mm/h$. Such high speeds do not occur on leaves that are fixed in a plane, but they do on unconstrained leaves, see Chap. 9.

The discussed systematic errors make a growth analysis on non-flat surfaces, e.g. unconstrained leaves, with a single camera impossible. The next section discusses how the 3D structure can be taken into account to compute the expansion rate.

6.2 Growth on Surfaces

Let the considered surface $\boldsymbol{s}$ be given by a set of range data. Then the surface is parameterised in sensor coordinates x and y:

$$\boldsymbol{s} = \boldsymbol{s}(x, y) = [X(x, y)\, Y(x, y)\, Z(x, y)]^T \ . \tag{6.11}$$

We want to compute the local surface area and its change caused by a movement. The area near a surface point can be described by the area of the triangle created by moving a small distance h along the two parameter directions (Fig. 6.3):

$$a = \frac{1}{2}\Big|[\boldsymbol{s}(x+h, y) - \boldsymbol{s}(x, y)] \times [\boldsymbol{s}(x, y+h) - \boldsymbol{s}(x, y)]\Big| \ . \tag{6.12}$$

By the displacement vector field $\boldsymbol{f}$ this area is transformed into:

$$\begin{aligned} a' \;&=\; \frac{1}{2}\Big| \,[\boldsymbol{s}(x+h, y) + \boldsymbol{f}(x+h, y) - (\boldsymbol{s}(x, y) + \boldsymbol{f}(x, y))] \\ &\qquad\quad \times \quad [\boldsymbol{s}(x, y+h) + \boldsymbol{f}(x, y+h) - (\boldsymbol{s}(x, y) + \boldsymbol{f}(x, y))] \Big| \ . \end{aligned} \tag{6.13}$$

The relative change in the local surface area is given by the quotient of d and a. Taking the limit $h \to 0$ we obtain:

$$\mathrm{d}a = \lim_{h \to 0} \frac{\frac{1}{h}a'}{\frac{1}{h}a} = \frac{|\partial_x(\boldsymbol{s} + \boldsymbol{f}) \times \partial_y(\boldsymbol{s} + \boldsymbol{f})|}{|\partial_x \boldsymbol{s} \times \partial_y \boldsymbol{s}|} \ . \tag{6.14}$$

We can think of $a = |\partial_x \boldsymbol{s} \times \partial_y \boldsymbol{s}|$ as a local area element. The entire surface area of a regular surface can then be defined as the integral over these area elements [do Carmo, 1976]:

$$A = \int |\partial_x \boldsymbol{s}(x, y) \times \partial_y \boldsymbol{s}(x, y)| \; dxdy \ . \tag{6.15}$$

We are now ready to define the relative expansion rate as:

$$e = (\mathrm{d}a - 1) \cdot 100\ \% = \frac{a' - a}{a}\, 100\ \% \ . \tag{6.16}$$

This quantity can be computed from derivatives, obtained via convolution, of both the range data and range flow arrays. Both range flow and range data should be on the same scale, thus the same smoothing procedure is used on both data sets.

Figure 6.4 shows two examples and the thus computed expansion maps. The first example consists of a sequence of a yeast dough taken with a Biris range sensor at a sampling rate of one minute. One example depth map is given in Fig. 6.4 a. The computed range flow field (Fig. 6.4 b and c) shows a diverging X, Y velocity and a movement in Z that is higher on top of the object. This range flow is part of the 3D diverging velocity caused by an increase in volume. On most parts of the dough we find an areal expansion around $20\%/h$ (Fig. 6.4 d). There are some places showing an even higher expansion mainly at locations with a steep depth gradient. Even though the expansion might vary

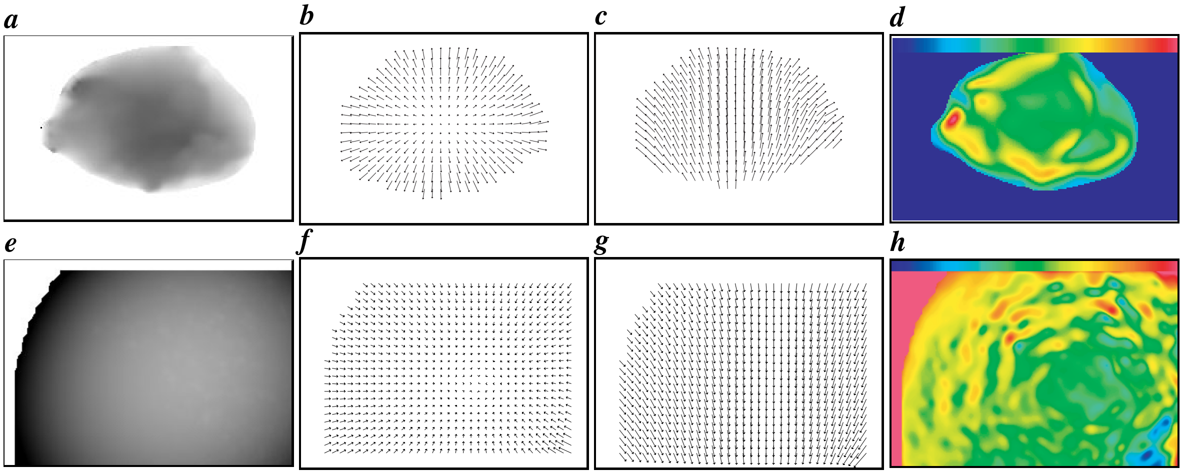


Figure 6.4: *Example expansion map on real sequences. An expanding yeast dough:* ***a*** *depth data,* ***b*** *XY-components of the range flow,* ***c*** *XZ-components of the range flow field and* ***d*** *expansion map in the range of* $[0, 40]\%/h$*. A shrinking air balloon:* ***e*** *depth data,* ***f*** *XY-components of the range flow,* ***g*** *XZ-components of the range flow field and* ***h*** *expansion map in the range of* $[-6, 0]\%/h$*.*

in the observed manner the extracted growth rate is probably not as reliable at these locations, see Sect. 6.2.2.

For the second example an air balloon is measured using a structured light system, see Fig. 6.4 e for an example depth map. During the observation time the air slowly vanishes and the balloon shrinks, scans are taken at one minute sampling intervals. This results in a velocity field with negative divergence in X, Y and an overall downward movement (Fig. 6.4 f and g). The expansion map (Fig. 6.4 h) shows a shrinkage of about $-3\%/h$. Again the expansion map is more noisy in regions with a steeper depth. This indicates that the computation of the growth is problematic in regions with a strong depth gradient. An analysis of the error in the estimated growth rate will be presented in Sect. 6.2.2.

6.2.1 Surface Expansion in 2D

Does the surface growth formula (6.16) reduce to the 2D formula (6.5) for the case of a fronto-parallel plane? Fronto-parallel implies $\partial_x Z = \partial_y Z = 0$. The change of the X and Y coordinates are constant and can thus be set to $\partial_x X = \partial_y Y = 1$, additionally $\partial_y X = \partial_x Y = 0$. If we further assume no movement in the Z direction the change in the area (6.14) is given by:

$$\mathrm{d}a = 1 + \partial_x U + \partial_y V + (\partial_x U \partial_y V - \partial_y U \partial_x V) \; , \tag{6.17}$$

resulting in a relative growth rate of:

$$e = (\partial_x U + \partial_y V + (\partial_x U \partial_y V - \partial_y U \partial_x V)) \cdot 100\% \; . \tag{6.18}$$

This is the previously determined 2D growth formula (6.5) plus an additional second order factor. This term can be explained by considering a unit square located at the centre of the coordinate system as shown in Fig. 6.5 a. The light grey regions form the area change given by the divergence. The darker corners correspond to the additional data term. However, because the derivatives of the velocity

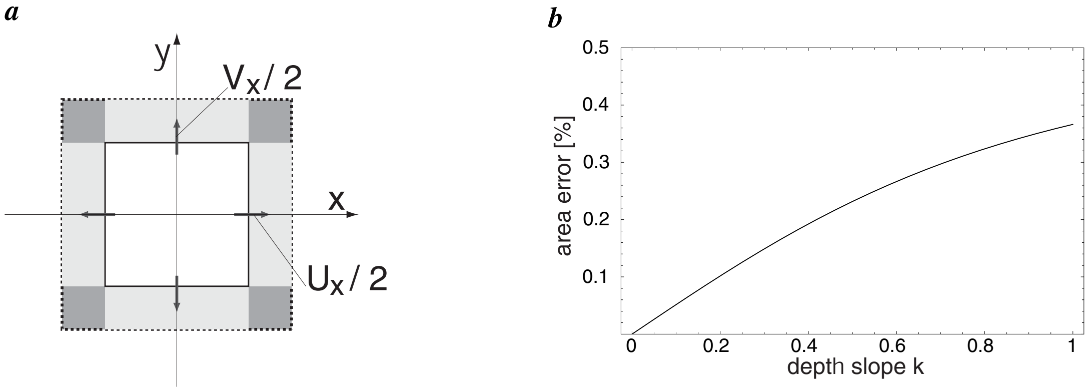


Figure 6.5: ***a*** *Interpretation of the second order term in the reduced 3D growth formula.* ***b*** *Relative error in the estimated surface area in dependence of the depth slope.*

are very small this term does not contribute significantly to the growth rate. Thus we obtain the desired result that for all practical purposes the more general surface expansion formula reduces to the classical 2D formula for horizontal movements on a fronto-parallel plane.

6.2.2 Error Analysis

The error in the growth estimation caused by noise in the range data and range flow is estimated next. An experimental accuracy analysis using synthetic data sequences will be given in Sect. 8.2.3.

First recall that the error in the derivative is simply given by the error in the data multiplied by a factor γ depending on the used derivative kernel (4.20). For example for the derivatives in the depth map we obtain:

$$\sigma_{\partial_x Z} = \sigma_{\partial_y Z} = \gamma \, \sigma_Z \quad \text{with} \quad \gamma = \sqrt{\sum_{i=1}^{n} \alpha_i^2} \, , \tag{6.19}$$

where the α_i are the filter coefficients used. In the following special derivative kernels optimised for directional invariance are used [Scharr et al., 1997; Jähne et al., 1999]:

$$\partial_x = (0.084,\, 0.332,\, 0,\, -0.332,\, -0.084)_x * (0.023,\, 0.242,\, 0.470,\, 0.242,\, 0.023)_y \, . \tag{6.20}$$

These filters result in a factor $\gamma = 0.28$. Next the error in the local area element a is discussed.

Because the lateral accuracy is about one order of magnitude higher than that in the depth (Sect. 2.1.1 and Sect. 7.3) only the noise in Z is considered. If we further assume no dependency of X on y and Y on x ($X_y = Y_x = 0$), the error in the area estimate follows from error propagation:

$$\sigma_a = \gamma \, \sigma_Z \sqrt{\left(\frac{\partial a}{\partial(Z_x)}\right)^2 + \left(\frac{\partial a}{\partial(Z_y)}\right)^2} \; 100\,[\%] = \gamma \, \sigma_Z \sqrt{\frac{Y_y^4 Z_x^2 + X_y^4 Z_y^2}{X_x^2 Y_y^2 + Y_y^2 Z_x^2 + X_x^2 Z_y^2}} \; 100\,[\%] \, . \tag{6.21}$$

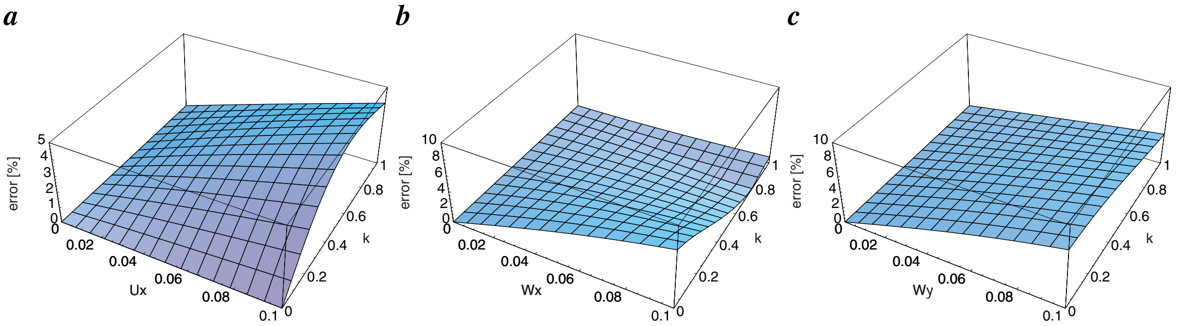


Figure 6.6: *Error in the estimated area change in dependence of the velocity gradient and the depth slope.*

Let the coordinate system be oriented such that $Z_y = 0$ and assume the change in X and Y to be equal: $X_x = Y_y = \Delta_{XY}$. Then the area error simplifies to:

$$\sigma_a = \gamma\, \sigma_Z \frac{\Delta_{XY} Z_x}{\sqrt{\Delta_{XY}^2 + Z_x^2}}\, 100\,[\%] = \gamma\, \sigma_Z \frac{k\, \Delta_{XY}}{\sqrt{1+k^2}}\, 100\,[\%]\,, \qquad (6.22)$$

where we used the slope $k = \frac{Z_x}{\Delta_{XY}}$ to express the error. Sensible values are a distance of $Z = 300\,mm$, a focal length of $12\,mm$ and a pixel size of $7.4\,\mu m$ (see Chap. 7) resulting in $\Delta_{XY} = 0.185\,mm/pixel$. Choosing $\sigma_Z = 0.1$, see Sect. 2.1 and Sect. 7.3, results in a relative error in the estimated area depending on the depth slope (Fig. 6.5 b). For higher slopes we can have a standard deviation in the estimated area of up to 0.4%. Such high slopes do for example occur at the steep edges of the examples shown in Fig. 6.4. On leaves we sometimes encounter slopes approaching $k = 1$ on the lobes, see Sect. 9.2.1.

To clarify the influence of the various variables they are first treated separately. Let's assume for the moment that the error in the depth dominates and use the same assumptions as above ($X_y = Y_x = 0$, $Z_y = 0$, $X_x = Y_y = \Delta_{XY}$). If the only change in the velocity field is in the X-component U in direction x ($U_x \neq 0$) then the error in the estimated growth rate becomes:

$$\sigma_e = \gamma\, \sigma_Z \frac{k\, U_x(U_x + 2\Delta_{XY})}{\Delta_{XY}^2 (1+k^2)^{\frac{3}{2}} \sqrt{U_x^2 + 2\, U_x \Delta_{XY} + (1+k^2)\Delta_{XY}^2}}\, 100\,[\%]\,. \qquad (6.23)$$

Similar formulas are obtained for the other variables. Under the above assumptions a change of V in y does not result in an estimation error because there is no change in depth along the Y-direction. The error in the growth rate as a function of the velocity gradient and the slope is shown in (Fig. 6.6), where the same numeric values as above for Δ_{XY}, σ_Z and γ are used. The error depending on the horizontal velocity components increases with the surface slope and the velocity gradient (Fig. 6.6 a), the curves for U_y and V_x are very similar. For high slopes ($k > 0.5$) the standard deviation in the estimated growth can be in the range of 5%. However, here and in the following one should bear in mind that the velocity gradients are seldom above 0.05. On a flat surface such a gradient corresponds to an expansion rate of 5%.

A change in the vertical velocity component W yields an increasing error in the growth rate even for a vanishing slope (Fig. 6.6 b and c). If we assume that there is no additional constant velocity a gradient in the vertical velocity corresponds to a rotation of the surface. Thus we conclude that a

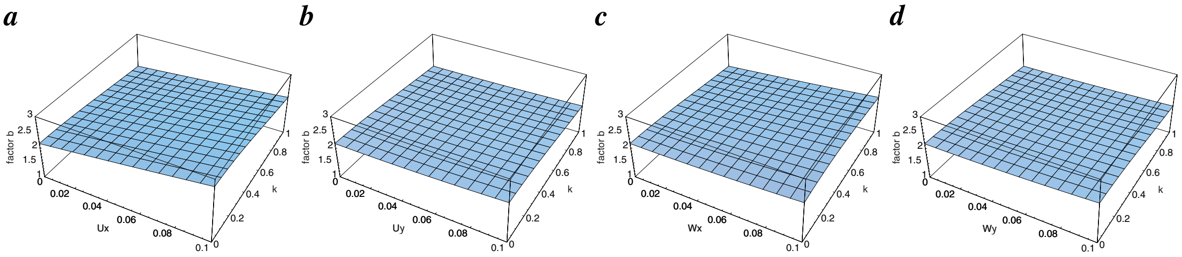


Figure 6.7: *Factor relating the error in the range flow estimation to the error in the growth rate, depending on a velocity gradient and the surface slope.* ***a*** U_x*,* ***b*** U_y*,* ***c*** W_x *and* ***d*** W_y*.*

fast tilt of the observed surface can severely deteriorate the computed growth rate. For an increasing slope in x the effect of the increasing vertical velocity is reduced (Fig. 6.6 b). As has to be expected the surface slope in x-direction has little influence on the growth error caused by a change of W in y-direction (Fig. 6.6 c).

The above considerations assumed that the error in the depth dominates the error in the growth estimate. For high surface slopes this is true but for less inclined surfaces and small velocity gradients the error in the range flow field is an important factor. Assuming the same error σ_f in all three velocity components the error in the estimated expansion rate becomes (here the 3D data is assumed to be error free):

$$\sigma_e = \gamma\,\sigma_f \frac{\sqrt{|\partial_x \boldsymbol{f}|^2 + |\partial_y \boldsymbol{f}|^2 + 2\,U_x\Delta_{XY} + 2\,V_y\Delta_{XY} + 2\,k\,W_x\Delta_{XY} + 2\,\Delta_{XY}^2 + k^2\,\Delta_{XY}^2}}{\Delta_{XY}^2\sqrt{(1+k^2)}}\,100\,[\%]\,. \tag{6.24}$$

The error in the growth rate is directly proportional to the error in the range flow $\sigma_a = b\,\sigma_f$ with a factor b that depends on both the velocity gradient and the surface slope. Varying only one velocity gradient component and the surface slope at a time results in relatively flat curves for this factor (Fig. 6.7). There is the general trend that higher velocity gradients result in higher growth errors and that an increasing slope reduces the error. Only for the change of the vertical velocity component in direction of the surface orientation W_x (Fig. 6.7 c) we find a slight increase with the slope for higher velocity gradients. Overall the deviation in the estimated expansion, caused by the error in the velocity alone, is given by about 2 times the deviation in the velocity itself. An error of $\sigma_f = 0.01\,mm/h$ in the velocity leads to a standard deviation in the growth of $2\%/h$.

Such a high variation is caused by the fact that growth is in fact a second order process. The absolute error in a derivative is about one third that in the data. But as the derivative values are at least a factor 10 smaller than the data values this increases the relative error by a factor 3.

Finally the combined error caused from both uncertain depth and uncertain velocities is examined qualitatively. As the error propagation formulas become very lengthy they are omitted here. The growth error for two example velocity errors $\sigma_f = 0.01$ and $\sigma_f = 0.02$ (Fig. 6.8) show that for higher velocity noise levels the error is dominated by the velocity uncertainty. In this regime the growth estimation is not very accurate and only the integration over a large area can reduce the error. For lower noise in the velocity the error is dominated by the velocity for small slopes and small velocity gradients. But for high velocity and depth gradients the error in the depth dominates the

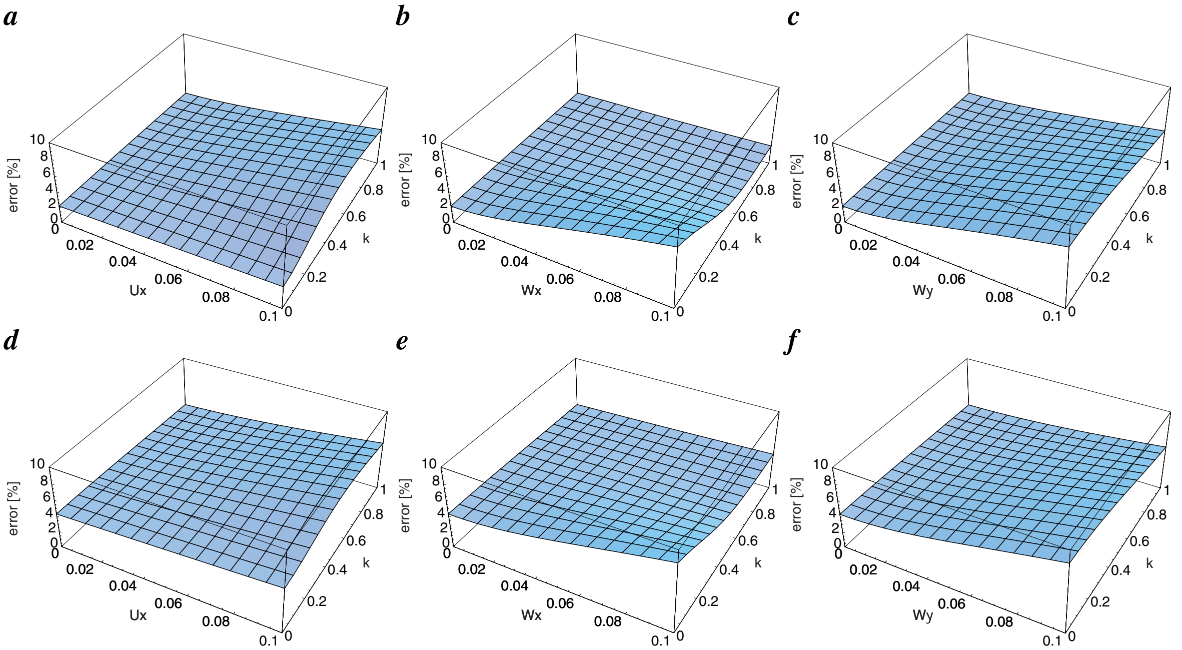


*Figure 6.8: Relative error in the estimated surface in dependence of the depth slope and the velocity gradient for two velocity noise levels: $\sigma_f = 0.01$ (**a** ,**b** ,**c**) and $\sigma_f = 0.01$ (**d** ,**e** ,**f**).*

growth estimation. Thus the growth estimation becomes unreliable for higher slopes. On leaves the angle ϑ between the surface normal and the Z-direction can reach 20°, this corresponds to a slope of $k = \tan\vartheta = 0.34$. In this case the growth uncertainty is still dominated by the velocity error for horizontal movements. However high gradients in the vertical velocity lead to high errors in the growth even for zero slope. This poses problems for the growth estimation on leaves at times when the tips of the lobes are bending rapidly, see Sect. 9.4.

Reducing the Error by Averaging.

In Chap. 7.3 an accuracy analysis of the estimated range flow will be given. For real range data the relative error remains below 1% only as long as the movement does not exceed a speed of $0.5\,mm/frame$ on structured light data. At a sampling rate of 3 minutes this implies an absolute error in the velocity less than $\sigma_f = 0.01\,mm/h$. The resulting maximal uncertainty in the growth estimate thus becomes $\sigma_e = 2\%/h$. As this lies in the same range as the growth found on real leaves we need to compute the growth on a coarser spatial scale than the velocity itself.

Under the assumption that the growth rate does not change rapidly averaging will reduce the statistical error. Such an averaging can be done very efficiently using a Gaussian pyramid [Jähne, 1995]. Here the Pyramid is realised by applying a 5-tap Binomial pre-filter then sub-sampling and finally applying a 3-tap Binomial post-filter [Haußecker and Jähne, 1993]. The resulting effective 1D mask is given by: $\frac{1}{64}(1,\ 4,\ 8,\ 12,\ 14,\ 12,\ 8,\ 4,\ 1)$. This filter is applied first in x and then in y-direction yielding an overall noise suppression factor $\gamma = 0.158$. Each level of the pyramid reduces the noise by this factor. On the second level the uncertainty in the growth estimate thus reduces to $\sigma_e = 0.025 \cdot 2\%/h = 0.05\%/h$ which is quite acceptable.

6.3 Summary

Growth estimation from the divergence of a 2D velocity field and the limitations of this technique were discussed. Then a method for the estimation of areal growth on 2D surfaces embedded in 3D space has been introduced. From error propagation we concluded that for high surface slopes and high gradients in the vertical velocity the uncertainty in the estimated growth is dominated by the error in the depth data. Under extreme circumstances the absolute error in the growth can be up to $10\%/t$. For more common situations with surface slopes below 0.3 and gradients in the vertical velocity below $0.05\,mm/(t\,pixel)$ expansion estimates are dominated by the error in the velocity. Then the variation in the estimated growth is roughly $\sigma_e = \frac{2\,\sigma_f}{[mm]}\;100\;[\%/t]$, where σ_f is the uncertainty in the velocity. For a deviation in the growth of maximal $\sigma_e = 0.1[\%/h]$ the error in the velocity has to remain below $\sigma_f = 0.0005\,mm/h$. As this accuracy is difficult to achieve in practice (Chap. 8) the estimated growth has to be averaged in order to reduce the estimation error.

Part II

Experiments

Chapter 7

Experimental Setup

The following chapter introduces the two range sensors that have been employed in this work. The first one is a Biris laser range finder, made available by Agriculture Canada, that was used during the initial part of this project at the University of Western Ontario in London, Canada. The main experiments, however, where done using a structured light sensor at the Institute for Botany at the University of Heidelberg. After describing the practical realisation of the two sensors their range estimation accuracy is analysed.

7.1 Biris Laser Range Finder

The principle of the *Biris* range sensor has already been described in Sect. 2.1.2. The experiments reported here are carried out using a Sensor bought from Vitana Corporation, Ottawa. The sensor head is shown in Fig. 7.1 a, it comprises both the laser and the camera with the double aperture mask, compare Fig. 2.3. From the projected laser line the 3D structure along this line can be recovered. In this particular sensor the line is sampled at 256 positions (pixels) along the x-axis. In order to obtain two-dimensional data arrays these scan lines have to be moved across the scene.

We used two different mounting stages for the Biris sensor. The first one is a pan-tilt unit that rotates the sensor perpendicular to the scan line. This unit can be mounted on a tripod and thus allows for a high flexibility, see Fig. 7.1 b. In the other setup, shown in Fig. 7.1 c, the sensor is placed on a linear positioning stage which moves above the scanned object.

The sensor is calibrated at the manufacturer and as there are no moving parts in the sensor head this calibration remains accurate for an extended period of time. However, to ensure the consistency and accuracy of the measurement, it is recommended that an additional calibration is performed after the sensor has been moved or more than 90 days (recommended by the manufacturer) have passed since the last calibration. This second calibration can correct for some small deviations from the internal factory calibration. Such an additional calibration is achieved by scanning a special planar target, provided by the manufacturer, at several orientations. The target consists of a glass plate with points marked at known relative positions. From the locations of these points in each image the relative 3D position of camera and laser can be recovered [Sturm and Maybank, 1999; Zhang, 2000].

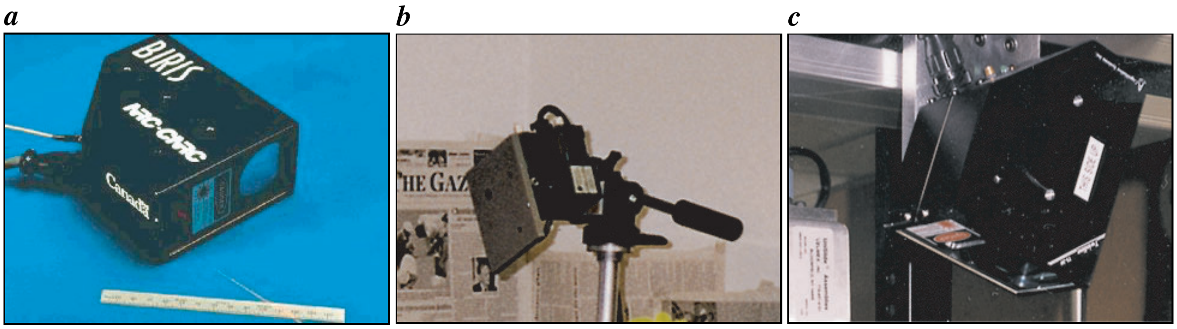


Figure 7.1: *Biris range sensor:* ***a*** *sensor head,* ***b*** *pan-tilt unit and* ***c*** *mounted on a linear positioner.*

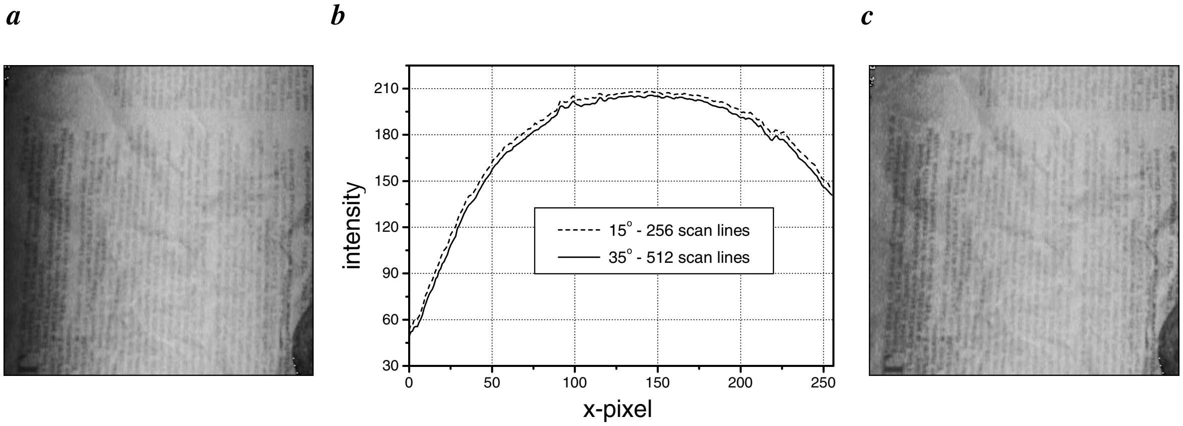
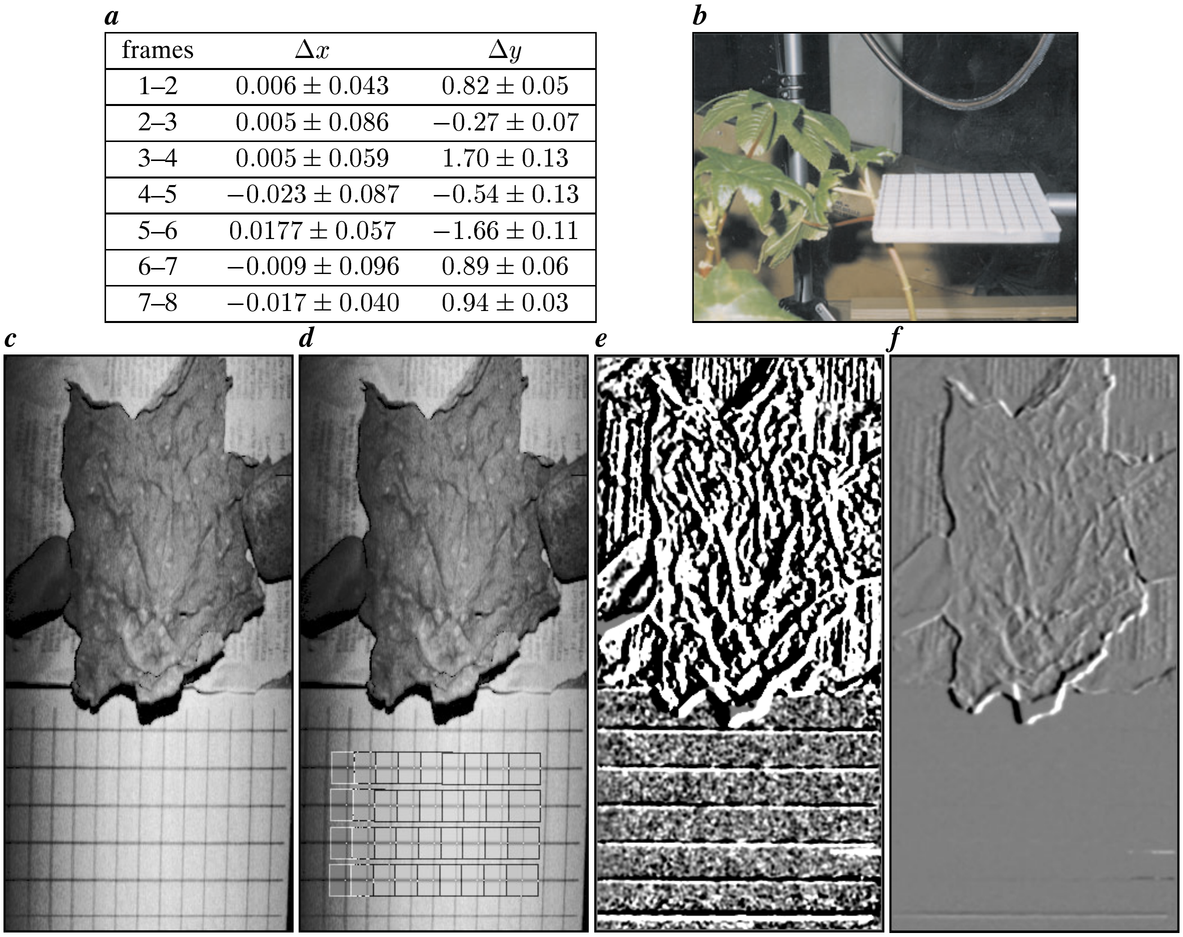


Figure 7.2: *Inhomogeneous intensity of the Biris sensor:* ***a*** *original intensity data,* ***b*** *two average intensity scan line profiles and* ***c*** *corrected intensity image.*

The calibration defines the world coordinate system in which the 3D structure is measured. In our case the coordinate system is attached to the sensor head. Therefore it is quite difficult to align the measurement coordinate system with an external coordinate system, for instance defined by the vertical axis.

7.1.1 Intensity Normalisation

The intensity data returned by the Biris range sensor is not homogenous, see Fig. 7.2 a. We find much higher values in the centre of the image than at the borders, especially on the left side. This introduces an artificial gradient which leads to a systematic error in the recovered velocity field. A simple way to account for such a fixed intensity gradient is by normalisation with the intensity data from a uniform object. As each scan line is taken with nearly the same part of the optics and the CCD chip it appears that they have the same intensity pattern. Figure 7.2 b shows the averaged scan lines of a white sheet of paper from two scans with different total pan angle. The averages agree well with each other within their standard deviations, which is between 1 and 5 percent. Deviations from the mean scan line are higher in the darkest area to the left. We then use the extracted mean scan line to scale the intensity. The result can be seen in Fig. 7.2 c which is a corrected version of Fig. 7.2 a. Clearly the intensity images become more homogeneous after scaling than before.



frames	Δx	Δy
1–2	0.006 ± 0.043	0.82 ± 0.05
2–3	0.005 ± 0.086	-0.27 ± 0.07
3–4	0.005 ± 0.059	1.70 ± 0.13
4–5	-0.023 ± 0.087	-0.54 ± 0.13
5–6	0.0177 ± 0.057	-1.66 ± 0.11
6–7	-0.009 ± 0.096	0.89 ± 0.06
7–8	-0.017 ± 0.040	0.94 ± 0.03

Figure 7.3: *Range data registration:* ***a*** *mean displacements between consecutive scans.* ***b*** *Calibration target placed next to a leaf.* ***c*** *Example intensity image,* ***d*** *grid points fitted to the image,* ***e*** *temporal derivative without registration,* ***f*** *temporal derivative with registration.*

7.1.2 Data Registration

Unfortunately the movement of the pan or linear unit and the acquisition by the sensor head are not synchronised. Therefore the movement can begin before or after the range acquisition is started, as a result consecutive scans *randomly* jump up and down.

Figure 7.3 a gives the displacement found by analysing grid points on a stationary calibration target. The positions were extracted from the intensity data by a Levenberg–Marquardt fit of a grid model function to the data [Garbe, 1998]. The data was taken with 512 scan lines and 25 degrees pan angle. The average displacement is based on 36 grid points tracked over 8 images.

The mean variation in the x-position is 0.006 ± 0.003 pixel, and the mean variation in the y-position is 0.72 ± 0.04 scan lines. It can be observed that the x-positions are almost perfectly in line, as expected of course. The y-positions on the other hand show major random deviations, due to the described lack of synchronisation. While the standard deviation of the y-displacements is typically in the same order as that of the x-displacements (less than 0.1 pixel) we sometimes encounter higher values. For the linear stage we get very similar results which are omitted here. In the following a correction method for these irregular jumps in y-direction is presented. However, using grid points as targets the achievable accuracy is limited by the above variations.

In order to align the range data we scan a stationary calibration target together with the actual moving object (Figure 7.3 b). Examining the positions of the grid points distributed over the entire field of view showed no dependency of the y-displacement on the image location. Thus it is assumed that the movement extracted on the stationary calibration part is identical to that on the object part. The data is then resampled using cubic interpolation. In order to suppress outliers and noise a 5×5 median filter followed by a 3-tap Binomial smoother is employed prior to the warping.

Figure 7.3 c shows an example intensity image of the calibration target and a scene that is moved horizontally by a positioning table. The fitted grid points and the computed time derivatives with and without registration are given in Fig. 7.3 d,e and f. On the stationary calibration target these derivatives should vanish. This clearly isn't the case without registration but is achieved after resampling. On the object part of the not registered sequence we also encounter very high time derivatives caused by the jumps from one frame to the next. After registration the time derivatives correspond to the movement in x-direction.

7.2 Structured Lighting

The structured lighting system, compare Sect. 2.1.3, put together at the Institute for Botany of the University of Heidelberg uses a *MiniLin* projector obtained from ABW GmBbH Frickenhausen, Germany. This special miniature line-projector employs a grating made of chrome on a 6×6 mm glass slide to produce predefined line patterns. A cold light source is then attached to the projector via a glass fiber connector for illumination. The complete sensor consists of the projector and a standard CCD camera. Here we use a Black-and-White Sony XC-55 progressive scan camera with square pixels of size $7.4 \times 7.4\,\mu m^2$. For both the camera and the projector a 12 mm lens is used. Because of their correction even for near infrared light we use Schneider-Kreuznach CNG 1.4/12 lenses. The resulting system is shown in Fig. 7.4 a. An example line pattern projected on a leaf is shown in Fig. 7.4 b. The sensor is mounted in a growth chamber where the observed plant can be placed underneath. Figure 7.4 c and d shows the system viewing a fixed and free leaf respectively.

7.2.1 Near IR Illumination

In order not to disturb the plant it has previously been found that an illumination in the near IR is advantageous as the leaf shows no physiological response in this waveband [Schmundt et al., 1998; Schmundt, 1999; Walter, 2001]. Thus LEDs with an emission peak at 950 nm are used for the observation of plants. Because standard CCD cameras can capture light of wavelengths up to 1050 nm they can be employed as imaging devices in such a setup.

However, in the structured light setup a projector is necessary and LEDs are less useful in this case. Standard halogen light bulbs typically produce even more intensity in the near IR than in the visible waveband. So called cold light sources use a special filter that cuts off the IR part of the

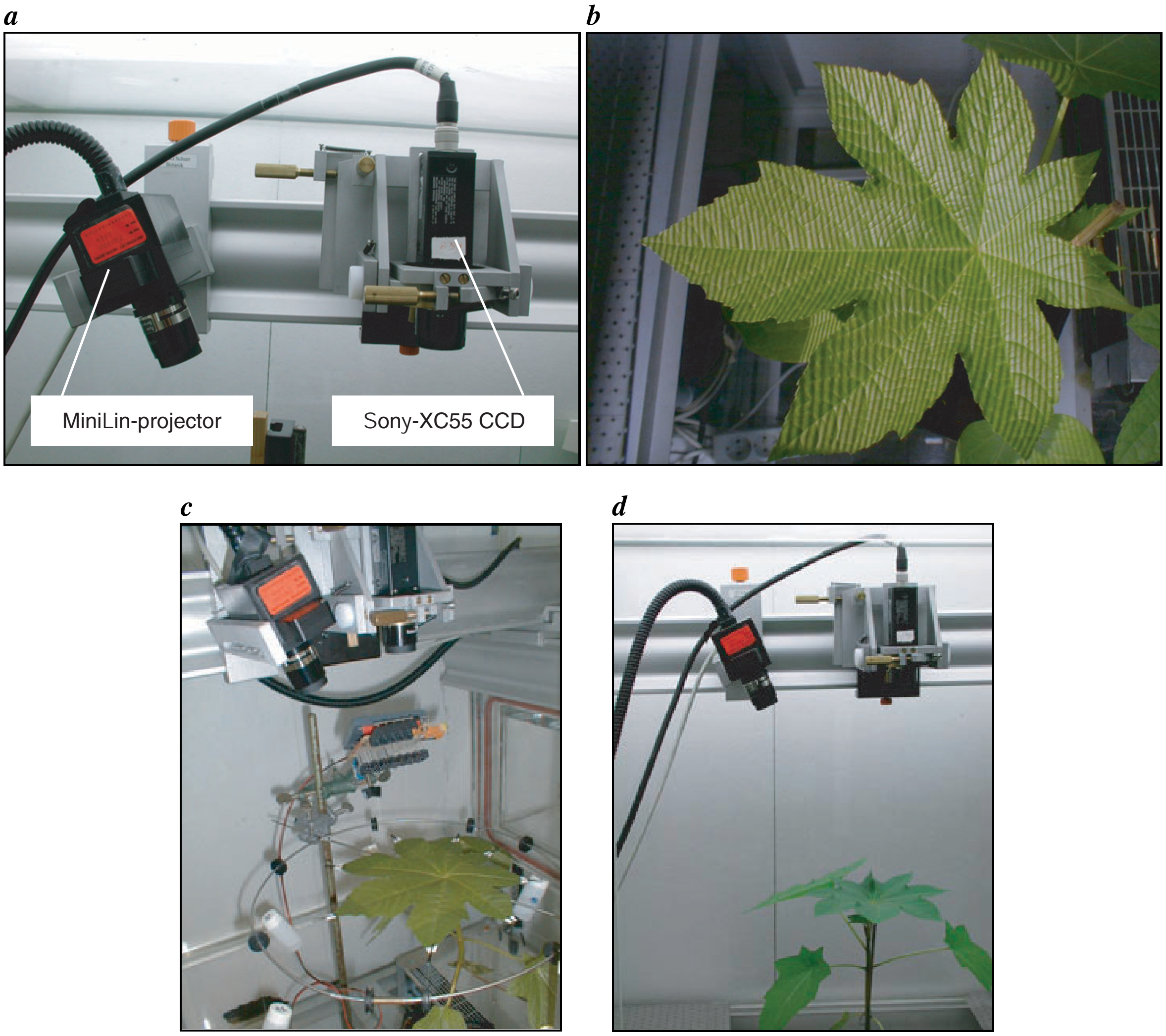


Figure 7.4: Structured lighting: ***a*** *setup with projector and camera,* ***b*** *example line structure projected on a leaf,* ***c*** *the sensor viewing a fixed and* ***d*** *a free leaf.*

spectrum. Hence all that has to be done to obtain a light source in the desired waveband is to replace the IR cut-off filter with one that has the opposite effect.

When using glass filters they have to be thick enough to tolerate the high intensity and associated heat production. This comes at the cost of a significantly reduced intensity. Unfortunately the sensibility of the CCD chip reduces towards the near IR. Therefore a better choice is to use a so called hot-mirror as a replacement of the IR cut-off filter. Such a mirror reflects only the visible light and transmits the IR. The special mirror used here has a transmittance curve as shown in Fig. 7.5 a. Additionally, the cut-off filter that is originally placed in front of the CCD chip in the camera has to be removed. In order to block off the ambient light we replace it by a RG9 filter glass that transmits only the longer wavelengths, see Fig. 7.5 b.

Examples of leaf intensity data using the standard visible lighting (Fig. 7.6 a,b) and the described near IR version (Fig. 7.6 c,d) look rather different. The near IR illumination has the additional benefit

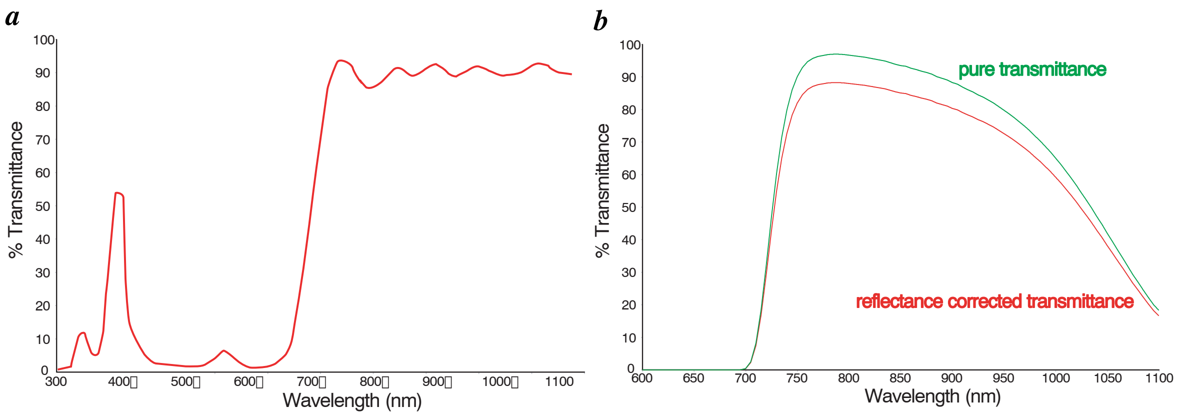


Figure 7.5: *Near IR illumination:* ***a*** *hot mirror (source: Lot-Oriel) and* ***b*** *RG9 glass filter (source: Schott) transmittance curves.*

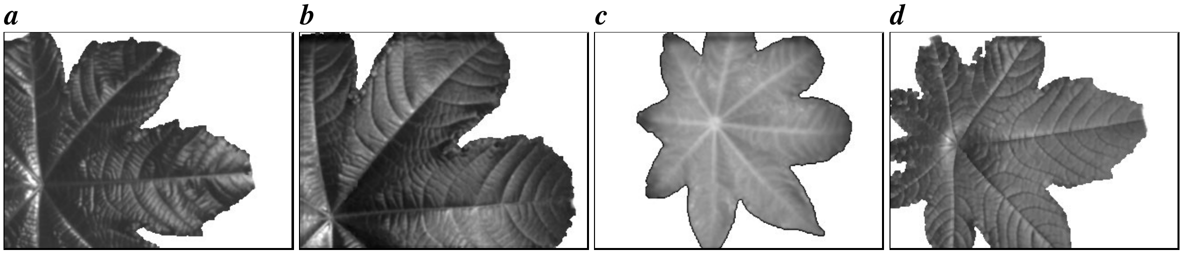


Figure 7.6: *Example intensity data. Visible illumination:* ***a*** *and* ***b*** *. Near IR illumination:* ***c*** *and* ***d*** *.*

that the leaves show less specular reflectance in this waveband. However, because the CCD camera is less sensible we need to increase the camera amplification, which introduces more noise. Therefore we do not use single images but rather the average of ten consecutive frames for every projected line pattern. This obviously increases the time needed to obtain one 3D scan. Yet a complete scan can still be achieved in less than 5 seconds which is sufficiently fast for the study of plant leaves.

7.2.2 Calibration

For an accurate estimation of the 3D scene both the internal and external parameters of camera and projector have to be known. Internal parameters are the focal length, principal point, aspect ratio and geometric distortion factors. The external parameters contain the positions relative to an external world coordinate system in terms of translation and rotation. More on geometric calibration of cameras and projectors can be found in [Faugeras, 1993; Luhmann, 2000; Wiora, 2001].

The world coordinate system that corresponds to the measurement coordinates is defined by a calibration target. The required three-dimensional target can be constructed from one planar target that is displaced by a known movement between several acquisitions. Here this movement is restricted to a translation along the plane normal. Fig. 7.7 a shows such a calibration target mounted on a manual positioner. In order to increase the accuracy a motorised positioning stage for the vertical displacement is used as well. An image of the calibration plate as seen by the camera is shown in Fig. 7.7 b. The x and y coordinate axes on the target define the world X and Y axis. The displacement of the target

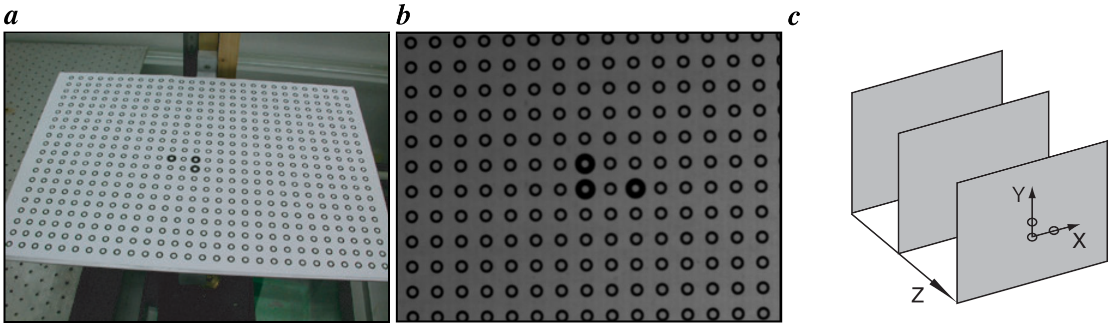


Figure 7.7: Calibration of the structured light sensor: ***a*** *target on a manual positioner,* ***b*** *target as seen by the camera and* ***c*** *world coordinate system.*

defines both the direction of the Z axis and the units used, here we always work with mm. The resulting world coordinate system is illustrated in Fig. 7.7 c. Note that because the target is moved towards the sensor higher Z values correspond to points closer to the camera.

7.3 Range Estimation Accuracy

Next we investigate the errors that have to be expected in the captured range data. For both sensors we first look at the 3D reconstruction error and then at the statistical noise in the data.

Reconstruction Error.

As we do not possess a three-dimensional target with known ground truth we are using planar objects to asses the quality of the 3D data. This is done by first fitting a plane to the data. The deviation of the measured data from the fitted plane indicates the accuracy of the measured data points. A plane is readily described by:

$$a\,X(x,y) + b\,Y(x,y) + c\,Z(x,y) + d = 0\;. \tag{7.1}$$

As there are errors in all coordinates we will use the total least squares method presented in Chap. 4 to estimate the plane parameters, see in particular Sect. 4.2.1. The resulting data vector becomes $\boldsymbol{d} = [X\;Y\;Z\;1]^T$ and the parameter vector is $\boldsymbol{p} = [a\;b\;c\;d]^T$. From our theoretical considerations in Sect. 2.1.1 we expect the error in the lateral coordinates to be about one order of magnitude smaller than that in the depth, see Sect. 2.1. The constant term (the one) is of course error-free. Therefore we use $\sigma_z = 1.0$, $\sigma_x = \sigma_y = 0.1$ and $\sigma_1 = 0.0001$ to construct the right hand equilibration matrix $\boldsymbol{W}_r = \operatorname{diag}\frac{1}{\sigma_i}$ in the TLS estimate, compare Sect. 4.1.4. As there are always missing data points and outliers we use the robust algorithm presented in Sect. 4.1.5. With the parameter in the Tukey biweight M-Estimator set to $r = 3$ and the number of iterations to a maximum of ten.

From the estimated parameters we obtain the plane normal as $\hat{\boldsymbol{n}} = \frac{1}{a^2+b^2+c^2}[a\;b\;c]^T$. With $d' = \frac{d}{a^2+b^2+c^2}$ the plane equation becomes $\hat{\boldsymbol{n}}^T\boldsymbol{x} + d' = 0$. The distance of a point $\boldsymbol{x}$ from the plane is then given by:

$$\delta = \hat{\boldsymbol{n}}^T(\boldsymbol{x} - \boldsymbol{x}_0) + d'\;, \tag{7.2}$$

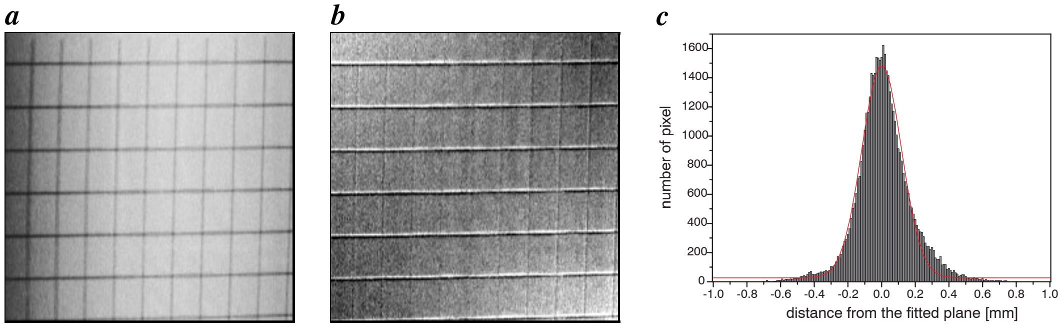


Figure 7.8: *Fitting a plane to a planar calibration target using the Biris sensor:* ***a*** *intensity image,* ***b*** *distance between measured 3D point and the fitted plane* $[-0.5, 0.5]\,mm$ *and* ***c*** *histogram of the distance.*

with an arbitrary point on the plane $\boldsymbol{x}_0$. Because vertical planes can not be measured $c = 0$ can not occur, thus we simply take $\boldsymbol{x}_0 = [0\ 0\ \frac{-d}{c}]^T$. As δ can take both positive and negative values we can use a map to reveal systematic deviations from the plane. In order to obtain a single number indicating the accuracy of the measured plane we compute the average of the absolute value of the distance$|\bar{\delta}|$ in mm.

Noise.

The noise level in the measured data can be determined by taking a number of repeated scans of the same stationary object. From the deviation of the mean values we can directly compute the noise in all four data channels X, Y, Z and I. Apart from averaged values we can examine the noise at each pixel to see if the noise is homogeneous.

7.3.1 Biris

According to the companies calibration report for the particular sensor head employed the achieved accuracy is about $0.1\,mm$ for objects in a distance of $250\,mm$. Similar results were also reported by Beraldin et al. [1995]. We will not provide a thorough analysis of the performance but only verify that the errors remain in the same order of magnitude in the actual experiments.

To investigate the accuracy of the sensor we examine the data obtained on the planar registration target, see Fig. 7.3 b and Fig. 7.8 a. This target consists of a $10\,mm$ thick aluminium plate upon which a self adhesive sheet of paper containing the grid is glued. When looking at the distance δ from the fitted plane in Fig. 7.8 b we clearly see the structure of the grid, even though there is no such depth variation. This is because the grid lines shadow the position of the projected laser line in the camera image. As a result there is a small jump in the depth. The error is far greater along the horizontal than on the vertical grid lines because the laser line lies in that direction.

Computing the average on 11 such data sets results in a mean distance to the plane of$|\bar{\delta}| = 0.12\,mm$. However on surfaces with a more diffuse reflectance, such as leaves, we are more likely to encounter deviations as found on the white areas. This can be approximated by taking the median rather than the mean of the absolute distance values. Doing so results in an average value of$|\bar{\delta}| =$

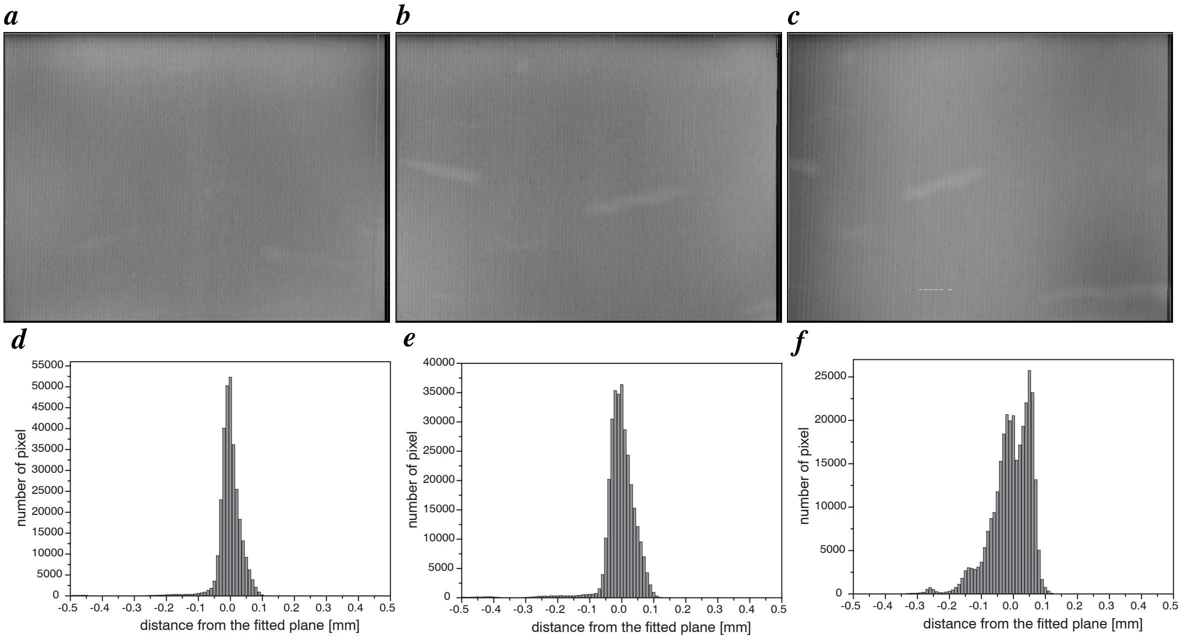


Figure 7.9: Distance of the measured 3D point on a planar target from the fitted plane for three examples in the range [-0.5,0.5] mm and the corresponding distance histograms.

$0.09\,mm$. From the histogram of the distance values (Fig. 7.8 c) we find that there is a slight bias towards positive deviations. This can also be observed in the distance map (Fig. 7.8 b) where the upper left and lower right corners appear brighter. This may either be caused by a slight bending of the attached sheet of paper or indeed hint at a small systematic error in the sensor.

From sequences of stationary targets we compute the variation in the X, Y, Z, I values. Doing so we find a fluctuation in the X coordinate of $\sigma_X \approx 0.0035\,mm$ and an error in the depth of $\sigma_Z \approx 0.03\,mm$. Due to the necessity of warping the images, compare Sect. 7.1.2, the error in Y becomes $\sigma_Y \approx 0.05\,mm$. When the intensity is scaled to lie in the range of $[0, 255]$ we find a standard deviation of $\sigma_I \approx 1.2$ greyvalues.

7.3.2 Structured Lighting

In order to examine the accuracy of the structured light sensor we performed two calibrations. Both are for a depth range of $150\,mm$ roughly placed between 450 and $300\,mm$ away from the camera. For the first calibration (**C1**) 11 plane positions at $5\,mm$ apart are used with the movement done by a manual positioner, see Fig. 7.7. The second calibration setup (**C2**) uses the same calibration plate now mounted on a motorised linear stage and 21 planes spaced $2.5\,mm$ apart from each other. This was done to see if the less complicated manual calibration is sufficiently accurate. The accuracy of the vertical positioning is in the order of $0.05\,mm$ for the manual and below $0.001\,mm$ (specification by the manufacturer) for the motorised stage.

In both setups we measured a glass plate covered with a self-adhesive sheet of paper at several orientations. In order to evaluate whether the plane orientation influences the accuracy we report the angles between the surface normal of the fitted plane and the three coordinate axes:

$$\alpha = \arccos([1\ 0\ 0]\,\hat{\boldsymbol{n}})\ ; \qquad \beta = \arccos([0\ 1\ 0]\,\hat{\boldsymbol{n}})\ ; \qquad \gamma = \arccos([0\ 0\ 1]\,\hat{\boldsymbol{n}})\ . \tag{7.3}$$

For the two calibration setups we find the following mean absolute distances of the measured data points from the fitted planes:

C1 - manual positioner

$\alpha\,[°]$	$\beta\,[°]$	$\gamma\,[°]$	$\|\bar{\delta}\|\ [mm]$
90.0	91.8	1.8	0.044
95.4	91.7	5.7	0.039
97.3	91.6	7.5	0.039
106.3	91.1	16.3	0.021
121.9	91.2	32.0	0.049
89.0	91.3	1.7	0.049
84.7	91.5	5.5	0.039
79.2	91.5	10.9	0.031
72.2	91.3	17.9	0.028

C2 - motorised positioner

$\alpha\,[°]$	$\beta\,[°]$	$\gamma\,[°]$	$\|\bar{\delta}\|\ [mm]$
90.5	90.2	0.5	0.036
89.9	98.5	8.5	0.036
90.7	81.9	8.1	0.028
80.9	90.0	9.2	0.031
98.0	90.4	8.0	0.036
90.1	88.8	1.2	0.037
99.0	89.5	9.0	0.038
81.7	88.3	8.5	0.032
89.1	98.7	8.8	0.037
90.5	80.3	9.7	0.032

As an averaged absolute deviation we find $|\bar{\delta}| = 0.035 \pm 0.009\,mm$ in the first and $|\bar{\delta}| = 0.034 \pm 0.003\,mm$ in the second case. This is in good agreement with findings by the manufacturer of the line projector [Wolf and Wolf, 1997]. We see that there is no significant difference in the mean performance but the results using the second calibration are more consistent. However this may also be caused by the larger variation in the plane orientations encountered in the first test case. In any way we find the 3D reconstruction accuracy to be well below $0.1\,mm$.

In order to determine if there are systematic errors in the reconstruction we look at the distance δ as a map of the sensor coordinates. Three such example maps are given in Fig. 7.9. For **C2** the map given in Fig. 7.9 a is quite homogeneous apart from the upper and right border. For a moderate inclination angle we find a similar behaviour for **C1**, see Fig. 7.9 b. This expresses itself also in the almost balanced histogram of these two distance maps (Fig. 7.9 d,e). When observing leaves they are usually in the centre of the imaged area, hence the problematic regions are not used in this application.

Using **C1** on a more tilted plane ($\alpha = 122°$, $\beta = 91°$, $\gamma = 32°$) results in the distance map and histogram given in Fig. 7.9 c and f. At the left border we find that the distance is negative, in the measurement coordinate system the Z values are estimated too low. This means the distance to the camera is slightly overestimated for higher distances, compare Sect. 7.2.2.

Noise.

We will now look at the noise level found by examining long sequences of stationary data. There are two different lighting environments that have to be considered. To simulate daylight the plants are illuminated using fluorescent tubes which have a relatively low emission in the near infrared. However there still is a significant change observed by the camera. Therefore we examine the noise level for the dark and light environment separately. To see whether there is a difference between a uniform

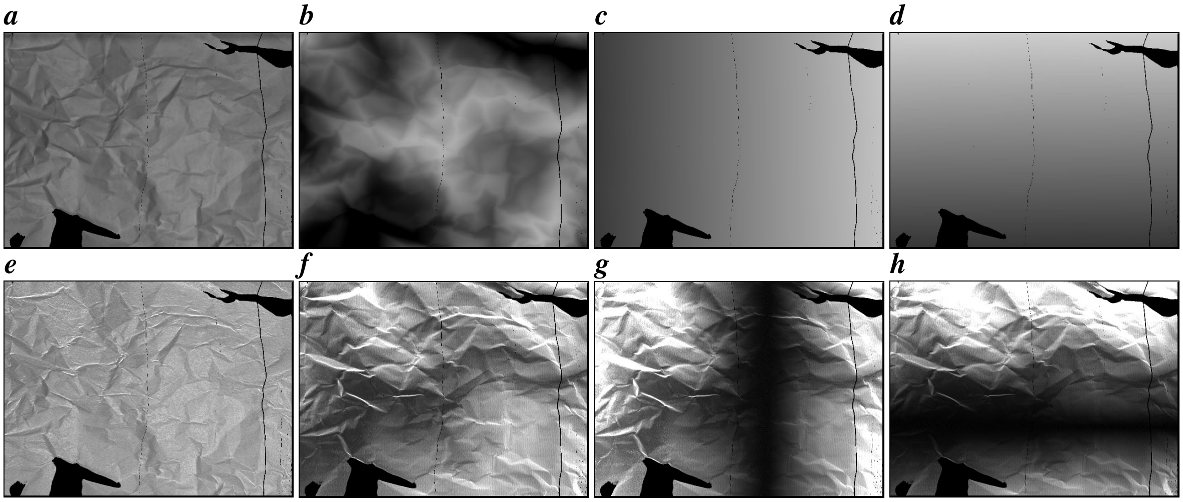


Figure 7.10: *Noise on a stationary scene. Mean data:* ***a*** *intensity,* ***b*** *depth,* ***c*** *X and* ***d*** *Y. Standard deviation:* ***a*** σ_I $[0, 2]$ *greyvalues,* ***b*** σ_Z $[0, 0.05]\,mm$*,* ***c*** σ_X $[0, 0.005]\,mm$ *and* ***d*** σ_Y $[0, 0.005]\,mm$.

scene and one with varying depth and intensity we examine both a flat sheet of paper and a scene with a crumpled sheet of paper. To find out if the laboratory environment has an impact the flat sheet is analysed during the day when there are other light sources in the surrounding. The average noise levels and their standard deviations computed from 200 scans at one minute intervals are as follows:

scene	σ_X $[10^{-3}mm]$	σ_Y $[10^{-3}mm]$	σ_Z $[10^{-3}mm]$	σ_I greyvalues
crumpled (light)	2.356 ± 0.004	2.187 ± 0.030	26.39 ± 0.12	1.2392 ± 0.0001
crumpled (dark)	1.599 ± 0.026	1.383 ± 0.015	17.00 ± 0.11	0.6789 ± 0.0003
flat (light)	1.373 ± 0.020	1.050 ± 0.014	13.13 ± 0.04	2.1540 ± 0.0009
flat (dark)	1.204 ± 0.008	0.947 ± 0.021	11.71 ± 0.07	5.1006 ± 0.0013

The variation in X and Y is about one order of magnitude below that in Z which agrees with the previous theoretical considerations in Sect. 2.1.1. We observe that for both scenes the noise in the 3D coordinates is lower when the additional illumination is turned off. However the effect is more pronounced for the crumpled paper. In general we see that the noise in X, Y, Z is significantly higher for the more realistic crumpled paper scene.

This observation does not hold for the intensity where we find higher noise levels on the flat sheet of paper. Additionally the noise in I dramatically increases without the illumination. While the crumpled sequences were taken with all light blocked off, the flat data was captured during the day without blocking the external light. In the latter case both the room light and the sunlight coming through the windows introduce changes in the intensity. The effect of these external light sources is more prominent when the plant illumination is not turned on. Therefore the dark period in the experiments should coincide with the actual night and the experimental chamber should be shielded against other light sources.

Let us look at the noise distribution dependent on the sensor coordinates. The resulting noise maps are shown in Fig. 7.10 for the crumpled paper example viewed with the illumination on. At the black areas the sensor did not yield a reliable estimate. We find a clear dependence of σ_I (Fig. 7.10 e) on the mean intensity (Fig. 7.10 a), this is a well known fact from radiometric calibration of CCD cameras [Healey and Kondepudy, 1994]. For σ_Z the dependence is not on the mean depth structure but rather on the slope. However the noise in Z is well below $0.1\,mm$ everywhere. For σ_X and σ_Y we observe a strong systematic behaviour. Both are greater for high and low values of the respective coordinates. Yet, as the actual noise values are quite low in X and Y ($< 0.01\,mm$) this is not so problematic as it looks at first sight.

Another way to examine the accuracy in the depth estimation is to move a plane a known distance along the Z direction. From five such movements over $5\,mm$ each the average change in the measured depth is found to be:

calibration	$\bar{\Delta Z}\ [mm]$
C1	5.030 ± 0.015
C2	5.003 ± 0.011

Apparently using the motorised positioner for calibration makes a difference. While there is a slight overestimation of the depth change for **C1** (manual) this is not the case using **C2** (motorised). Again the error in the measured depth is well below $0.1\,mm$.

7.4 Summary

This chapter presented the two sensors employed to capture the range data used throughout this thesis. After a description of the experimental setup the accuracy of the recovered data has been investigated. Even though the noise depends on both the scene and the calibration method we conclude that $\sigma_Z \approx 0.1\,mm$ and $\sigma_X \approx \sigma_Y \approx 0.01\,mm$ are conservative estimates. The noise in the intensity depends on the environment and typically lies in the range of $\sigma_I \approx 1 - 2$ greyvalues.

Chapter 8

Range Flow Accuracy

The purpose of the following chapter is to establish the accuracy the recovered range flow field. This will be exploited systematically for synthetic data with known amounts of noise added. The thus obtained results give a lower bound on the performance that can be expected on real data. Therefore we also examine the accuracy on real sequences where the true movement is known.

8.1 Error Measures

In order to quantify the results for range flow the following error measures are used. Let the correct range flow be denoted as $\boldsymbol{f}_c$ and the estimated flow as $\boldsymbol{f}_e$. The first error measure describes the *relative error* in the velocity magnitude:

$$E_r = \frac{|\,(|\boldsymbol{f}_c| - |\boldsymbol{f}_e|)\,|}{|\boldsymbol{f}_c|} \cdot 100\ [\%]\ . \tag{8.1}$$

Because E_r measures only the difference between the estimated and the correct velocity magnitude, no deviation from the correct direction is captured. Therefore we use the *directional error* as a second error measure:

$$E_d = \arccos\left(\frac{\boldsymbol{f}_c \cdot \boldsymbol{f}_e}{|\boldsymbol{f}_c|\,|\boldsymbol{f}_e|}\right)\ [^\circ]\ . \tag{8.2}$$

The value of E_d directly gives the angle in 3D between the correct velocity vector and the estimated vector and thus describes how accurate the correct direction has been recovered. Apart from the averaged error values we report their standard deviation to determine what range of error values has to be expected.

8.2 Results on Synthetic Data

First the achieved accuracy on some synthetic data sequences is evaluated. For the TLS normal flow types we use some simple test data based on orthogonal projection, see below. Otherwise more realistic test data of a planar surface and a sphere are employed. The computation of such data sets as seen

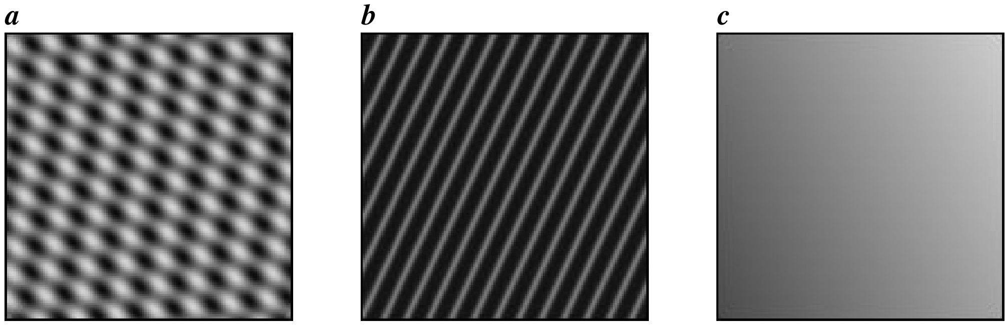


__Figure 8.1:__ Synthetic depth data to investigate the normal flow: __a__ sinusoidal grid for full range flow, __b__ line pattern for line flow and __c__ an oriented plane for plane flow.

by a sensor with perspective projection is described in Appendix D. In order to model the structured light sensor we use a focal length of $f = 12\,mm$, a pixel size of $7.4 \times 7.4\ \mu m^2$ and 256×256 sensor elements, see Sect. 7.2. To eliminate all border effect, especially for the regularised solution, we only use the inner 200×200 pixel to compute averaged error values.

For the synthetic plane we choose a viewing distance of $300\,mm$ and typically an angle between the surface normal and the Z-axis of $\vartheta = 5°$, see Sect. D.1 for more details. The intensity textures consists of a sinusoidal plaid pattern (D.20) with a wavelength of $1\,mm$ in both directions. Thus the intensity variation is of the same spatial variation as that caused by the veins on real leaves.

The second type of synthetic data used consists of a sphere with a radius of $300\,mm$ with its centre initially placed $700\,mm$ away from the camera. The generation of this kind of test data and the corresponding correct flow fields is detailed in Sect. D.2. The intensity on the sphere is described as a sinusoidal plaid in terms of the spherical angles of the surface point, see Eq. (D.31). Here we choose wavelengths of $\lambda_\theta = 1°$ and $\lambda_\phi = 30°$.

8.2.1 TLS Range Flow

The accuracy achieved using the TLS estimation framework developed in Chap. 4 is examined first. Next we look at the quality of the normal flow estimates. However, as we are mainly interested in the recovered full flow we will then only examine the full flow results in more detail.

Normal Flows.

In order to investigate the principal performance characteristics of plane, line and normal flow, see Sect. 3.3 and Sect. 4.3.1, we use some simple test sequences constructed using an orthogonal projection model. The depth data used (Fig. 8.1), were a sinusoidal plaid pattern for full flow, narrow spaced lines with Gaussian profiles for line flow and an oriented plane with surface normal $[0.7\ -0.34\ -1]^T$ for plane flow. No intensity data is used here.

We want to establish the accuracy of the velocity estimation for controlled amounts of random Gaussian noise. Thus we set the flow magnitude to vary between low and quite high displacements.

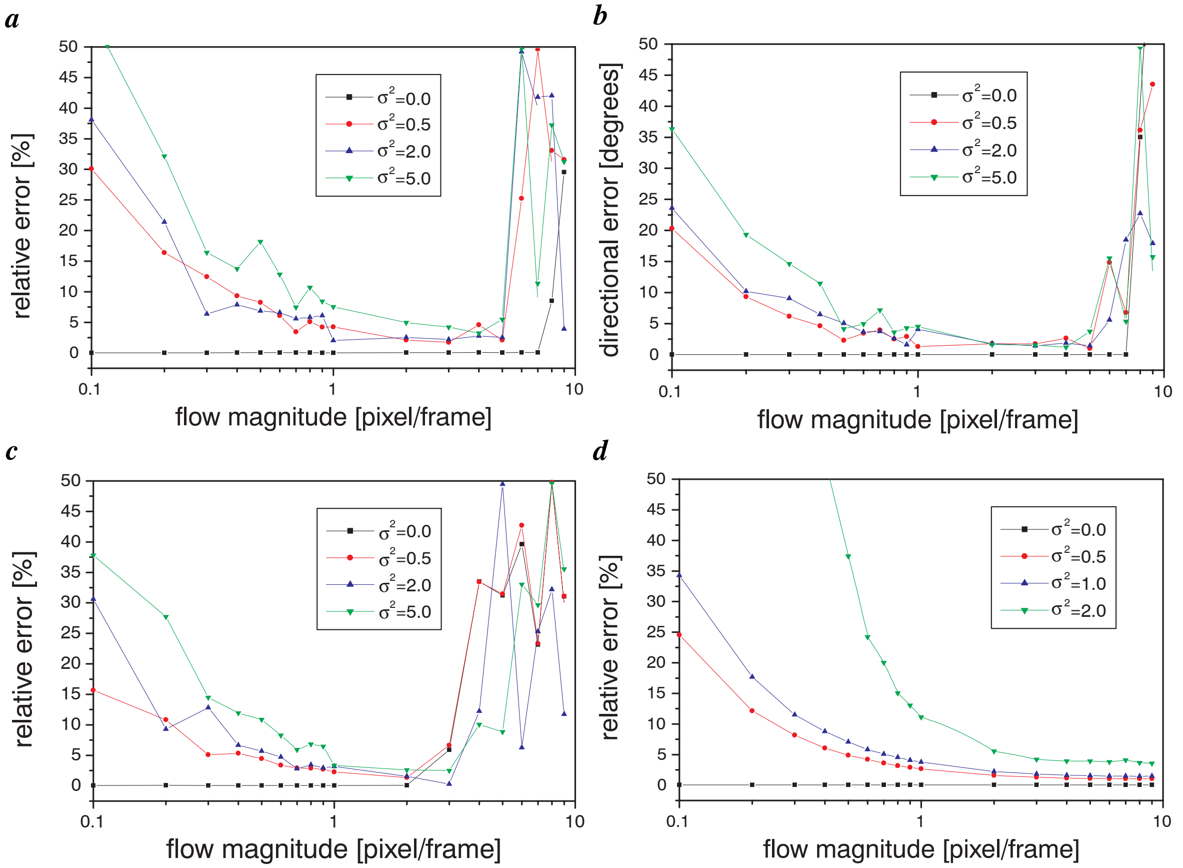


Figure 8.2: *Errors in the estimated normal flows for different noise variances σ^2 as a function of the correct magnitude:* ***a*** *,***b*** *magnitude and direction error for full range flow,* ***c*** *magnitude error for line normal flow and* ***d*** *plane normal flow.*

For each of these values five different directions of translation are chosen at random. The results reported are the average values.

Figure 8.2 shows the obtained errors for the various types of flow fields. Thresholds used were $\tau_2 = 0.1$ for noise variance $\sigma^2 \leq 1$, $\tau_2 = 1.0$ for $\sigma^2 = 2$ and $\tau_2 = 1.5$ for $\sigma^2 = 5$ (Sect. 4.1.1). Firstly it may be observed that the error in the direction correlates well with that in the magnitude (Fig. 8.2a,b). Full flow may reliably be estimated in the range of 0.5 to 5.0 pixel/frame. Line flow on the other hand can not be estimated reliably above 2.0 pixel/frame on this particular data. Both upper limits are due to aliasing. Therefore these limits will depend on the actual data, however both point and line structures will always contain considerable high frequency content. Planes on the other hand only contain low frequencies, thus plane flow can be recovered correctly even for quite high displacements, Fig. 8.2d.

From Fig. 8.2a,b,c it can be seen that both full and line flow calculation is not very sensitive to additional noise, which does not hold for plane flow, Fig. 8.2d. This is due to the moderate slope of the plane used, for higher gradients we observe less noise sensitivity. Yet, we do not normally encounter such high surface gradients. Thus we conclude that plane flow can only be computed reliably for low noise levels or large movements.

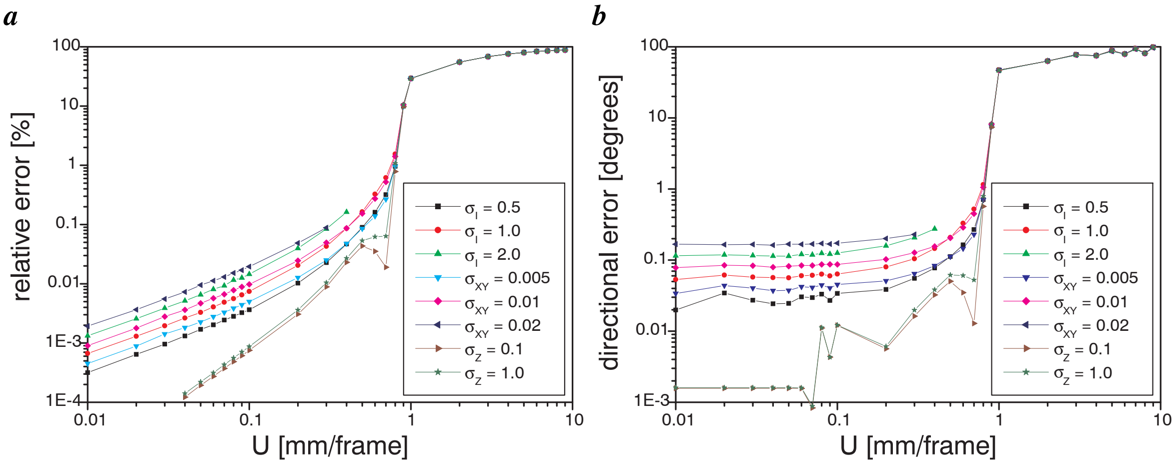


Figure 8.3: *Error depending on noise in the individual data channels:* ***a*** *relative magnitude and* ***b*** *direction error for movements in X direction.*

In the above examples a simple orthographic projection model was used. In this case range flow is mathematically identical to optical flow with an additional source term, compare Sect. 4.2.3. However, on real range data we are likely to encounter much lower noise levels (Sect. 7.3). In the following we will use more realistic movements expressed in $mm/frame$ and a projection model corresponding to a real range sensor.

Full Flow.

We continue by analysing the full flow estimation performance using the simulated structured light sensor. In the following the intensity data is always incorporated using $\beta = 1$ (Sect. 4.1.3). How does noise in the four data channels influences the accuracy of the recovered velocity. The results for a translation in X-direction (only U is nonzero) on the plane tilted by $\vartheta = 5^\circ$ are shown in Fig. 8.3. The results for other movements are practically identical and not shown here. For both the relative magnitude error and the directional error we observe the same qualitative behaviour. The range flow estimation fails for movements above $1\,mm/frame$. This is due to aliasing and thus dependent on the data. In this case the intensity structure limits the maximum velocity. However on real leaves we do observe vein structures with a width around $1\,mm$ and thus a maximum velocity of $1\,mm/frame$ is quite sensible. See also Sect. 8.3 for some results on real data.

We find that the results do not depend very much on the noise in the Z data but are much more influenced by the noise in the intensity and X, Y coordinates. In particular we see that a rise of σ_{xy} from 0.005 to 0.02 mm decreases the estimation accuracy by almost one order of magnitude. Hence we conclude that on more or less planar data, we consider leaves to be locally planar, it is not the depth accuracy but that in the other coordinates which limits the velocity estimation. In the following we will examine the performance for three typical noise situations, compare Sect. 7.3:

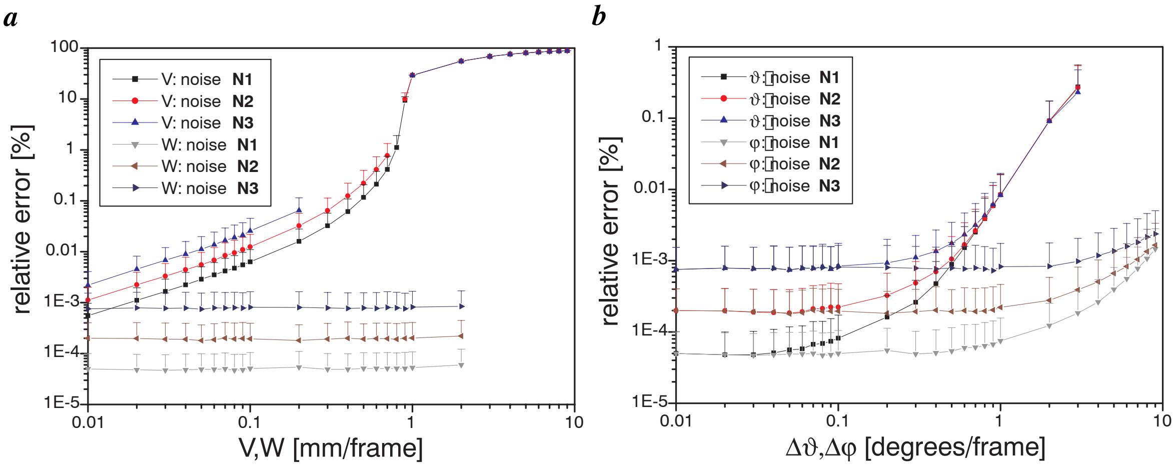


Figure 8.4: Relative magnitude error for different kinds of motion: ***a*** *translation in Y- or Z-direction,* ***b*** *rotation of the plane by ϕ and by ϑ.*

name	$\sigma_X = \sigma_Y$	σ_Z	σ_I
N1	0.005	0.05	0.5
N2	0.01	0.1	1.0
N3	0.02	0.2	2.0

As we have seen in the previous chapter this is a sensible choice for the range sensors used in this work. In Fig. 8.4 the dependency of the relative error on four types of movements are given for the above noise models. For a translation in X-direction we find the same behaviour as above with similar error values even though there is noise on all data channels now. The accuracy in the estimation of the Z-movement does not decrease for higher velocities until $2\,mm/frame$ are reached. For even greater velocities no range flow is estimated as the confidence value drops to zero. Unfortunately this is not the case for high X-velocities which are computed wrongly.

Another interesting type of motion is a rotation of the plane. This can be achieved by changing the spherical angles of the surface normal, resulting in a non-uniform velocity field. The relative error for an increasing rotational velocity is given in Fig. 8.4 b. For small changes in the angles the resulting flow field is calculated very accurately. This remains true for rotations around the Z-axis ($\Delta\phi$) even for a rotation up to $10^\circ/frame$. However for a change in ϑ, which is the angle between surface normal and the Z-axis, we see that the accuracy quickly deteriorates for changes above $1^\circ/frame$. This implies that a fast tilt of the observed surface can not be captured accurately using this method. That is if the frame rate can not be increased accordingly.

The previously discussed rotational movements already result in a non-uniform velocity field. This is also the case when there is a non-zero expansion rate. For an oriented plane with $\vartheta = 15^\circ$ the accuracy of the computed flow is examined in Fig. 8.5 for varying expansion rates. We see that both the relative and directional error increases for higher growth rates. Yet even for very high growth rates up to $10\%/frame$ we achieve a relative error below 1% and a directional error below 1°. However, note that for the highest noise level (**N3**) no full flow is computed for growth rates exceeding $2\%/frame$. The accuracy of the computed growth rate in combination with other movements will be further discussed in Sect. 8.2.3.

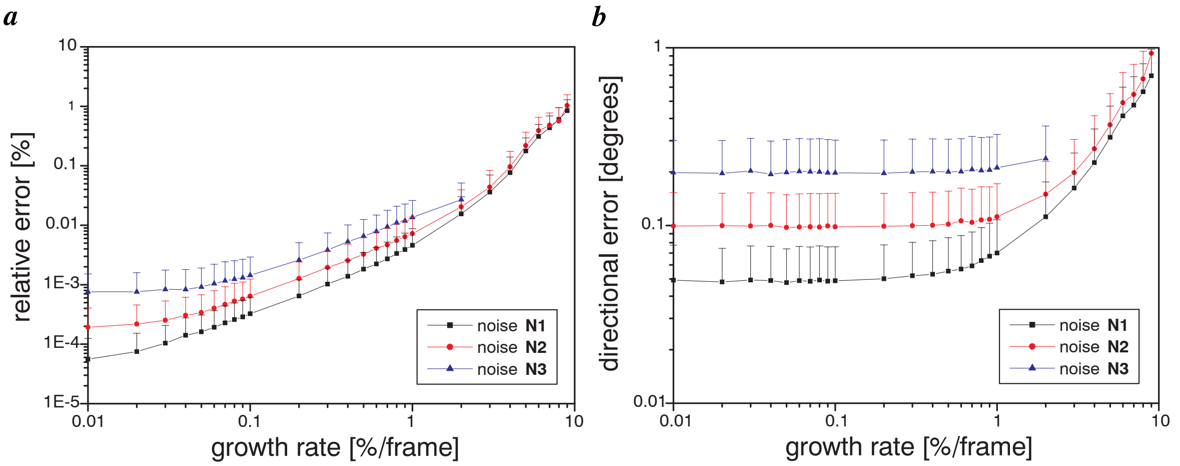


Figure 8.5: *Error in the TLS range flow depending on the expansion rate:* ***a*** *relative error and* ***b*** *directional error.*

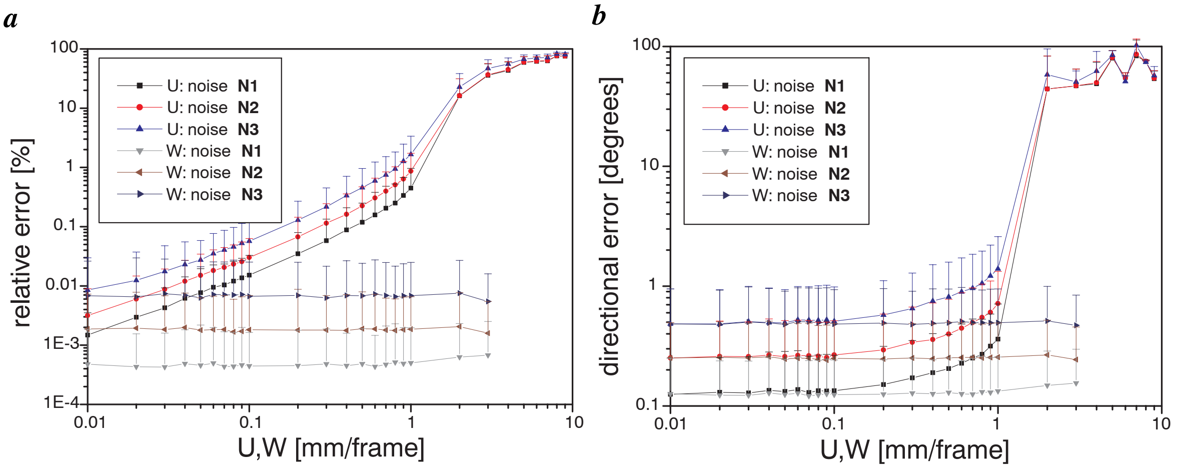


Figure 8.6: *Error in the TLS range flow on the sphere depending on the velocity in X- and Z-direction:* ***a*** *relative error and* ***b*** *directional error.*

We proceed by looking at the range flow accuracy on the data made up of a synthetic sphere, the generation of these data sets is explained in Sect. D.2. The relative and directional error for a translation in X- and Z-direction is given in Fig. 8.6. We obtain the same qualitative results as for the plane test data (Fig. 8.4 a) but on this data the error is almost one order of magnitude higher. Yet for motions smaller than $1\,mm/frame$ the error is still below 1% respectively 1°.

8.2.2 Regularised Range Flow

The results of the TLS estimation on the synthetic data is already very good indeed. Therefore we only examine the regularisation results for the example of the moving sphere to see whether there are further improvements in the accuracy. The relative magnitude error is given in Fig. 8.7 a and the directional error in Fig. 8.7 b. Comparing the results to those found using only TLS (Fig. 8.6) we see that both error values drop by about one order of magnitude. This can be attributed to the smoothing

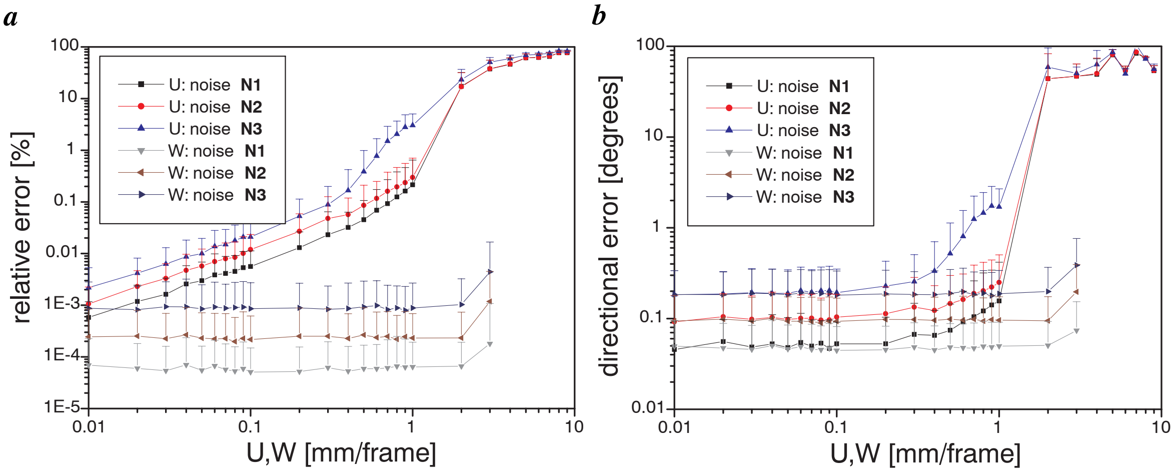


Figure 8.7: Error in the regularised range flow on the sphere depending on the velocity in X- and Z-direction: ***a*** *relative error and* ***b*** *directional error.*

accompanied by the regularisation. Here and in the following we used $\alpha = 10$ and stopped after 100 iterations, see Sect. 5.2.1.

8.2.3 Growth Estimation

An analysis of the achievable accuracy in the estimated expansion rates is presented. Due to the need of smooth velocity fields we only examine the accuracy with which the expansion rates are estimated on the regularised range flow fields. For the growth estimation both range flow and range data are required to be available on the same scale. Thus we also look at the effect of smoothing the data using the membrane model ($\alpha = 10$) described in Sect. 2.2.2. Note that the smoothing is only done for the expansion rate estimation and not for the range flow computation. This is because on real data such a regularisation step would reduce much of the needed variation in the data.

The mean computed growth rate and its standard deviation as a function of an increasing correct growth rate on the sphere data are shown in Fig. 8.8 a. On average the expansion rate is computed accurately from 0.01 $\%/frame$ up to 1 $\%/frame$, for even higher growth rates the estimation quickly fails completely. However, there is a substantial spread in the result. This implies that a lot of averaging is needed in this case to achieve reliable results. The variance in the computed growth increases with the noise level and we find a rather big difference between **N1** and **N3**. Note that the effect of the noise is already reduced by the regularisation of the data, without it we find an even higher variance in the computed expansion rate (Fig. 8.8 a).

To quantify the accuracy of the calculated growth rate we report the mean relative error. Let the correct growth be given by e_c and the estimated rate by e_e, then the *relative growth error* becomes:

$$E_g = \frac{|\,(|e_c| - |e_e|)\,|}{|e_c|} \cdot 100\ [\%]\ . \tag{8.3}$$

The resulting mean relative error in dependence of the correct expansion rate is shown in Fig. 8.8 b. First we note that the error increases with the noise level as has to be expected. There is only a minor

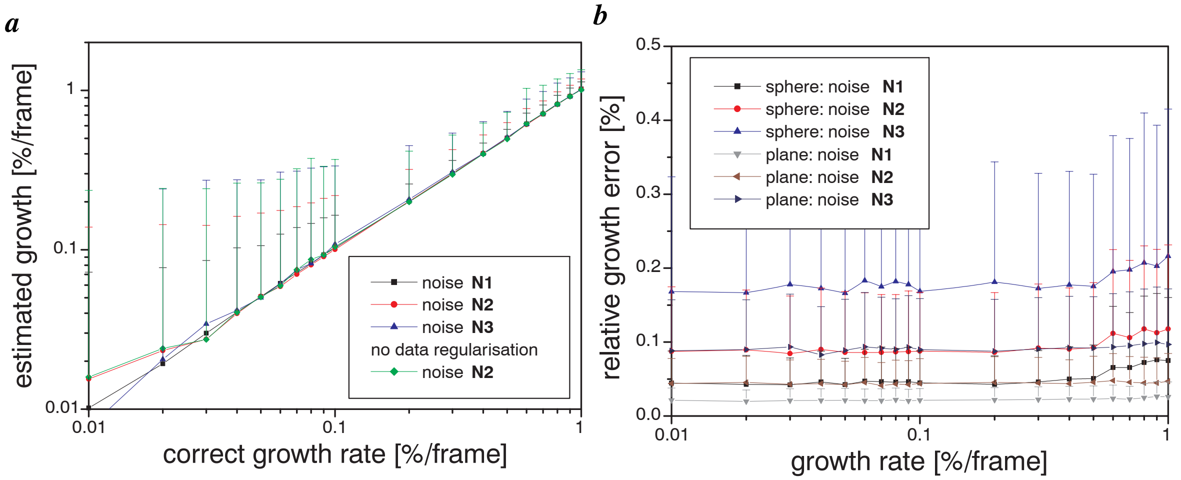


Figure 8.8: *Expansion rate estimation:* ***a*** *estimated versus correct growth rate on the expanding sphere and* ***b*** *relative error in the growth rate for increasing correct growth on the sphere and plane data.*

increase of the mean error for higher growth rates. The error is somewhat smaller on the planar data than it is on the sphere. Yet in both cases the estimated growth rate agrees very well with the correct rate.

In the following we examine the influence of additional movements on the estimated growth rate. This will be done on the realistic noise regime **N2**. Furthermore we only investigate a few realistic growth rates. Typical growth on plant leaves can be up to 5 $\%/h$ [Walter, 2001]. For a sampling rate of 3 minutes this corresponds to an expansion rate of $0.25\ \%/frame$. Hence we choose the following three example growth rates:

name	growth rate $[\%/frame]$	growth at 3 min sampling $[\%/h]$
G1	0.025	0.5
G2	0.1	2.0
G3	0.25	5.0

The error for these growth rates on the plane in dependence of the vertical angle ϑ is given in Fig. 8.9 a. There is a small increase in the error for larger angles but the error is very small even then. On the sphere we explore whether the radius and thus the curvature has an effect on the growth estimation. The relative error as a function of the radius is given in Fig. 8.9 b. No dependency on the radius can be observed. Also note that in both cases the value of the correct growth rate has no effect on the achieved accuracy, in the considered cases. Thus we will only look at a rate of $0.1\ \%/frame$ (**G2**) in the following.

Next we analyse the influence of an additional movement on the accuracy of the estimated growth rate. The error in the growth depending on an extra translation in the X- or Z-direction is shown in Fig. 8.10 a for the plane at $\vartheta = 5°$ and the sphere with radius $r = 300\,mm$. For both data sets we find better results for a motion along the Z-axis (W). This is obviously caused by the more accurate velocity estimation for such movements, see Fig. 8.7. In the same way as the velocity estimation fails for movements above $1\,mm/frame$ in X-direction does the growth estimation become unreliable. Unfortunately we sometimes encounter leaf movements that are greater when using a sampling rate

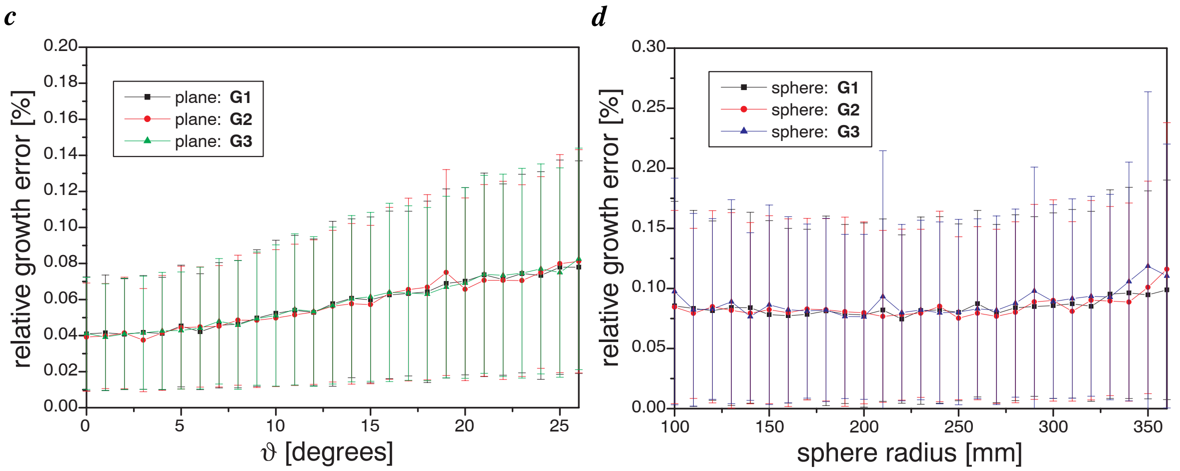


Figure 8.9: *Error in the estimated growth rate:* ***a*** *depending on the orientation of the plane ϑ and* ***b*** *on the initial radius of the expanding sphere.*

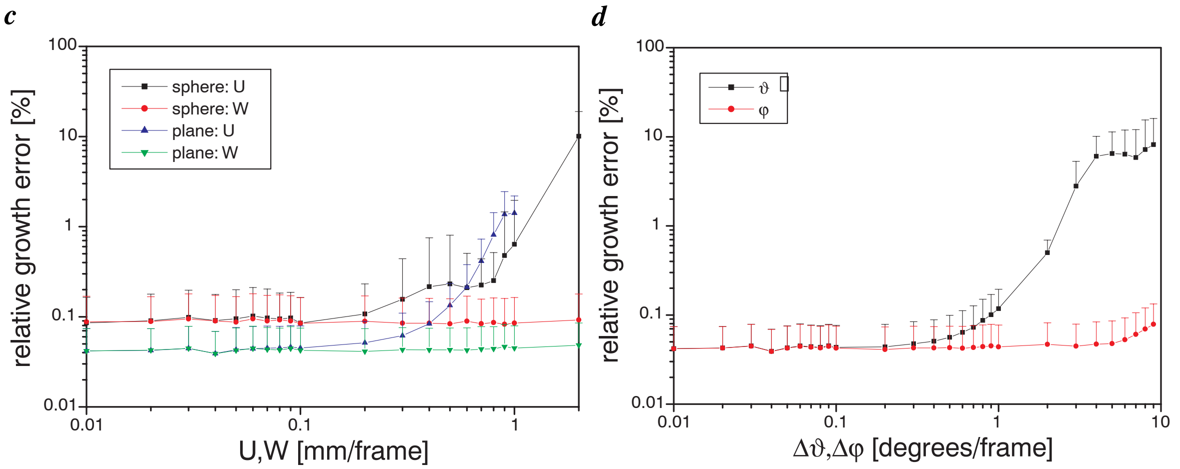


Figure 8.10: *Error in the estimated growth rate of* $0.1\ \%/frame$ *depending on additional movement:* ***a*** *translation in X- or Z-direction and* ***b*** *rotation (ϕ) and tilt (ϑ) of the plane.*

of 3 minutes per frame, which indicates the need for higher sampling rates. Especially at the beginning of the night when the outer part of the lobes are bending downwards.

On the synthetic plane data we can investigate whether the determined growth rate is influenced by a rotational movement. Figure 8.10 b gives the error in the computed growth rate when there is an additional change in either ϑ or ϕ. A rotation around the Z-axis, i.e. a change in ϕ, has no effect for rotations of up to $10°/frame$. This is not the case for a rotation increasing the angle ϑ between Z-axis and surface normal. Here we find that a change by more than $1°/frame$ leads to a large error in the growth rate. The same has already been observed for the error in the recovered velocity, see Fig. 8.4 b. Again this sometimes causes problems when observing leaves which often show a significant amount of tilting at the rim.

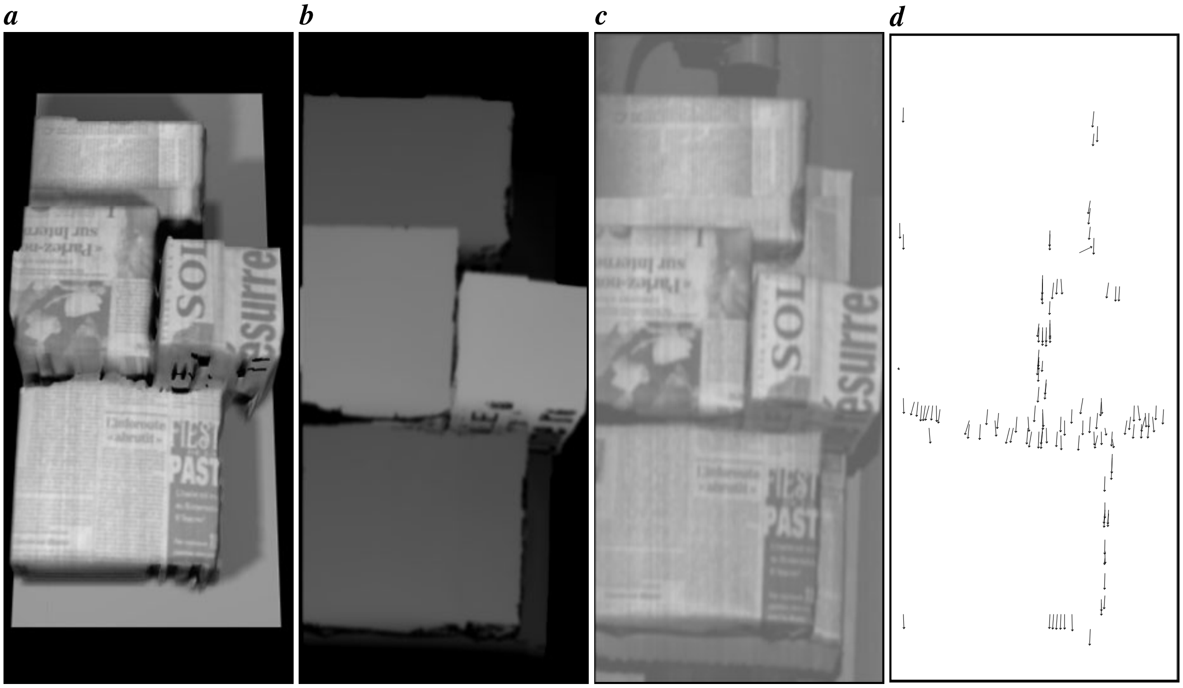


Figure 8.11: NRC newspaper sequence: a texture–mapped, b depth, c intensity, d XY-full flow using only 3D data.

8.3 Results on Real Data

As the results from synthetic data can only be transferred to a certain extent to real data we examine a few real sequences. In all cases the observed objects are moved using linear positioners. Thus we can only generate pure translational movements without any deformations. The thus achievable accuracy is analysed both for Biris and structured light range data, compare Chap. 7.

8.3.1 Biris Range Data

The range flow performance on Biris laser range data is first analysed in a bit more detail on the so called NRC newspaper sequence. Then we present some results on sequences taken with the sensor at the University of Western Ontario.

NRC Newspaper Sequence.

Figure 8.11 shows an example real range sequence taken with a Biris sensor at NRC in Ottawa, Canada. The objects in the scene are boxes covered with newspaper. Between frames the complete scene is moved by $[0.095\ 1.425\ 0.082]^T\ mm$. As there are mainly planar and linear regions we can get only 1% full range flow vectors when only the 3D data is used, Fig. 8.11 d. The faces of the boxes are turned upwards, therefore the plane flow points in the Z-direction. As a result of the motion being practically confined to the X,Y–plane the resulting plane flow is close to zero. At horizontal edges the line flow gives the correct vertical movement. Depending on the weight of the intensity data we find the following results for full range flow:

method	β^2	density [%]	E_r[%]	E_d[°]
only Z	0.0	1.5	8.7 ± 13.9	12.4 ± 17.0
I+Z	0.5	17.0	10.3 ± 11.2	9.6 ± 8.7
I+Z	1.0	30.0	12.1 ± 13.0	10.5 ± 11.9
I+Z	10.0	41.9	12.0 ± 13.2	9.7 ± 11.3

With increasing influence of the intensity we see a great increase in density without too much of a performance penalty. However due to the relatively noisy data the TLS results are not very accurate. This can be improved upon by regularisation where we obtain different results depending on whether we use the flow obtained from using both intensity and depth (I+Z) with $\beta = 1$ or just the depth (Z) data ($\alpha = 10$):

method	iterations	E_r[%]	E_d[°]
Z	0	81.1 ± 34.9	86.0 ± 47.9
Z	10	78.7 ± 27.4	39.3 ± 35.4
Z	100	61.4 ± 32.1	13.7 ± 17.7
Z	1000	19.9 ± 18.7	2.4 ± 2.8
Z	5000	2.8 ± 3.6	2.1 ± 2.7
I+Z	0	54.1 ± 43.9	83.5 ± 83.4
I+Z	10	44.7 ± 32.3	3.2 ± 3.9
I+Z	100	27.1 ± 23.6	1.8 ± 2.4
I+Z	1000	4.0 ± 4.2	1.4 ± 2.2
I+Z	5000	4.5 ± 4.2	1.4 ± 2.2

Clearly the estimation based on the depth alone leaves large areas with only plane normal flow. Thus it takes many iterations for the full flow to spread into these areas. Starting off with the TLS result obtained using both intensity and depth we reach a good solution considerably faster. However as was previously observed using the intensity as well does not give quite as accurate results in the very end. As a matter of fact after 1000 iterations the result does not improve any more, it even deteriorates slightly.

UWO Sequences.

In order to get additional real test data we placed some test objects on a set of linear positioners underneath the Biris scanner at the University of Western Ontario (UWO), see Fig. 8.12. The positioners allow for translations along all three axes. Because the measurement coordinate system is attached to the sensor we can not guarantee that the axes of the positioning system do align exactly with the coordinate axes. Even though the positioners have been placed carefully there might still be some misalignment.

Using the objects shown in Fig. 8.12 b,c,d we performed a number of controlled movements. We choose crumpled newspaper as it provides data with a rich depth structure and also shows significant intensity variation. The leaves are obviously chosen as it is such data that we will be concerned with in the main application considered here. As the leaves are placed on a flat surface we segment them prior to any computation.

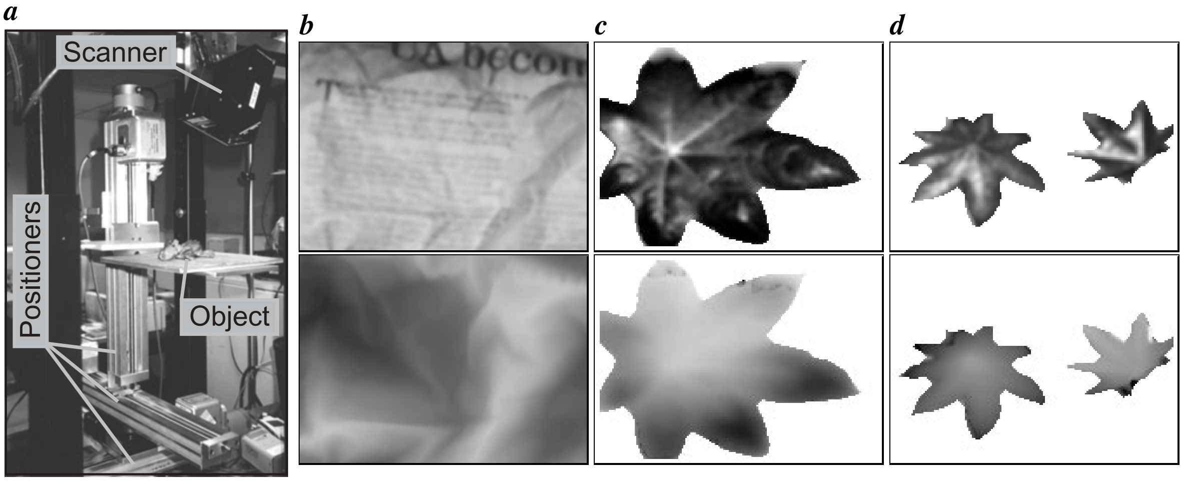


Figure 8.12: *Real test data.* ***a*** *Biris scanner and positioners. Test data, intensity (upper row) and depth (lower row):* ***b*** *a sheet of crumpled newspaper,* ***c*** *a cut leaf and* ***d*** *two younger leaves.*

The resulting relative E_r and directional E_d errors are shown in Fig. 8.13 as a function of the absolute value of the correct velocity. To maintain readability we omit the error bars in this case, the standard deviation is in the same order of magnitude than the error value itself. For small velocities, below $0.15\,mm/frame$, the relative error is consistently below 1%. In this velocity range we find that the regularisation does improve the result, in particular on the newspaper the improvement is rather big. For the initial TLS solution best results are obtained on the big leaf, followed by the small leaves.

The directional error (Fig. 8.13 b) does not increase as much with increasing speed but is relatively high on average. However this can be somewhat attributed to the slight misalignment of positioner and sensor coordinate frame. We have also used some other objects to test the range flow algorithm. The intensity and depth data of a few examples are shown in Fig. 8.14. On these data sets we obtain the following errors where the TLS results are shown on the left of the regularised results:

object	correct flow $[mm]$	E_r [%]	E_d [°]	density [%]	E_r [%]	E_d [°]
a	$[0.10\;0.26\;0.22]^T$	2.7 ± 5.5	6.6 ± 4.9	30.4	1.2 ± 0.7	5.3 ± 4.2
b	$[0.13\;0.51\;0.25]^T$	2.2 ± 2.2	4.1 ± 2.9	35.8	1.2 ± 0.8	2.1 ± 1.1 8
c	$[0.45\;\;-0.25\;0.38]^T$	5.4 ± 7.2	12.0 ± 9.7	28.8	3.8 ± 2.7	10.4 ± 7.2
d	$[0.16\;0.19\;\;-0.32]^T$	0.7 ± 0.9	2.5 ± 2.3	28.8	0.3 ± 0.4	1.1 ± 1.3

Clearly the regularisation not only yields 100% dense full flow but also significantly enhances the results of the initial TLS algorithm. In order to study the 3D velocity the achieved accuracy seems to be good enough. The main factor that limits the quality of the computed range flow fields is the need for resampling (Sect. 7.1.2) which should clearly be avoided in future experiments. In this work additional experiments have been carried out using the structured light sensor where the data is already registered. The range flow performance on the thus obtained data sets is examined next.

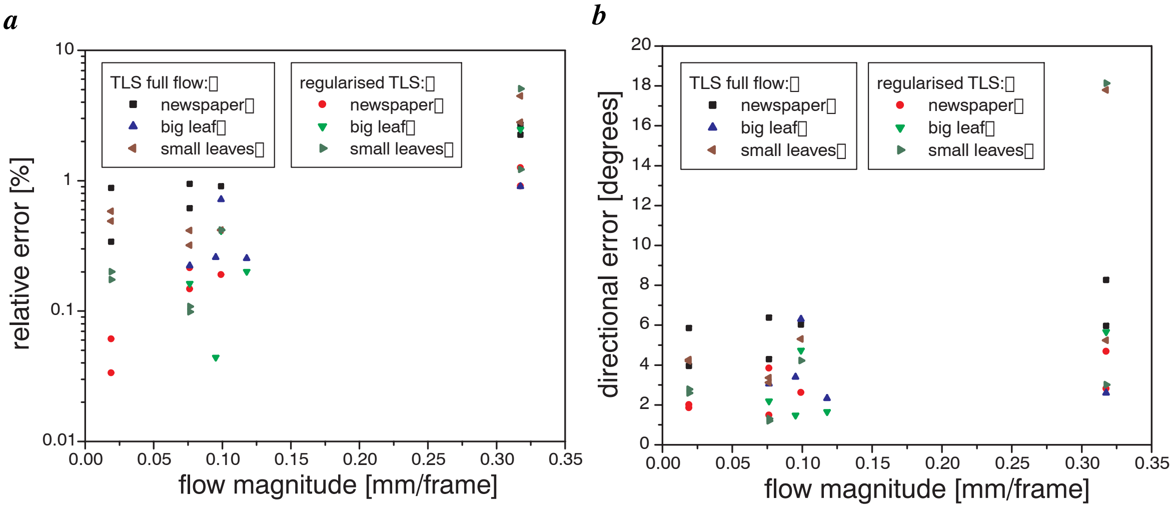


Figure 8.13: *Results on the real Biris data as a function of the velocity magnitude:* ***a*** *relative error and* ***b*** *directional error.*

8.3.2 Structured Light Data

We will now look at the achievable range flow accuracy using the structured light sensor assembled at the Institute for Botany at the University of Heidelberg, see Sect. 7.2. Again some test objects are placed on a set of linear positioning tables and moved by a known displacement between consecutive scans. As the same positioners are used to calibrate the sensor, the positioner coordinate frame and the sensor coordinate system are identical in this case. First we examine a sheet of crumpled paper and a freshly cut leaf in more detail. Example intensity and depth data from these two test scenes are given in Fig. 8.15.

As it makes a difference whether the motion is in horizontal or vertical direction we show the relative and directional errors depending on U,V and W separately in Fig. 8.16. We did not observe any difference between movements in X or Y or an influence of the sign. Therefore we plot the results as function of the absolute value of the horizontal or vertical motion. Those movements containing both horizontal and vertical motion are plotted among the horizontal velocities, because we found this component to be clearly dominating the error.

For the relative error depending on U or V (Fig. 8.16 a) we find the previously determined rise with increasing magnitude. Again the regularisation does significantly improve the accuracy for small velocities. On these two test objects we get better results for the crumpled paper than for the leaf. In order to maintain a relative error below 1% the speed should not exceed 0.5 $mm/frame$ in the horizontal direction. Please note that the full flow density on the crumpled paper data is not very high and quickly decreases with increasing velocity. The directional error depending on the horizontal movement (Fig. 8.16 b) shows a similar rise towards higher velocities. To ensure an error below 5° in the estimated velocity direction the speed should not exceed $0.5\,mm/frame$.

For the error in dependence of the vertical velocity we observe only a minor increase for velocities up to 3 $mm/frame$. This is in qualitative agreement with the results found on synthetic data. However the errors on real data are significantly higher. Again we see that the regularisation improves

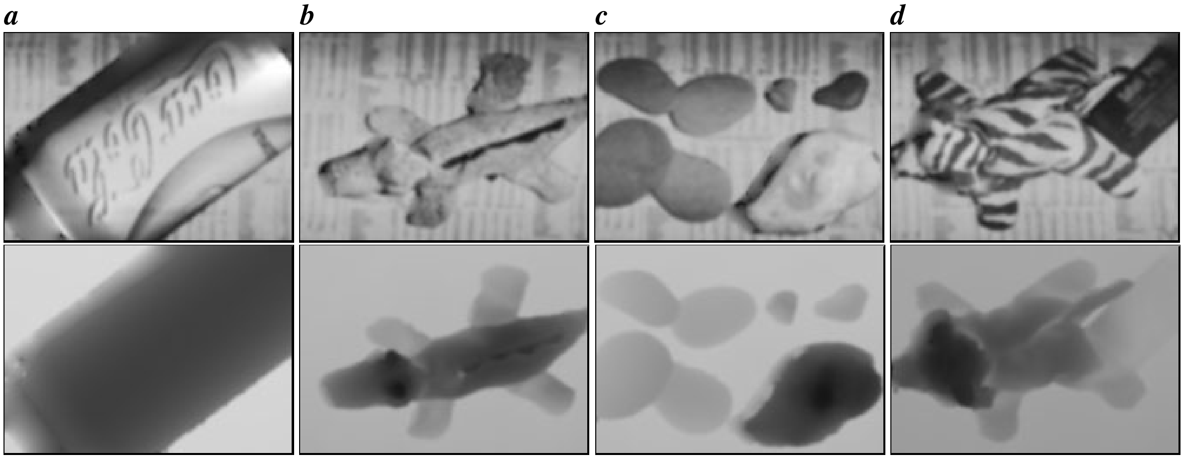


Figure 8.14: Real test data using the Biris scanner. Intensity (upper row) and depth data (lower row): ***a*** *a coke can,* ***b*** *a toy crocodile* ***c*** *some pebbles and* ***d*** *a toy tiger.*

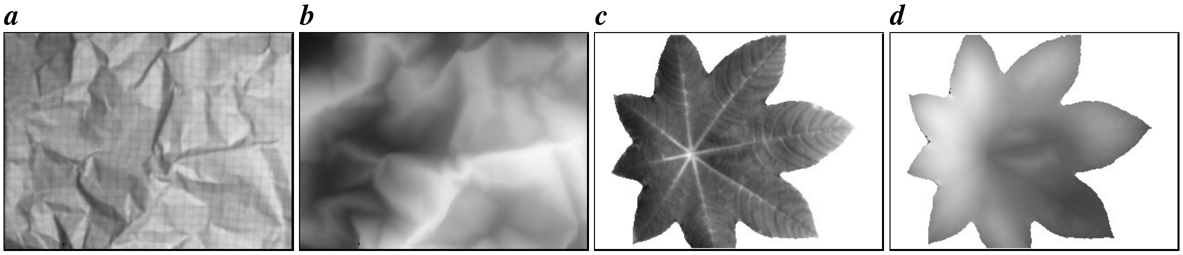


Figure 8.15: Real structured light data to asses the range flow performance. Crumpled paper: ***a*** *intensity and* ***b*** *depth. Freshly cut leaf:* ***c*** *intensity and* ***d*** *depth.*

the results and that we get better results on the crumpled paper than on the leaf. The relative error remains on average well below 1% for the complete velocity range. Except for one outlier we also find the directional error on the regularised flow to be less than 5° for all considered movements.

To investigate if the presented range flow estimation framework also works on other objects we test the performance for a range of different scenes, see Fig. 8.17. The results for some example movements are given below. We also use the toy tiger already shown in Fig. 8.14 d, this data will be denoted with e in the following table:

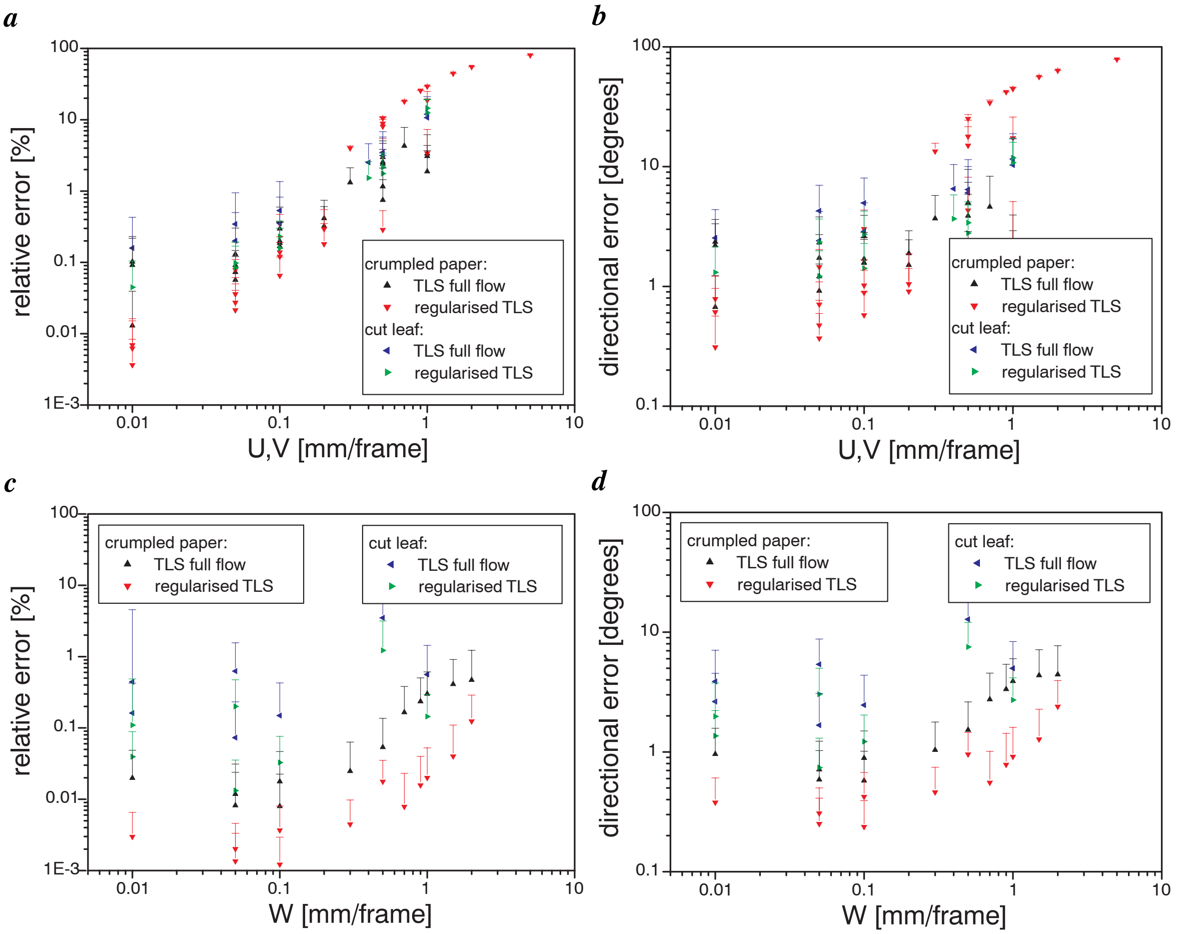


Figure 8.16: *Results on the real data for different movements:* ***a*** *relative error and* ***b*** *directional error depending in the motion in X- or Y-direction.* ***c*** *relative error and* ***d*** *directional error depending in the motion in Z-direction.*

object	correct flow $[mm]$	E_r [%]	E_d [°]	density [%]	E_r [%]	E_d [°]
a	$[0.0\ 0.3\ 0.4]^T$	1.80 ± 2.08	6.18 ± 4.65	28.2	1.4 ± 0.9	3.5 ± 2.2
a	$[0.0\ \ -0.1\ \ -0.5]^T$	0.44 ± 0.80	4.09 ± 3.31	53.8	1.7 ± 0.1	1.6 ± 1.3
b	$[0.0\ 0.5\ 1.5]^T$	3.75 ± 6.32	8.32 ± 10.81	7.6	4.2 ± 4.0	7.9 ± 8.0
b	$[0.0\ \ -0.3\ \ -0.6]^T$	3.03 ± 5.42	9.91 ± 10.16	3.4	2.7 ± 1.5	9.4 ± 6.4
b	$[0.0\ \ -0.05\ \ -0.05]^T$	0.29 ± 0.59	3.39 ± 3.10	45.0	0.05 ± 0.06	1.0 ± 0.8
b	$[0.0\ 0.0\ 0.3]^T$	0.48 ± 1.00	3.93 ± 3.95	25.6	0.02 ± 0.04	0.9 ± 0.7
c	$[0.0\ 0.5\ 1.2]^T$	5.53 ± 7.41	17.77 ± 17.68	4.1	8.0 ± 2.2	20.4 ± 11.1
c	$[0.0\ 0.0\ \ -0.2]^T$	0.26 ± 0.41	3.28 ± 2.44	73.9	0.11 ± 0.13	2.20 ± 1.42
c	$[0.0\ \ -0.2\ 0.0]^T$	1.07 ± 0.78	5.80 ± 3.68	40.0	1.23 ± 0.43	5.09 ± 2.19
d	$[0.0\ 0.2\ 0.2]^T$	0.41 ± 0.31	1.94 ± 1.12	32.3	0.25 ± 0.15	0.99 ± 0.46
d	$[0.0\ \ -0.3\ 0.4]^T$	0.40 ± 0.35	1.29 ± 0.93	62.5	0.22 ± 0.17	0.64 ± 0.37
d	$[0.0\ 0.0\ \ -0.2]^T$	0.20 ± 0.17	3.36 ± 1.38	14.0	0.05 ± 0.05	1.58 ± 0.91
d	$[0.0\ \ -0.2\ 0.0]^T$	0.25 ± 0.21	1.16 ± 0.88	63.0	0.16 ± 0.12	0.60 ± 0.41
e	$[0.0\ \ -0.1\ 0.1]^T$	0.25 ± 0.32	2.79 ± 2.10	70.4	0.16 ± 0.12	1.46 ± 0.99
e	$[0.0\ \ -0.2\ 0.5]^T$	0.59 ± 0.73	2.47 ± 2.47	7.1	0.75 ± 0.42	3.32 ± 2.13
e	$[0.0\ 0.3\ \ -0.2]^T$	1.69 ± 1.30	5.42 ± 4.57	19.4	2.01 ± 1.08	5.15 ± 3.10
e	$[0.0\ 0.2\ 0.0]^T$	0.65 ± 0.85	3.08 ± 3.66	3.7	1.38 ± 0.45	7.65 ± 4.00

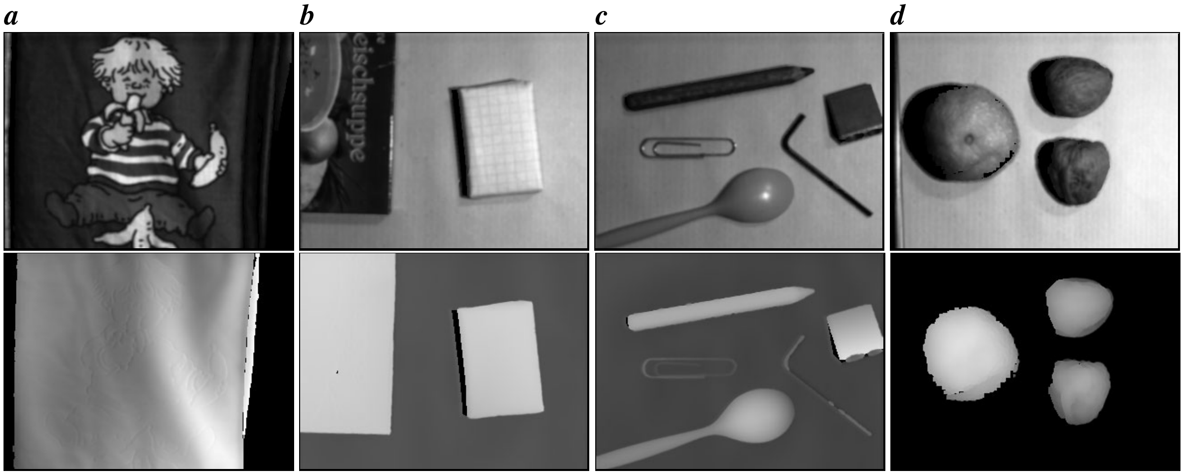


Figure 8.17: Real test objects using the structured light sensor. Intensity (upper row) and depth data (lower row): ***a*** *a printed cloth,* ***b*** *some boxes,* ***c*** *pen, spoon and other objects and* ***d*** *nuts and an orange .*

We find the presented range flow algorithm to be able to deal with a broad range of scenes. Overall the results indicate the same range flow accuracy on these data sets than on the previously considered cases. For motions less than roughly 0.5 $mm/frame$ we achieve very good range flow results. Usually the regularised flow is more accurate than the TLS full flow. Only in some cases where the initial full flow density is quite low we find higher error values after the regularisation. However one should bear in mind that the regularised flow is always at 100 $\%$ density.

8.4 Summary

The achievable accuracy of the estimated range flow has been analysed in terms of relative magnitude and directional errors. Towards this end we used synthetic data sets constructed using a model of a range sensor employing a pinhole camera, such as our structured light system. From the performance analysis using such synthetic data we conclude that movements up to roughly $1 mm/frame$ can be estimated with a relative error below 1%. Rotations around the Z-axis can be tolerated up to $10°/frame$. But changes in the angle between surface normal and Z-axis have to remain below $1°/frame$ to allow a reliable range flow computation. It has been shown that the error in the estimated growth rates is strongly correlated with the error in the range flow. For the given limits the expansion rate can be recovered with a relative error well below 1%.

However, from findings on real data sets obtained by moving several test objects using linear positioners we conclude that the maximal velocity where errors remain below 1% and $5°$ is slightly lower. For the Biris sensor a limit of roughly 0.15 $mm/frame$ has been determined. On real structured light data we find that on castor bean leaves the errors are acceptable for velocities smaller than 0.5 $mm/frame$. For a sampling rate of 3 minutes this corresponds to 10 mm/h.

Chapter 9

Leaf Motion and Growth

This chapter applies the previously introduced range flow framework to study the dynamic behaviour of castor bean leaves. For this application a wealth of information can be extracted from range data and range flow. Here some of the possible ways to look at the data are presented and illustrated on a few real world examples. This chapter is not meant as a thorough botanical analysis, but rather as an exemplary proof that useful botanical results can be obtained.

9.1 Example Leaf Data and Movement

Range flow fields from living castor bean leaves show rather different types of movement under different conditions (Fig. 9.1). Leaves that are confined in a plane by pulling at the lobe tips, see Fig. 7.4 c, are relatively flat but not entirely so (Fig. 9.1 a,b). The movement on such a fixed leaf is predominantly in the horizontal plane, but there is some vertical motion as well. In the given example there are locations where the leaf is moving upwards and others where it moves downwards.

An unconstrained leaf just after nightfall already shows some bending of the lobes (Fig. 9.1 c,d). This bending expresses itself in a large downward velocity. Another example of a free leaf taken during the night shows an overall upward movement paired with a fair bit of lateral motion (Fig. 9.1 e). At day time, on the other hand, much smaller velocities are encountered (Fig. 9.1 f).

9.2 Leaf Surface Parameters

Two main parameters are of interest when analysing this kind of data. First the orientation of the leaf varies greatly during the day, see Fig. 1.1 and Fig. 9.1. The orientation can be described by the angle between the surface normal and the Z-direction. The distribution of this angle also captures the bending of the surface. Second the surface curvature is of interest. In the following it is necessary to recall a few elementary concepts of differential geometry. The needed formulas will be given without proof and the interested reader is referred to the literature for more details [do Carmo, 1976; Dineen, 1994; Casey, 1996; Kühnel, 1999].

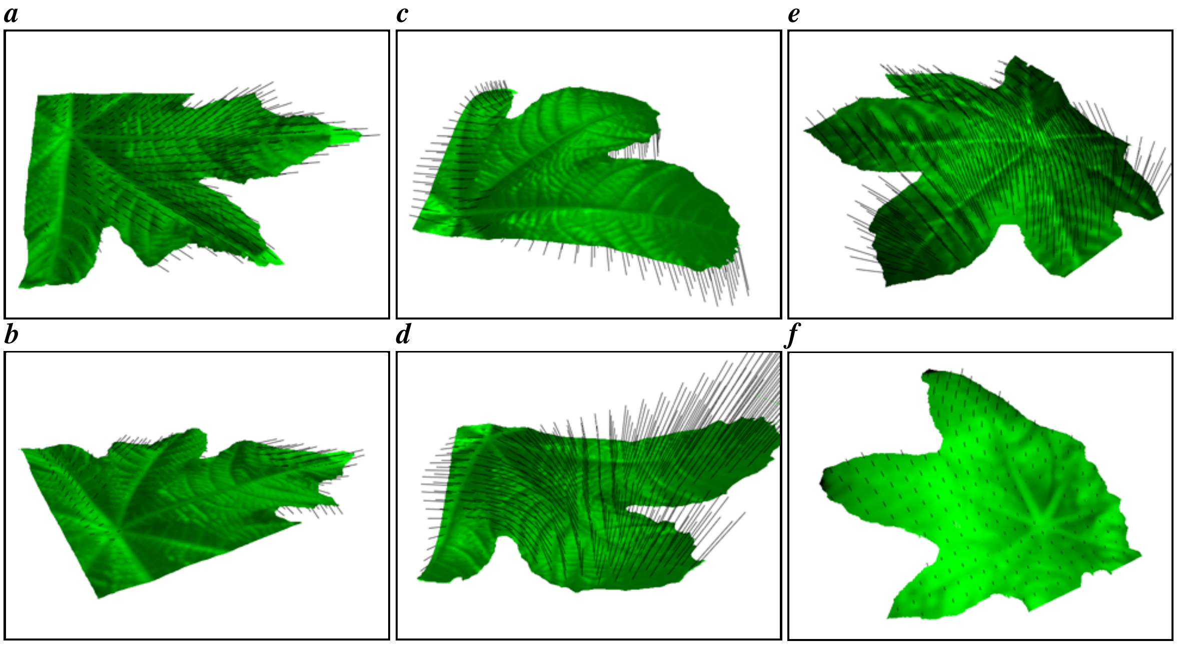


__Figure 9.1:__ Example data and movements of castor bean leaves, __a__ ,__b__ ,__c__ ,__d__ are taken with the structured light sensor at 3 minute sampling and __d__ ,__e__ with the Biris sensor with a 5 minute sampling interval: __a__ fixed leaf viewed from above and __b__ the same leaf from underneath. __c__ An unconstrained leaf with rapidly folding lobes and __d__ the same leaf viewed from underneath. __e__ Unconstrained leaf during the night and __f__ one at daytime.

9.2.1 Vertical Angle

For a surface parameterised as $\boldsymbol{s}(x,y) = [X(x,y)\,Y(x,y)\,Z(x,y)]^T$ the surface normal is given by:

$$\boldsymbol{n} = \frac{\partial_x \boldsymbol{s} \times \partial_y \boldsymbol{s}}{|\partial_x \boldsymbol{s} \times \partial_y \boldsymbol{s}|}\,. \tag{9.1}$$

We choose the angle between surface normal and the vertical direction to describe the orientation of a leaf. This is done firstly because gravitation defines this axis and secondly because in our experiments illumination is always from above. Note that we always orient our sensor accordingly. If the illumination came from a different direction we might need to consider the angle relative to this second orientation as well. At each surface point the orientation angle can be computed as:

$$\gamma = \arccos([0\,0\,1]^T\boldsymbol{n})\,. \tag{9.2}$$

Small angles are observed along the veins for the attached leaf (Fig. 9.2a). This is clearly due to the pulling force that acts along these directions. The other regions have an inclination angle that can be up to 30°. This tilt of the surface introduces systematic errors in the 2D growth analysis, see Sect. 6.1.2. Yet for freely moving leaves (Fig. 9.2b,c) even greater vertical angles occur frequently.

In order to examine the dynamic changes of the leaf orientation the mean vertical angle averaged over the entire observed region is given in Fig. 9.3. Note that due to the motion the size of the measured region can change over time, hence the average is not always computed on exactly the same part of the leaf. The drop in the angle seen at the onset of the night for the free leaf (Fig. 9.3 b) is

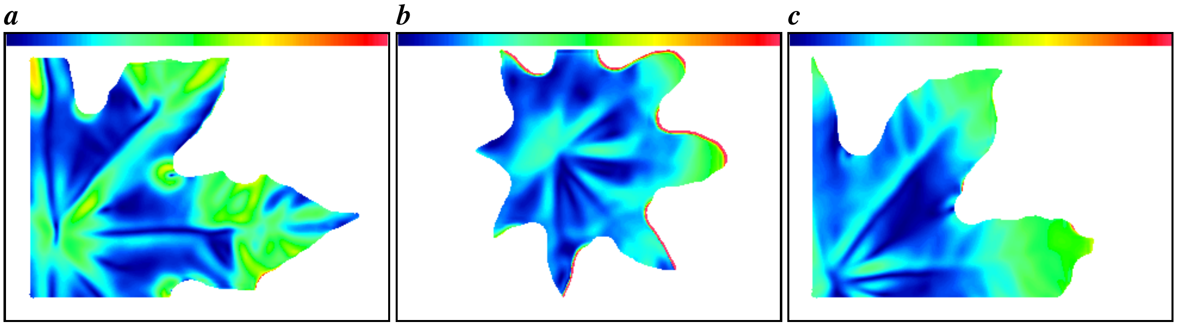


***Figure 9.2:** Vertical angle: **a** fixed leaf* $[0°, 45°]$, ***b** free leaf at day time* $[0°, 90°]$ *and **c** free leaf at the beginning of the night* $[0°, 90°]$.

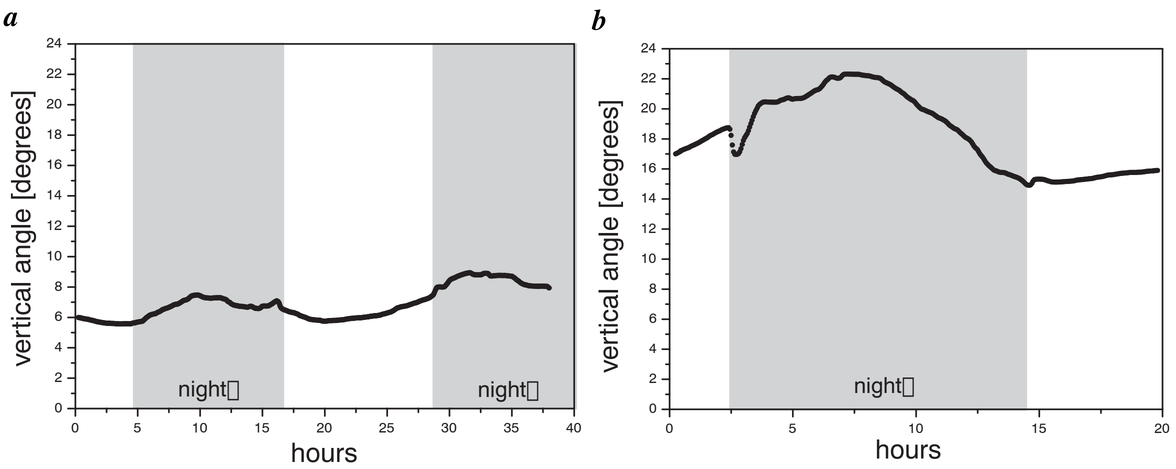


***Figure 9.3:** Mean vertical angle over time: **a** fixed leaf, **b** unfixed leaf.*

a result of this effect. Because the lobes are not visible any more the average can only be computed for the middle part of the leaf. As expected the mean inclination angle is significantly higher for the unfixed leaf. Both free and attached leaf show a clear diurnal rhythm, with maximum angles reached during the night.

9.2.2 Curvature

The mapping between the surface and its normal (9.1) is also called Gauss map. For an orientable surface we can define the differential of the Gauss map $\mathrm{d}\boldsymbol{n}$ on the tangent space. This important linear mapping is called shape operator or Weingarten mapping [Kühnel, 1999]. The shape operator describes how the surface normal changes along the surface and thus characterises the curvature. Its eigenvalues give the two principal curvatures κ_1 and κ_2 and the corresponding eigenvectors the principal curvature directions. The Gaussian curvature is given by the determinant of the shape operator:

$$K = \kappa_1 \, \kappa_2 \, , \tag{9.3}$$

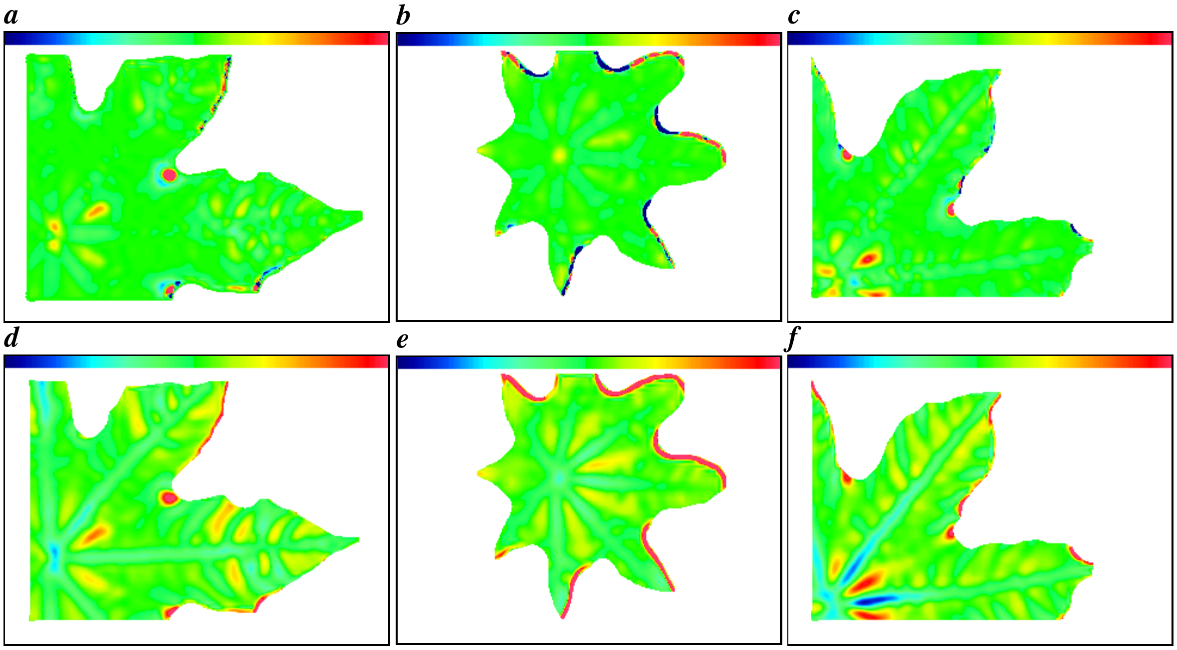


Figure 9.4: *Curvatures: Gaussian curvature* $[-0.02, 0.02]\ 1/mm^2$ ***a*** *fixed leaf,* ***b*** *unfixed leaf at daytime and* ***c*** *unfixed leaf early in the night. Mean curvature* $[-0.2, 0.2]\ 1/mm$ ***d*** *fixed,* ***e*** *unfixed (day) and* ***f*** *unfixed (night).*

and the mean curvature by the trace:

$$H = \frac{1}{2}(\kappa_1 + \kappa_2)\ . \tag{9.4}$$

It is the remarkable statement of Gauss famous *theorema egregium* that the Gaussian curvature is a measure of the intrinsic geometry of the surface. Its value can be determined from surface measurements alone. This is not the case for the mean curvature or the principal curvatures which are extrinsic quantities - they can be changed by bending the surface. Thus Gaussian curvature measures the deviation from a possible Euclidean description of a surface. If it vanishes the surface can be developed into a plane. For example the surface of a cylinder ($K = 0$) can be ruled onto a plane. A sphere ($K = \frac{1}{r^2}$) on the other hand can not even locally be transformed into a plane.

While the mean curvature describes the bending of the surface in the embedding three-dimensional space, Gaussian curvature describes it's ”intrinsic bending”. In Fig. 9.4 examples for the Gaussian and mean curvature found on leaves are shown. The Gaussian curvature is different from zero only in the area around the leaf base. In the constrained leaf (Fig. 9.4a) there are also some regions towards the tip of the lobes with some intrinsic curvature. The mean curvature (Fig. 9.4d,e,f) on the other hand is dominated by the bending due to the veins.

To analyse the overall leaf structure rather than the bending of the vein we examine the data at a coarser scale to compute average curvature values over time. Towards this end the smoothing strength in the surface membrane model is set to $\alpha = 100$, see Sect. 2.2.2. The resulting averaged values are plotted in Fig. 9.5. First we note that the values found for the fixed leaf are very small indeed. However, a change in the sign of the mean curvature can be seen. Even though the effect is small this shows that on average the leaf bends outwards towards the end of the first night. The unfixed leaf

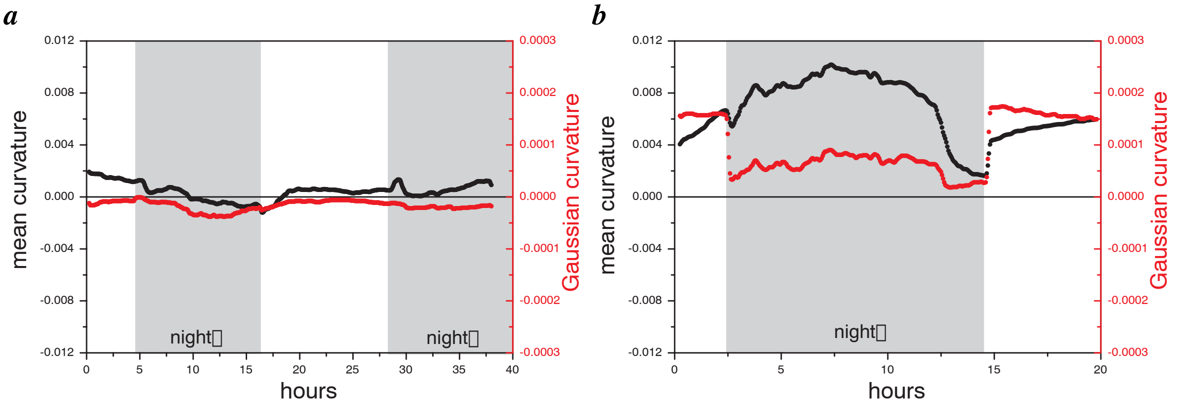


Figure 9.5: *Average curvature values over time:* ***a*** *fixed leaf,* ***b*** *unfixed leaf.*

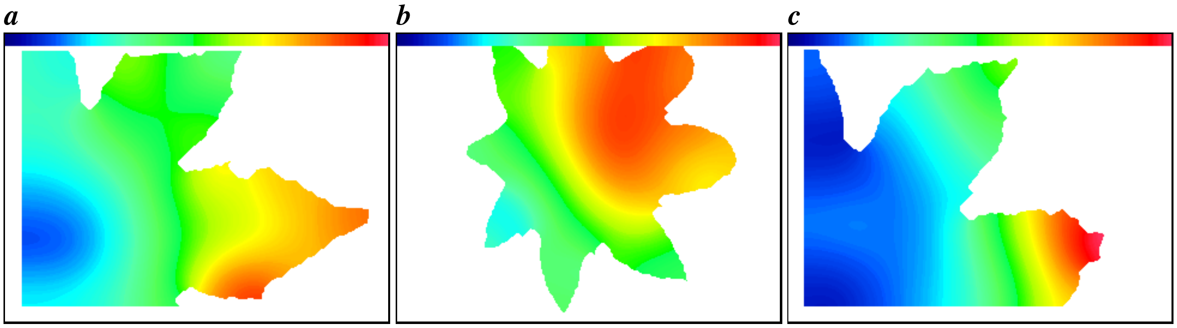


Figure 9.6: *Absolute velocity magnitude:* ***a*** *fixed leaf* $[0, 1]mm/h$*,* ***b*** *unfixed leaf at day time* $[0, 7]mm/h$ *and* ***c*** *unfixed leaf early in the night* $[0, 50]mm/h$*.*

(Fig. 9.5) clearly exhibits a rise in the mean curvature value during the night. This can be attributed to the bending of the lobes, interestingly this effect already decreases before the day sets in. This decrease coincides with the visual observation that the lobes are straightening again. There is a lower Gaussian curvature during the night, probably this can be attributed to the reduced observed area. Again the Gaussian curvature is very low.

9.3 Leaf Movements

The computed range flow field contains the movements along the sensor coordinate axes. This is obviously not a natural frame to describe the leaf movement. In the following a few other measures are extracted that can be used to illustrate the motion patterns of leaves.

9.3.1 Absolute Velocity Magnitude

The amount of motion present at a given time can be described by the absolute value of the velocity magnitude:

$$m = \sqrt{U^2 + V^2 + W^2} \,. \tag{9.5}$$

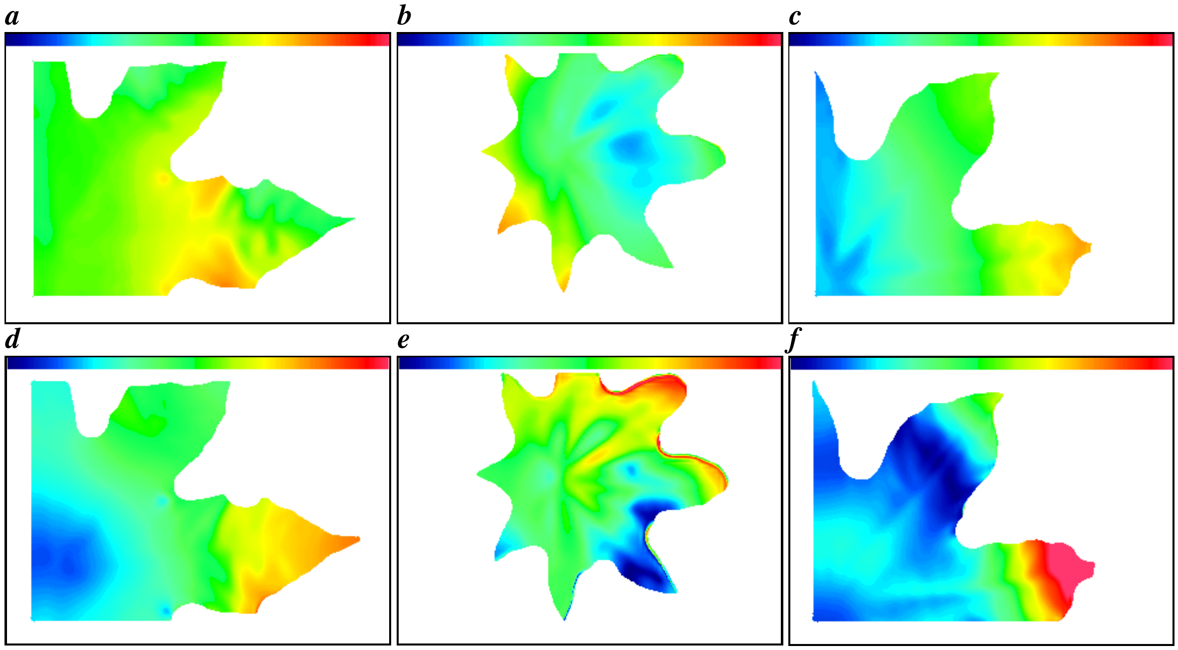


Figure 9.7: *Normal velocity magnitude **a** fixed leaf $[-1,1]mm/h$, **b** unfixed leaf at daytime $[-8,4]mm/h$ and **c** early in the night $[-10,40]mm/h$. Tangential velocity magnitude **d** fixed leaf $[0,1]mm/h$, **e** unfixed leaf at daytime $[0,6]mm/h$ and **f** early in the night $[0,30]mm/h$.*

This value can be computed for each surface point. Such speed maps can be used to determine which parts of the leaf move more or less than others. Example maps are given in Fig. 9.6, where we can observe the general trend that the tip regions move significantly more than the inner regions. Also when looking at the different ranges the fixed leaf (Fig. 9.6a) exhibits far less movement than the free leaves. Additionally the movement of the free leaf is a lot smaller during the day (Fig. 9.6b) than that observed when the lobes are bending (Fig. 9.6c). In order to examine the diurnal rhythm of leaf movement the mean absolute velocity magnitude over time is calculated (Fig. 9.8).

9.3.2 Normal and Tangential Velocity

A natural way to look at the leaf motion is to separate it into components normal and tangential to the surface. Clearly any normal velocity component can only move and bend the surface but not lead to growth. The velocity perpendicular to the surface is easily computed as:

$$\boldsymbol{f}_n = (\boldsymbol{f}^T \boldsymbol{n})\, \boldsymbol{n} \,. \tag{9.6}$$

For the tangential component we then find:

$$\boldsymbol{f}_t = \boldsymbol{f} - \boldsymbol{f}_n \,. \tag{9.7}$$

The magnitude of the normal velocity becomes positive if the motion is in direction of the surface normal and negative otherwise. For the tangential velocity only the absolute value of the magnitude is reported. These two values are shown in Fig. 9.7. Obviously there is some normal movement even in

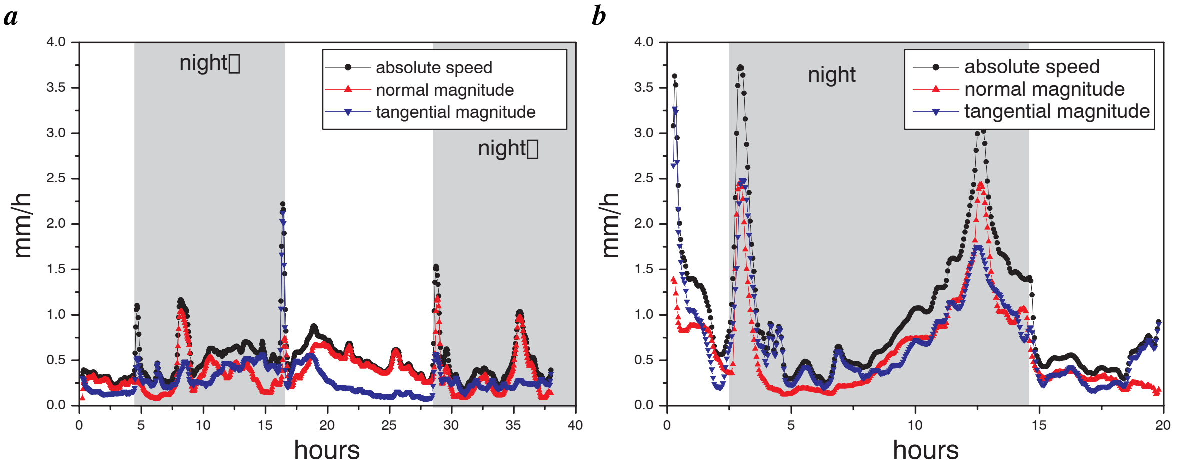


Figure 9.8: *Averaged velocity magnitudes over time:* ***a*** *fixed leaf,* ***b*** *unconstrained leaf.*

the fixed leaf (Fig. 9.7 a), which causes problems for a 2D analysis. However the tangential velocity shows the expected increase towards the tip. For the free leaf at day time we observe that while there still is some normal motion the tangential motion is dominant (Fig. 9.7 b). The unconstrained leaf shows both a large normal and tangential movement in the outer regions at the beginning of the night (Fig. 9.7 c). Also note that this leaf moves far more than the other two examples.

Taking the average value of these entities results in curves as shown in Fig. 9.8. For the fixed leaf a slight increase in the speed towards the end of the first night is observed. After a small drop at the beginning of the day a second peak can be seen. There is a peak in the speed towards the end of the second night. This time, however, the speed reduces significantly towards the end of the night. As the leaf has only been fixed shortly before the acquisition started the motion pattern observed during the second night is probably more characteristic for this kind of experiment. The average normal magnitude is positive indicating an overall upward motion. We see that in the second half of the day almost the entire motion can be attributed to the tangential velocity while there is a significant normal component otherwise. During the motion peak at the onset of the second night there is a substantial normal motion. But the second peak is again dominated by the tangential component.

The free leaf (Fig. 9.8 b) exhibits a similar motion pattern than the fixed leaf during the second night. There is a clear rise in speed during the beginning of the night and a second broader peak towards the end. The motion decreases already before the next day begins. After the initial strong motion, which corresponds to the bending of the lobes, a periodic motion pattern of the leaf is observed. Here the entire movement consists of tangential velocity. The leaf is moving from one side to the other without any movement perpendicular to the surface. Apart from this time interval tangential and normal velocity components are correlated indicating a truly three-dimensional motion.

9.4 Leaf Growth

How the areal extension of the observed leaves can be computed from the range data and flow has been described in Chap. 6. As mentioned there, we require the 3D data to be on the same scale as the

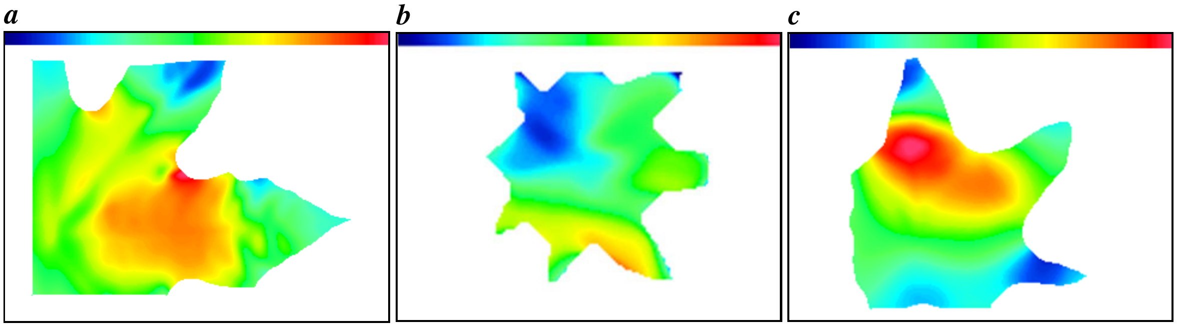


Figure 9.9: *Relative growth rate of some example leaves:* ***a*** *fixed leaf* $[-1,3]\%/h$*,* ***b*** *free leaf at day time* $[-1,3]\%/h$ *and* ***c*** *free leaf at night* $[-2,6]\%/h$*.*

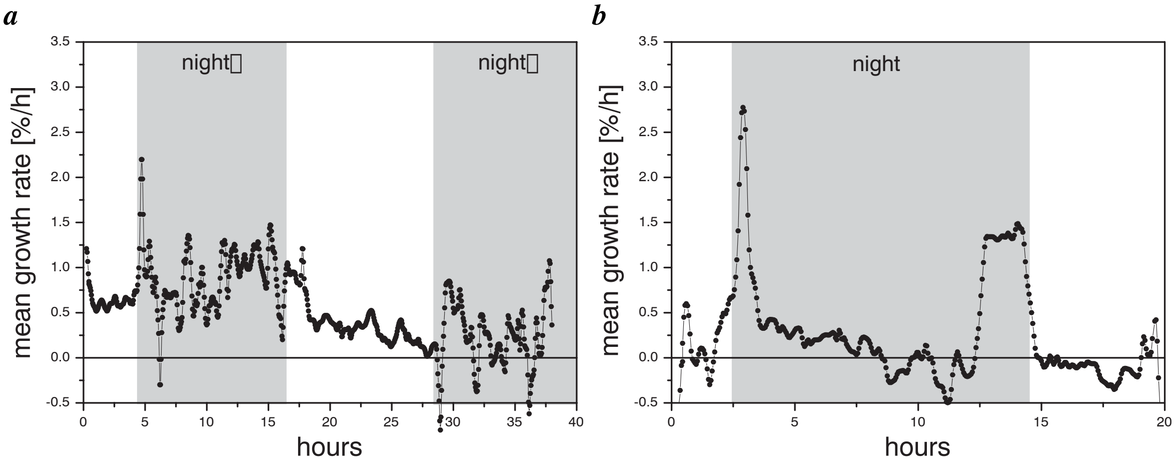


Figure 9.10: *Averaged relative growth rate over time:* ***a*** *fixed leaf,* ***b*** *free leaf.*

flow. Thus a smoothness parameter $\alpha = 10$ is used for both range data and range flow, see Sect. 2.2.2 and Sect. 5.2.1. Because a second order process is computed small perturbations do already result in large estimation errors. As discussed in Sect. 6.2.2 the second level of the Gaussian pyramid is used to average the computed growth rates.

Growth is not distributed uniformly but seems to occur at different locations (Fig. 9.9). Such a patchy behaviour has also been found on fixed leaves [Walter, 2001]. There are some areas with ”negative growth”. As it is not very likely that the leaves are shrinking this indicates a problem with the growth estimation algorithm. Such negative expansion rates occur mainly at the outer regions where high velocities and surface slopes are encountered. Under such conditions the growth estimation is not very reliable (Sect. 6.2.2).

Under the assumption that the errors are randomly distributed reliable mean values can be computed (Fig. 9.10). For the fixed leaf (Fig. 9.10 a) rather noisy results are obtained during the night periods. There is no clear pattern in the first night after which the mean growth rate drops during the day. In both nights we find a rise shortly after the light was turned off. There seems to be an increase in the growth towards the end of the night but the results do not allow a definite statement.

For the free leaf (Fig. 9.10 b) on the other hand a more pronounced diurnal growth pattern is observed. Shortly after the night sets in there is a strong peak in the growth rate. Afterwards the

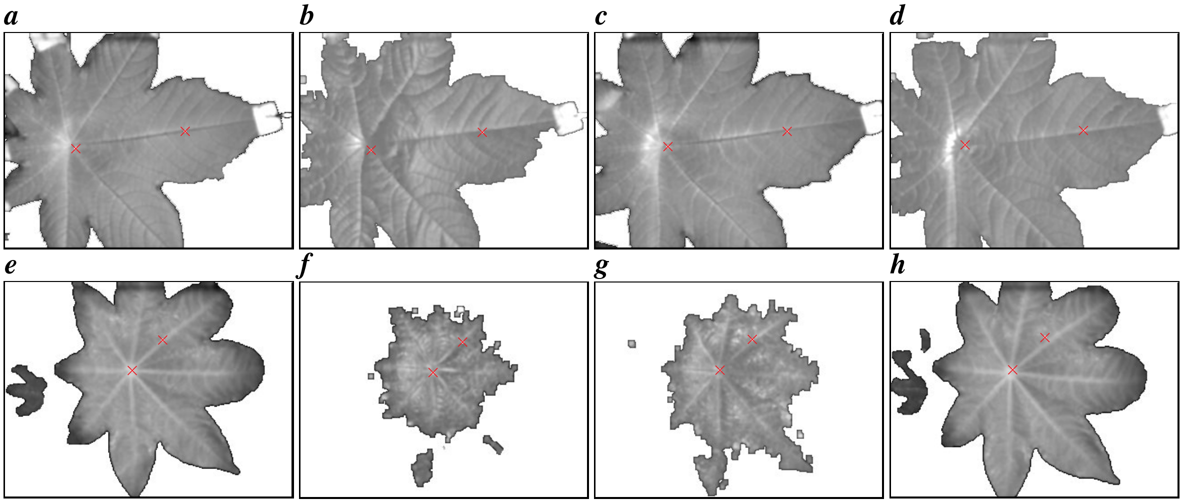


Figure 9.11: *Tracking a selected point over time. Fixed leaf at time:* ***a*** *0 h,* ***b*** *10 h,* ***c*** *20 h and* ***d*** *30 h. Free leaf at time:* ***e*** *0 h,* ***f*** *6 h,* ***g*** *12 h and* ***h*** *18 h.*

growth activity drops until about 2 hours before dawn. Then there is a large increase in the mean growth rate which remains high until the day begins. Thereafter the growth falls to zero where it remains for the rest of the observed day.

9.5 Single Point Observation

So far when examining the dynamic changes over time the values were averaged over the entire measured region. As we have seen in the presented maps the values do often vary greatly over a leaf at a given time. Therefore also the temporal changes for a small area around a selected point on the leaf are examined. Because the observed physical point moves it has to be tracked through the sequence. This is by no means an easy task for a long image sequence.

We track a selected point by computing the optical flow in the depth images. The normal structure tensor was used to compute optical flow in a relatively large (25×25 pixel) region around the selected point, see Sect. 4.2.2. In order to eliminate any global changes a high-pass filter ($\mathbb{1} - \boldsymbol{B}^7$) is used in a preprocessing step. To be robust against outliers Tukey's biweight M-estimator (4.41) is used with $r = 1$ (Sect. 4.1.5).

The resulting point positions are shown for the previously used example sequences of a free and a fixed leaf in Fig. 9.11. As there is no ground truth available the accuracy of the tracking can only be assessed visually. It can be seen that the selected points remain on the veins and that there is no large deviation even over an extended period of time. Thus it becomes possible to follow a selected region using this tracking algorithm.

9.5.1 Point Values

We present the measurements found at the tracked points (Fig. 9.11). Even though the tracking accuracy is certainly not below one pixel the various values at the point position are computed by means

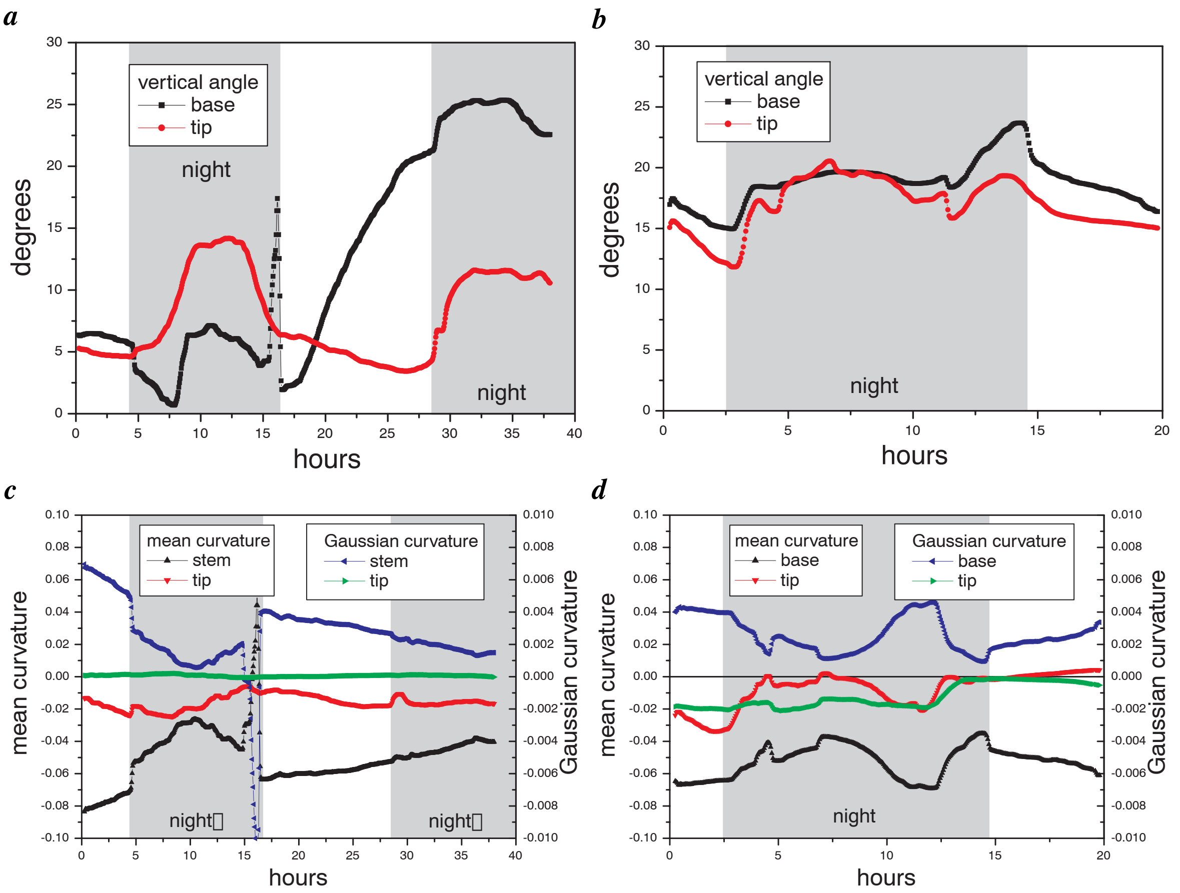


Figure 9.12: *Surface parameters at the tracked locations over time. Vertical angle for* ***a*** *fixed leaf and* ***b*** *free leaf. Mean and Gaussian curvatures for* ***c*** *fixed leaf and* ***d*** *free leaf.*

of a bilinear interpolation. Note that because the data is strongly smoothed the resulting values are representative for the surrounding region.

The vertical angles for the two points on the fixed leaf are shown in Fig. 9.12 a. At both locations the angle increases for some time after the night begins and decreases again during the last dark hour. While the angle measured at the tip drops continuously during the day we observe a steep incline in the angle near the base. This point is in an area that significantly bends over time.

In the case of the free leaf (Fig. 9.12 b) the vertical angles measured at the two locations are well correlated. As expected we find higher values at the point towards the tip. Initially the angle decreases until about 0.5 hours after the beginning of the night, then it quickly increases and remains at a higher value. For approximately the last two hours of the night a further increase in the angle is observed. As soon as the day begins the angles first drop strongly as the leaf orients itself more horizontally. For the rest of the day they slowly decrease further.

The two averaged curvature values over time for the fixed leaf both have greater magnitudes in the basal region (Fig. 9.12 c). For the tip a vanishing Gaussian curvature is found. The rather small mean curvature decreases slightly in magnitude during the first night before it almost reaches zero at the end of the night. The region is parabolic at first, becomes nearly flat and then parabolic again.

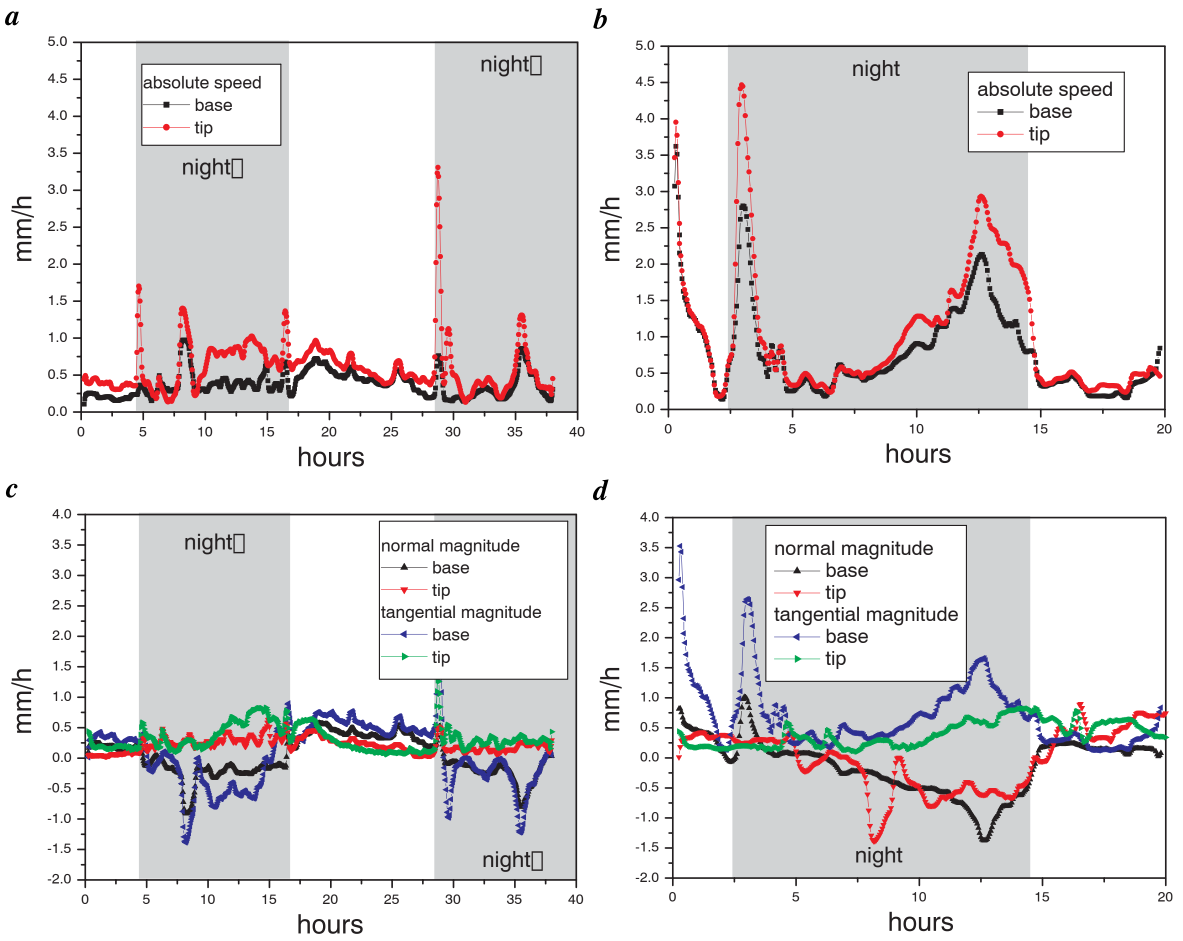


Figure 9.13: *Velocity values at the tracked locations over time. Absolute values of the velocity for* ***a*** *fixed leaf and* ***b*** *free leaf. Magnitudes of the normal and tangential velocities for* ***c*** *fixed leaf and* ***d*** *free leaf.*

At the base we find a different behaviour. The Gaussian curvature decreases steadily until the middle of the night after which there is a small increase. Just around daybreak the two curvatures change sign for less than an hour. During this time the surface bends upwards while otherwise it is bending downwards. After this sudden change in the orientation both curvatures reach a relatively high value which is slowly decreasing until the end of the observation period. The second day/night shift does not influence the curvatures. The leaf is flattening with increasing time. This suggests that one should wait nearly two days after fixing the leaf before a 2D analysis is performed.

On the free leaf we also find higher curvatures at the base than at the tip (Fig. 9.12 d). The two values are clearly correlated at the base where decreasing magnitudes in the first half of the night are found. During the second half there is an increase until about two hours before dawn. After this time the values get smaller and slowly increase again after the light was turned on. At the point closer to the tip there is an interesting phase in the first half of the night where the mean curvature vanishes but the Gaussian curvature does not. Hence both principal curvatures are of equal magnitude with opposite signs corresponding to a saddle point. Towards the end of the night and during the following day both values are very small which indicates a flat region.

The velocity values found at the tracked points are given in Fig. 9.13. The absolute speed curves (Fig. 9.13 a,b) do not differ greatly from the averaged values (Fig. 9.8). Only that for the fixed leaf there is sometimes a higher speed at the tip than at the base. The normal and tangential velocities for the fixed leaf (Fig. 9.13 c) interestingly show a negative normal velocity during both nights and a positive one during the day. The two positions move upwards at day time and downwards at night. The main difference in the tangential velocity occurs at the end of the first night when the tip exhibits much more movement.

The two tracked points on the free leaf show roughly the same normal velocities. As in the fixed leaf negative values are found during the night. This corresponds to the inward bending of the complete leaf. In contrast to the fixed leaf a higher tangential velocity is found at the base than at the tip for the freely moving leaf. The tangential velocity pattern found at the stem closely follows the overall speed.

9.5.2 Relative Values

The sensor coordinate system is obviously not a natural system to study leaves. The vein system would certainly be a better choice. First attempts towards an automated extraction of this coordinate system have already been done [Kirchgeßner, 1999; Kirchgeßner et al., 2000]. However, further improvements are needed before this can be coupled with the 3D analysis. Thus in this work a single point, usually the base, is used as reference and the leaf's structure and motion is examined relative to this point. A reference point is selected and tracked over time using the method described above.

The computed quantities at each surface point are then related to the distance to the tracked point. Let the tracked point be denoted by $\boldsymbol{x}_0 = [X_0 \; Y_0 \; Z_0]^T$, then the Euclidean distance to a point $\boldsymbol{x}$ is given by:

$$d = \sqrt{(X - X_0)^2 + (Y - Y_0)^2 + (Z - Z_0)^2} \, . \tag{9.8}$$

Note that the direct 3D distance is used rather than the distance along the surface for simplicity in this first analysis. Next all values found in a small distance range, here $1\,mm$, are averaged. An example is the difference in the absolute speed found near the base and the rest of the leaf. Another example is the growth rate. As a distance profile is obtained for each frame the result can be given as a map with the x axis indicating the time and the y axis the distance. Such maps are given in Fig. 9.14 for the fixed and free leaf sequence.

Looking at the speed relative to the base for the fixed leaf (Fig. 9.14 a) a large movement directly at the beginning and the end of the dark period is found. However this is mainly caused by the illumination change. Significantly more motion is observed at night than at day time. During this period of activity an increasing speed with rising distance from the base is observed. The growth rate as a function of the distance to the stem is given Fig. 9.14 b. Again we find much more activity at night than at day time. During the first night we clearly see that the growth rate is on average higher near the base and then drops towards the tip. However in the second night this effect is not very pronounced.

For the free leaf the relative speed also shows an incline away from the leaf base Fig. 9.14 c. There is a broad motion peak at the onset of the night when the lobe tips are bending downwards. This causes the observed area to decrease and only smaller distance values are available. Interestingly the outer

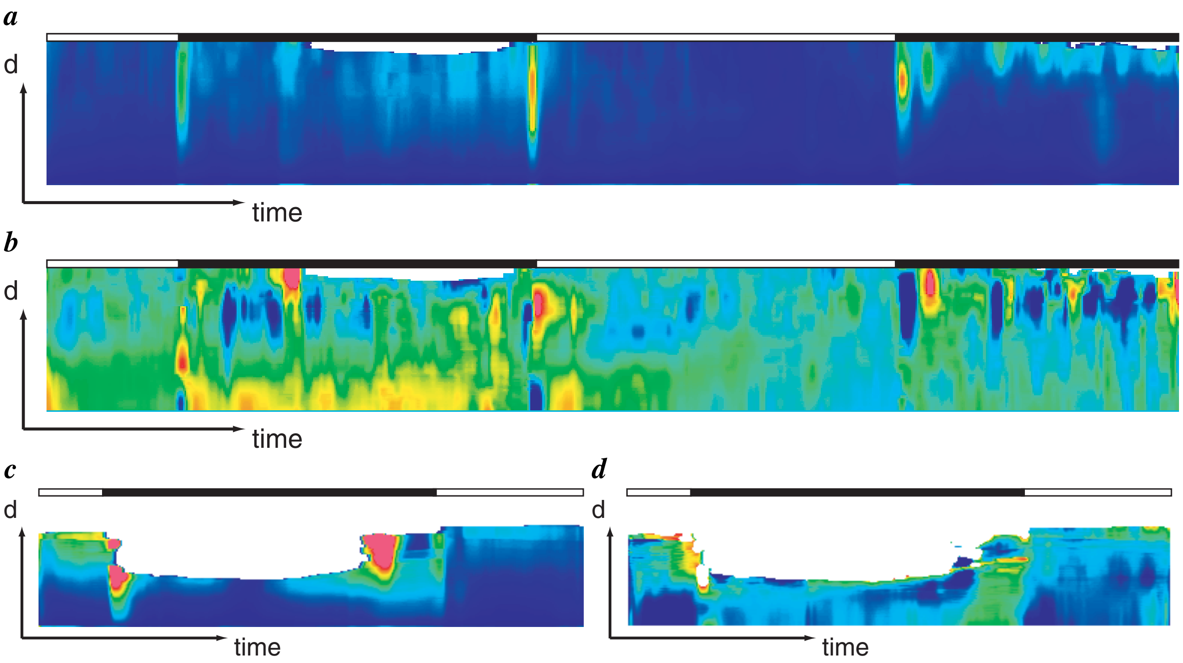


Figure 9.14: *Maps related to the distance to the tracked stem, distance range is* $[0, 100]\,mm$. *The black bars indicate the night periods. Fixed leaf over 38 hours:* ***a*** *relative speed* $[0, 4]\,mm/h$ *and* ***b*** *growth rate* $[-1, 3]\%/h$. *Free leaf over 20 hours:* ***c*** *relative speed* $[0, 5]\,mm/h$ *and* ***d*** *growth rate* $[-1, 5]\%/h$.

lobes are already moving upwards again before the illumination begins. This results in large speeds found in the frontal areas which also become observable again. After the day begins much smaller velocities are found which are increasing towards the tip. The distance related growth map for the free leaf is given in Fig. 9.14 d. There is a large growth rate found in the outer regions just after nightfall. Yet because of the very high velocities and tilting due to the bending this may not be accurate, see Sect. 6.2.2 and Sect. 8.2.3. There is little growth in the inner regions for most of the night. Quite interestingly a significantly increased growth activity is found at the end of the night when the outer lobes have already begun to orient themselves horizontally again. At day time both before and after the night smaller growth rates are observed which are increasing towards the outer regions.

Taking the average of the distance related profiles over the entire observed time results in mean profiles as shown in Fig. 9.15. The difference in the velocity magnitude between the base velocity and that found at a certain distance is increasing as we move away from the base for both the fixed and the free leaf (Fig. 9.15 a,b). This implies that on average the basal region moves the least. An increasing spread of velocities with increasing distance from the base is observed. Of course the areas near the base will have a similar movement than that point itself.

The averaged growth profiles show a decrease of the growth rate from the base towards the tip for the fixed leaf (Fig. 9.15 c). However, due to the large variation further away from the base this can only be seen as a trend which does not hold for any arbitrary time. For the free leaf we do on average observe the opposite trend with an increasing growth rate away from the base (Fig. 9.15 d). Yet as we have seen above the growth distribution changes quite a lot from one time step to another.

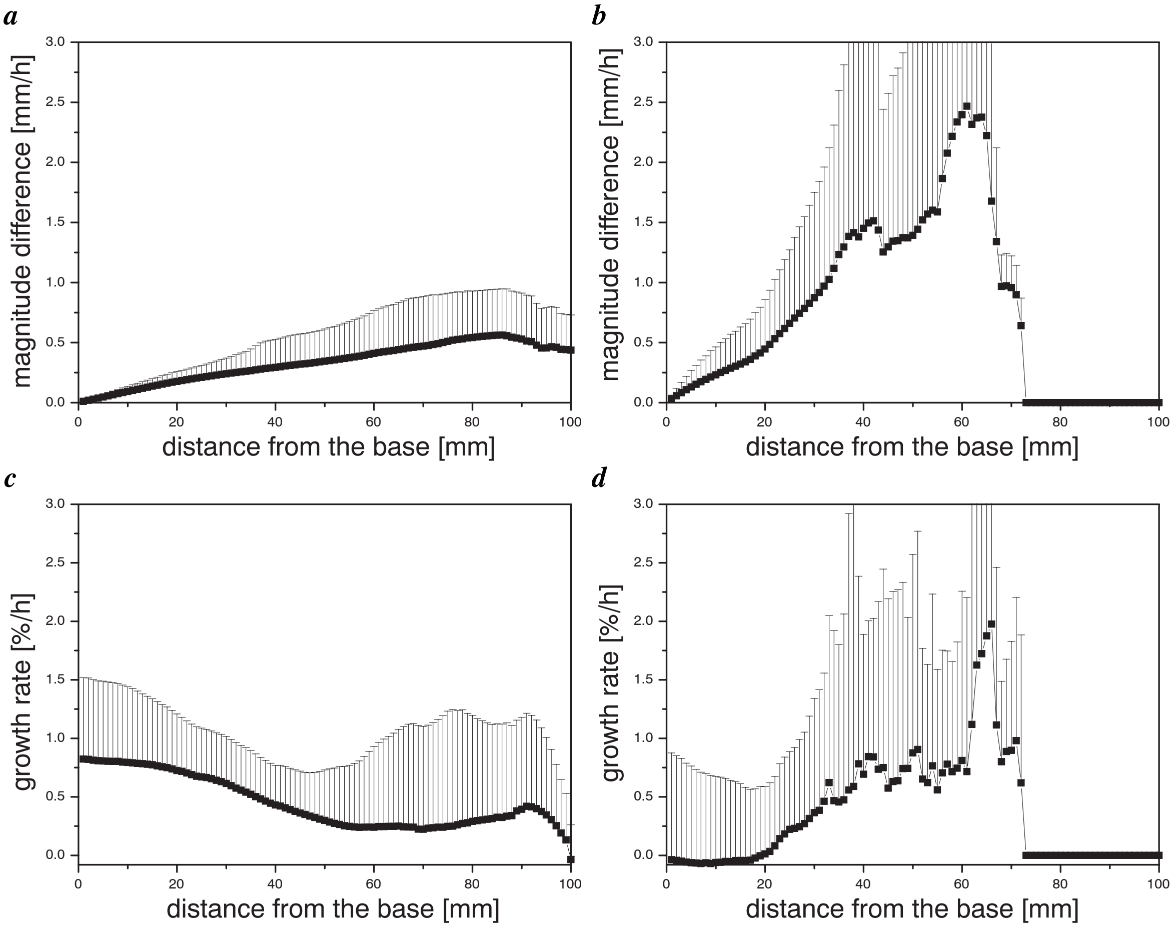


Figure 9.15: *Averaged values as a function of the distance from the base. Difference in the absolute value of the velocity for **a** fixed leaf and **b** free leaf. Growth rate for **c** fixed leaf and **d** free leaf.*

9.6 Summary

This chapter presented an exemplary analysis of range data and range flow to study the dynamic behaviour of castor bean leaves. The examined features range from the surface curvature over the normal and tangential velocities to the relative growth rate. For all the observable variables spatial maps and time varying averaged curves found on example leaves have been presented. In order to investigate the behaviour of a selected point on the leaf surface we showed how such a point can be tracked over time. This makes it possible to express the results in a radial leaf coordinate system attached to the base of the leaf. As an example the speed and growth has been related to the distance from the base.

In summary the presented method provides a framework for future botanical studies aimed to reveal the motion and growth patterns of plant leaves in a systematic experimental study.

Chapter 10

Conclusion and Outlook

In this work both an algorithmic framework and its experimental realisation for the measurement of the instantaneous velocity field and expansion rate of a deforming surface have been developed. This final chapter summarises the presented achievements and discusses some possible future extension.

10.1 Thesis Summary

To enable the computation of the velocity field (range flow) from range data sequences a novel constraint equation has been introduced (Chap. 3). Unfortunately does the constraint equation pose only one equation in the three unknown components of the velocity. Depending on the local data structure this ambiguity may be resolved by pooling the constraints over a small spatio-temporal neighbourhood. We were able to extend the 2D aperture problem known from optical flow computation to 3D and showed that three types of neighbourhoods have to be considered in this case.

On the simultaneously available intensity data a second constraint on the X and Y components of the velocity field has been derived. Enforcing both constraints can often, but not always, help to reduce the aperture problem. The local estimation of the range flow field is accomplished in a general total least squares (TLS) framework (Chap. 4). It could be demonstrated that the presented algorithm allows to compute the parameters of any linear partial differential equation. It has been shown that depth and intensity can be regarded as two separate channels that are readily combined in a multi-channel extension of the TLS algorithm put forward here. The developed computational scheme directly yields a measure on the confidence of the solution. Furthermore the presence and the degree of linear dependencies in the data can automatically be detected and appropriately constraint solutions computed.

The TLS algorithm does not generally yield a dense full parameter field. Due to the mentioned linear dependencies (the aperture problem) in the local data arrays only a certain subspace of the complete parameter space can be determined uniquely. On a bigger data array such as image or range data sequences the surrounding information can be used to regularise the solution. Towards this end a new data constraint has been introduced in a variational scheme (Chap. 5). This data term exploits all the information available from the previous TLS solution. In particular the subspace where a unique solution exists is treated differently from its orthogonal complement. It could be demonstrated that this regularisation scheme spreads the information until a dense and full parameter field is reached.

Additionally, the orientation analysis given by the image velocity version of the TLS algorithm, i.e. the standard structure tensor, has been used to derive a novel smoothness constraint. This leads to a new version of anisotropic diffusion. It has been shown that this diffusion greatly improves the velocity estimation in noisy image sequences.

Another dynamic parameter of a deforming surface is the local expansion rate. A method to calculate this growth rate from the dense range flow field and the range data has been developed (Chap. 6). The analysis of the uncertainty in the estimated growth rate revealed that for high surface slopes and fast tilts the error is dominated by the noise in the measured depth data and otherwise by the error in the range flow field.

For an experimental realisation a Biris laser range finder and a structured light sensor have been employed (Chap. 7). An experimental analysis of the noise in the captured 3D data revealed a high accuracy. Using the thus determined noise levels the performance of the range flow and growth estimation algorithms have been examined on synthetic data (Chap. 8). This analysis showed that the 3D velocity can be determined very accurately, which could also be verified on real test sequences.

Finally the presented method has been applied to the study of living castor bean leaves (Chap. 9). In these experiments a diurnal rhythm in surface parameters such as the overall orientation and the curvature has been observed. The curvature is significantly higher during the night with a peak around the middle of the night. The observed movements were also higher during the night with a large peak at the beginning and another around 2 hours before the end of the night. These motion peaks result from the bending and straightening of the leave lobes. After this straightening, an increased growth rate has been found during the last 2 hours of the dark period. This is also the only prolonged time with a substantial mean growth rate. In conclusion it could be demonstrated that the developed method can be used in practice to study the dynamic behaviour of plant leaves.

10.2 Future Work

The main future activity is the use off the developed measurement system for systematic botanical studies. Apart from this practical work the following experimental improvements are suggested:

- The analysis of leaf motion showed that the speed varies greatly during the day. In particular in the beginning of the night large movements are encountered. The accuracy of the estimated velocity field and the growth rate depend on the displacement per frame. Therefore the results could be improved upon by adapting the frame rate to the observed speed. This would also require an adaptation of the temporal derivative calculation to the change in the frame rate.

- The present setup is limited by the maximum angle between sensor and the surface that can still be measured. In future experiments a sensor that can capture the complete leaf surface regardless of the amount of bending would be desirable. Example solutions are a moving camera or a moving sensor consisting of two cameras or a camera and a projector.

- The above mentioned techniques as well as the two sensors used here require the scanned object to be stationary while the 3D data is measured. In a controlled laboratory environment this poses no restriction, however when experiments are to be carried out in the field or even in a ventilated growth chamber this requirement is not fulfilled. For such environments a sensor that can measure the 3D surface data within parts of a second is required. A multi camera system with cameras placed surrounding the observed leaf suggests itself. Yet in such a system rapid leaf movements create the need for a registration of the leaf surface, which might introduce additional hurdles.

With respect to the developed algorithms the following possible extensions can be named:

- Sensible thresholds for the TLS algorithm can be derived from the noise level in the data. Currently this noise is estimated off-line. In order to make the estimation framework more robust in changing environments it would be useful to automatically set the thresholds. This would also allow to locally adapt the threshold values to the data structure. In principle such an adaptation can be done by analysing the variation of the eigenvalues over the images.
- So far diagonal matrices are used for the TLS equilibration. If the covariance matrix of the data vector is not diagonal this should be taken into account to further increase the accuracy. In particular an investigation of the dependencies introduced by applying derivative filters to the same noise source appears promising.
- The TLS parameter regularisation for range or optical flow has been investigated using a membrane smoother. On the continuous plant leaves this is well justified. However, it would be interesting to employ the anisotropic smoother on parameter fields as well. This should enable the computation of dense parameter fields even on discontinuous data.
- The diffusion process to interpolate dense parameter fields is currently realised on the sensor grid. In the case of range data the actual objects form surfaces embedded in 3D space. It would therefore be more sensible to directly regularise on this surface.
- Growth in 2D can be incorporated as a parameter in the presented TLS framework. But further investigations are needed to do the same for growth on surfaces. Here it seems promising to work in the tangential surface coordinate frame.

Appendices

Appendix A

Least Squares Range Flow

This appendix describes an ordinary least squares (OLS) approach to the problem of range flow estimation. Using this method it is also possible to obtain good range flow estimates [Barron and Spies, 2000; Barron et al., 2000]. As with the TLS solution in Sect. 4.3 we assume a locally constant range flow field. For the sake of notational simplicity we use (3.19) instead of (3.24), the same argumentation also holds for the latter case. Stating the range flow motion constraint equation (3.19):

$$W = Z_t + Z_X U + Z_Y V \ , \tag{A.1}$$

at every point in the considered neighbourhood, we arrive at a system of equations (here we replace W with -W):

$$\underbrace{\begin{bmatrix} Z_X^1 & Z_Y^1 & 1 \\ & \vdots & \\ Z_X^n & Z_Y^n & 1 \end{bmatrix}}_{\boldsymbol{A}} \underbrace{\begin{bmatrix} U \\ V \\ W \end{bmatrix}}_{\boldsymbol{s}} = \underbrace{\begin{bmatrix} -Z_t^1 \\ \vdots \\ -Z_t^n \end{bmatrix}}_{\boldsymbol{b}} . \tag{A.2}$$

The least squares solution is then found via the pseudo-inverse [Press et al., 1992]:

$$\boldsymbol{f} = \left(\boldsymbol{A}^T \boldsymbol{A}\right)^{-1} \boldsymbol{A}^T \boldsymbol{b} \, . \tag{A.3}$$

For completion we state the matrix to be inverted:

$$\boldsymbol{A}^T \boldsymbol{A} = \begin{bmatrix} < Z_X^2 > & < Z_X Z_Y > & < Z_X > \\ < Z_X Z_Y > & < Z_Y^2 > & < Z_Y > \\ < Z_X > & < Z_Y > & < 1 > \end{bmatrix} . \tag{A.4}$$

Here $< \cdot >$ denotes local averaging. Note that this is simply the upper 3×3 part of the structure tensor $\boldsymbol{J}$ defined in (4.11).

We need to determine if that solution is actually the desired full flow or if there is an aperture problem in the considered neighbourhood. This can be done by analysing the eigenvalues and eigenvectors of the matrix $\boldsymbol{B} = \boldsymbol{A}^T \boldsymbol{A}$. If there is enough spread in the data all three eigenvalues will be non-vanishing. This is accounted for by simply thresholding the smallest eigenvalue λ_3, we assume the eigenvalues to be ordered with descending magnitude:

$$\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq 0 \; . \tag{A.5}$$

If there are only two directions in witch the flow can be determined accurately only the two larger eigenvalues will exceed the threshold. In that case the computed velocity $\boldsymbol{f}$ is only accurate in the directions given by the eigenvectors $\hat{e}_1, \hat{e}_2$ to the eigenvalues λ_1 and λ_2. Thus the line flow can be computed by projecting the computed flow onto these two directions:

$$\boldsymbol{f}_l = (\boldsymbol{f} \cdot \hat{e}_1)\hat{e}_1 + (\boldsymbol{f} \cdot \hat{e}_2)\hat{e}_2 \; . \tag{A.6}$$

Following the same argument we can compute plane flow when only the largest eigenvalue is greater than the set threshold:

$$\boldsymbol{f}_p = (\boldsymbol{f} \cdot \hat{e}_1)\hat{e}_1 \; . \tag{A.7}$$

When all eigenvalues are non-zero we have obviously computed the full flow already:

$$\boldsymbol{f}_f = \boldsymbol{f} = (\boldsymbol{f} \cdot \hat{e}_1)\hat{e}_1 + (\boldsymbol{f} \cdot \hat{e}_2)\hat{e}_2 + (\boldsymbol{f} \cdot \hat{e}_3)\hat{e}_3 \; . \tag{A.8}$$

The above scheme breaks down if $\boldsymbol{A}^T\boldsymbol{A}$ can not be inverted, because $\boldsymbol{f}$ needed in (A.6) and (A.7) can not be computed. We nevertheless want to find appropriate normal flow vectors. In order to do so we reformulate the problem. In the coordinate system given by the eigenvectors of $\boldsymbol{A}^T\boldsymbol{A}$ the system of equations takes the form:

$$\begin{bmatrix} \lambda_1 & & \\ & \lambda_2 & \\ & & \lambda_3 \end{bmatrix} \begin{bmatrix} U' \\ V' \\ W' \end{bmatrix} = \begin{bmatrix} d_1' \\ d_2' \\ d_3' \end{bmatrix} \; . \tag{A.9}$$

Primed variables indicate rotated quantities:

$$\begin{bmatrix} U' \\ V' \\ W' \end{bmatrix} = \boldsymbol{R} \begin{bmatrix} U \\ V \\ W \end{bmatrix} \quad \text{with} \quad \boldsymbol{R} = [\hat{e}_1 \; \hat{e}_2 \; \hat{e}_3] \, , \tag{A.10}$$

and

$$\boldsymbol{d}' = \boldsymbol{R}\boldsymbol{d} \quad \text{where} \quad \boldsymbol{d} = \boldsymbol{A}^T\boldsymbol{b} \, . \tag{A.11}$$

The solution to (A.9) is straightforward if all eigenvalues are non-vanishing. As the trace of a matrix is invariant under rotations and we are dealing with a positive definite matrix[1], we can exclude the case $\lambda_1 = \lambda_2 = \lambda_3 = 0$ by requiring a non-vanishing trace. If one or two of the eigenvalues are less than a set threshold τ we are dealing with line or plane flow respectively. The correct normal flow can then be computed as follows:

[1] All eigenvalues are greater or equal to zero.

- **line flow** ($\lambda_1, \lambda_2 > \tau;\ \lambda_3 < \tau$): In this case there are only two equations from (A.9) which can be written explicitly as:

  $$\begin{aligned} e_{11}U + e_{12}V + e_{13}W - \frac{d'_1}{\lambda_1} &= 0 && \text{(A.12)} \\ e_{21}U + e_{22}V + e_{23}W - \frac{d'_2}{\lambda_2} &= 0\,. && \text{(A.13)} \end{aligned}$$

  These two equations define two constraint planes and the line normal flow is given by the point on their intersecting line with minimal distance from the origin. Thus the line flow can be computed as described in Sect. 3.3.

- **plane flow** ($\lambda_1 > \tau;\ \lambda_2, \lambda_3 < \tau$): Equation (A.9) reduces to only one constraint:

  $$e_{11}U + e_{12}V + e_{13}W - \frac{d'_1}{\lambda_1} = 0\,. \tag{A.14}$$

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

  $$\boldsymbol{F}_p = \frac{\frac{d'_1}{\lambda_1}}{e_{11}^2 + e_{12}^2 + e_{13}^2} \begin{bmatrix} e_{11} \\ e_{12} \\ e_{13} \end{bmatrix} = \frac{d'_1}{\lambda_1} \begin{bmatrix} e_{11} \\ e_{12} \\ e_{13} \end{bmatrix}\,. \tag{A.15}$$

As $\boldsymbol{A}^T\boldsymbol{A}$ is real and symmetric the eigenvalue decomposition can always be carried out via Jacobi rotations, [Press et al., 1992]. Thus the above line and plane flow estimates can be calculated even when the matrix is singular.

Appendix B

Direct Regularisation

In Chap. 4 we described a method to estimate a parameter field at every pixel. Because of the aperture problem this requires that further smoothness constraints are imposed on the estimated parameter field. The constraint used in Chap. 4 amounts to the assumption of a locally constant parameter field. Instead of performing such a local analysis we might want to find a global solution to the sought parameter field. This can be achieved using the variational approach described in Chap. 5. The here presented iterative method extends the optical flow algorithm by Horn and Schunk [1981] to compute smooth parameter fields.

B.1 Smooth Parameter Fields

As stated in Sect. 5.1.1 the original constraint equation (4.1) at every data point results in the following data term:

$$D = (\boldsymbol{a}^T \boldsymbol{m} - b)^2 \; . \tag{B.1}$$

In the case of multiple data channels this is readily extended to:

$$D = \sum_c \beta_c (\boldsymbol{a}_c^T \boldsymbol{m} - b_c)^2 \; . \tag{B.2}$$

Where the β_c are used to weight the individual channels, compare Sect. 4.1.3. The membrane model smoothness term (5.15) becomes:

$$V = \sum_{i=1}^{m} (\nabla m_i)^2 \;\; , \tag{B.3}$$

Combining data and smoothness term yields the Lagrange function for the direct parameter estimation:

$$L = \sum_c \beta_c (\boldsymbol{a}_c^T \boldsymbol{m} - b_c)^2 + \alpha \sum_{i=1}^{m} (\nabla m_i)^2 \;\; . \tag{B.4}$$

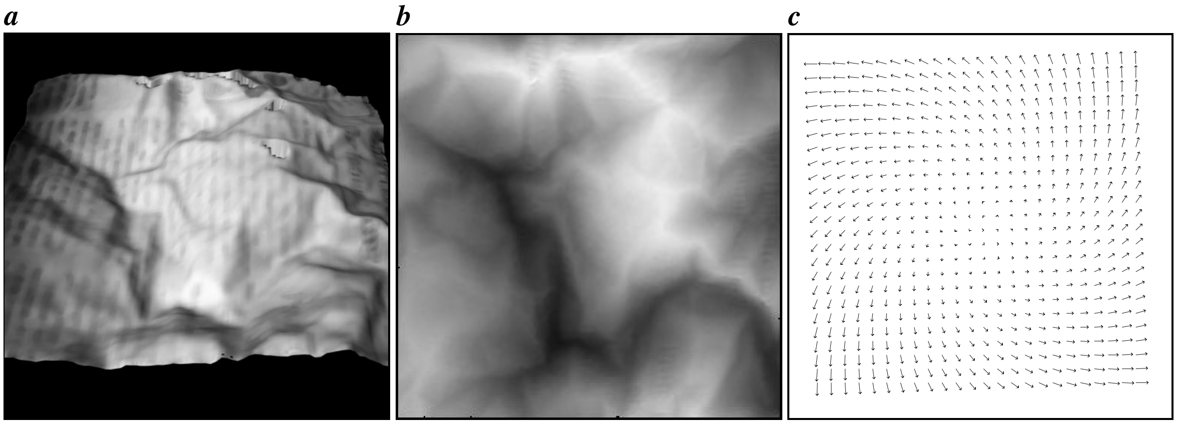


Figure B.1: *Test sequence generated from a real depth map:* ***a*** *rendered texture-mapped data,* ***b*** *depth map,* ***c*** *correct X, Y–flow field.*

Written in vector notation the Euler-Lagrange equations (5.2) become:

$$2\sum_c \boldsymbol{a}_c(\boldsymbol{a}_c^T\boldsymbol{m} - b_c) - 2\alpha\Delta\boldsymbol{m} = 0 \ . \tag{B.5}$$

If we approximate the Laplacian as difference between a local average and the central value $\Delta\boldsymbol{m} = \bar{\boldsymbol{m}} - \boldsymbol{m}$ this can be rewritten as:

$$\sum_c (\boldsymbol{a}_c\boldsymbol{a}_c^T)\boldsymbol{m} - \alpha\bar{\boldsymbol{m}} + \alpha\boldsymbol{m} = \sum_c \boldsymbol{a}_c b_c \tag{B.6}$$

$$\left(\alpha\mathbb{1} + \sum_c (\boldsymbol{a}_c\boldsymbol{a}_c^T)\right)\boldsymbol{m} = \alpha\bar{\boldsymbol{m}} + \sum_c \boldsymbol{a}_c b_c \ . \tag{B.7}$$

An iterative solution is found using the following update:

$$\boldsymbol{m}^{k+1} = \alpha\boldsymbol{A}_1^{-1}\bar{\boldsymbol{m}}^k + \boldsymbol{A}_1^{-1}\boldsymbol{b} \ . \tag{B.8}$$

With $\boldsymbol{A}_1 = \alpha\mathbb{1} + \sum_c(\boldsymbol{a}_c\boldsymbol{a}_c^T)$ and $\boldsymbol{b} = \sum_c \boldsymbol{a}_c b_c$. The existence of the inverse of $\boldsymbol{A}_1$ follows from the Sherman-Morrison-Woodbury Lemma for $\alpha > 0$, see Sect. 5.1.3.

B.2 Range Flow

We explicitly state the resulting formula for a direct range flow estimation. Treating depth and intensity as two channels with the constraint equation (3.19) and (3.36) respectively we arrive at the following Lagrangian:

$$\begin{aligned} L \ = \ & (Z_XU + Z_YV + Z_t + W')^2 + \beta\,(I_XU + I_YV + I_t)^2 \\ & + \alpha\,(U_x^2 + U_y^2 + V_x^2 + V_y^2 + W_x^2 + W_y^2) \end{aligned} \tag{B.9}$$

Here we choose β as the relative weight on the two channels instead of an individual β_c. Also we write $W' = -W$ to avoid the negative sign. Then the Euler-Lagrange equations (B.7) read:

$$\begin{bmatrix} (\alpha + Z_X^2 + \beta I_X^2) & (Z_XZ_Y + \beta I_XI_Y) & Z_X \\ (Z_XZ_Y + \beta I_XI_Y) & (\alpha + Z_Y^2 + \beta I_Y^2) & Z_Y \\ Z_X & Z_Y & \alpha \end{bmatrix} \begin{bmatrix} U \\ V \\ W' \end{bmatrix} = \begin{bmatrix} \alpha\bar{U} - Z_XZ_t - \beta I_XI_t \\ \alpha\bar{V} - Z_YZ_t - \beta I_YI_t \\ \alpha\bar{W}' - Z_t \end{bmatrix} . \tag{B.10}$$

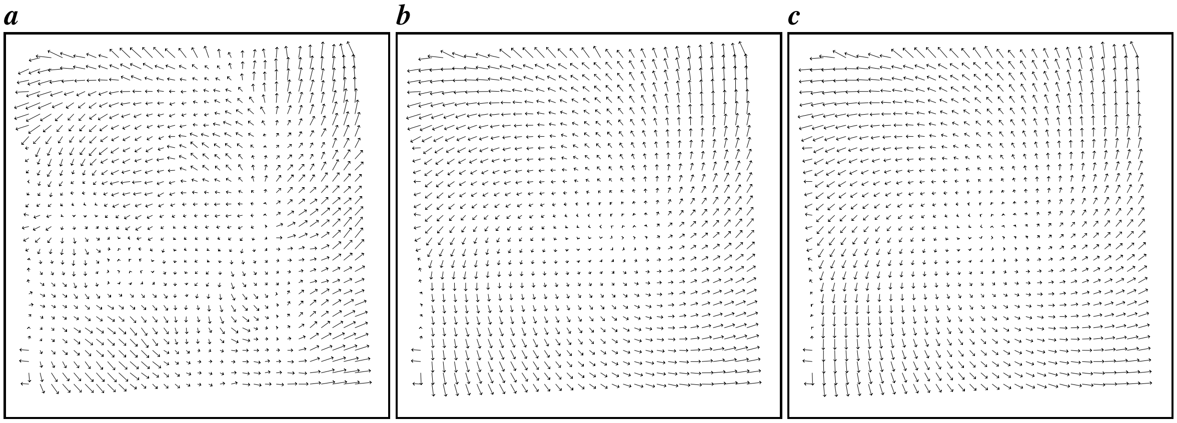


Figure B.2: *X, Y–flow obtained with the iterative global smoothness algorithm:* ***a*** *100 iterations,* ***b*** *500 iterations and* ***d*** *1000 iterations.*

This is solved iteratively using (B.8). As an example the acquired depth map of a crumpled sheet of paper is used. This depth map is then warped with a known flow field containing both divergence and rotation in the XY-components and a constant Z-flow. Here even sampling is assumed. Figure B.1 shows the data and the correct XY-flow field.

We use the algorithm described above with $\alpha = 10$ and a 5×5 Box filter for the mean. Figure B.2 shows the obtained XY-flow field after increasing number of iterations. Quantitative results are (density is 100%):

iterations	magnitude error [%]	directional error [°]
100	37.9(36.7) ± 17.1	24.5(24.4) ± 10.6
200	26.5(25.2) ± 13.3	15.9(16.1) ± 7.4
500	9.8(8.9) ± 6.2	6.4(6.0) ± 3.3
1000	1.9(1.4) ± 1.8	2.9(2.8)± 1.2

When running a high enough number of iterations very good results are achieved using this algorithm.

Appendix C

Anisotropic Diffusion Discretisation

This appendix shows how the anisotropic diffusion system described in Chap. 5 can be implemented in a simple and efficient way. More details on this discretisation and a comparison to other algorithms can be found in Scharr [2000].

The equations (5.36) can be solved numerically using finite differences. This reduces to the diffusion systems (5.35) for $c = 0$ and $\alpha = 1$. For simplicity we drop the index k again, yet one should be aware that the following has to be applied to each component of the parameter vector $\boldsymbol{p}$. Spatial derivatives can be replaced by derivative filters, while the easiest way to discretise $\partial p/\partial s$ consists in a forward difference approximation. The resulting explicit scheme allows to calculate all values at a new time step directly from the ones in the previous step. It has the basic structure:

$$\frac{p^{l+1} - p^l}{\tau} = \frac{c}{\alpha}\,(p^l - q) + \boldsymbol{A}^l * p^l \; , \tag{C.1}$$

where τ is the time step size and p^l denotes the approximation of $p(s)$ at every pixel at time $s = l\tau$. The expression $\boldsymbol{A}^l * p^l$ is a discretisation of $\nabla \cdot (\boldsymbol{D}\nabla p)$. It boils down to the convolution of the image stack with a space (3D) and (diffusion) time dependent mask $\boldsymbol{A}^l$. The explicit update now reads as:

$$p^{l+1} = \left(\mathbb{1} + \tau \boldsymbol{A}^l\right) * p^l + \tau\,\frac{c}{\alpha}\,(p^l - q) \; . \tag{C.2}$$

In order to compute $\boldsymbol{A}$ we rewrite the differential operator in (5.35) as:

$$\begin{array}{rcl} \nabla \cdot (\boldsymbol{D}\nabla p) & = & \partial x_1 (D_{11}\partial x_1 p + D_{12}\partial x_2 p + D_{13}\partial x_3 p) \\ & + & \partial x_2 (D_{12}\partial x_1 p + D_{22}\partial x_2 p + D_{23}\partial x_3 p) \\ & + & \partial x_3 (D_{13}\partial x_1 p + D_{23}\partial x_2 p + D_{33}\partial x_3 p) \\ & = & \partial x_1 j_1 + \partial x_2 j_2 + \partial x_3 j_3 \; . \end{array}$$

This expression can be evaluated using only known values from time step l. The derivatives ∂x_i are computed using recently derived kernels that are optimised with respect to rotational invariance [Scharr et al., 1997; Jähne et al., 1999]:

$$\begin{array}{rcccccc} \partial x_1 & = & \frac{1}{512} & (1,0,-1)_{x_1} & * & (3,10,3)_{x_2} & * & (3,10,3)_{x_3} \\ \partial x_2 & = & \frac{1}{512} & (3,10,3)_{x_1} & * & (1,0,-1)_{x_2} & * & (3,10,3)_{x_3} \\ \partial x_3 & = & \frac{1}{512} & (3,10,3)_{x_1} & * & (3,10,3)_{x_2} & * & (1,0,-1)_{x_3} \end{array} \; . \tag{C.3}$$

The complete algorithm can thus be broken into single steps as follows:

1. Calculate the structure and diffusion tensor using the optimised derivative filters.
2. Calculate the flux components j_1, j_2 and j_3.
3. Calculate $\nabla \cdot (\boldsymbol{D}\nabla p) = \partial x_1 j_1 + \partial x_2 j_2 + \partial x_3 j_3$.
4. Update in an explicit way using Eq. (C.2).
5. Repeat from step 1.

For completeness we state the explicit update equation for (5.37) where the TLS subspace data term is used as data term together with the anisotropic regulariser:

$$p^{l+1} = \left(\mathbb{1} + \tau \boldsymbol{A}^l\right) * p^l - \tau \, \frac{\omega}{\alpha} \, \boldsymbol{P}(p^l - q) \; . \tag{C.4}$$

Using $\mathbb{1} = \boldsymbol{P} + \boldsymbol{P}^\perp$ this can also be written as:

$$p^{l+1} = \left(\mathbb{1} + \tau \boldsymbol{A}^l\right) * \boldsymbol{P}^\perp p^l + \left(\mathbb{1} + \tau \boldsymbol{A}^l\right) * \boldsymbol{P} p^l - \tau \, \frac{\omega}{\alpha} \, \boldsymbol{P}(p^l - q) \; . \tag{C.5}$$

This is the previous anisotropic diffusion without any data term in the perpendicular subspace. In the TLS subspace, on the other hand, there is an additional term that corrects for the deviation from the TLS solution $\boldsymbol{q}$.

Appendix D

Synthetic Expansion Data

To test the range flow estimation and in particular the computation of expansion rates some artificial test sequences were used. Here it is described how this data was produced. For the generation of range data we need to employ a model of the sensor. Towards this end we use both parallel or orthogonal and perspective projection, this corresponds to a sensor using a moving stage and one using a lens, e.g. structured lighting. For each point on the sensor array we then construct the corresponding ray of sight and intersect it with the observed object. For an analytical solution to exist the observed object thus needs to have an analytical description. Here we consider a planar object and a sphere.

For simplicity we always use the sensor coordinate system as world coordinate system. The line of sight can be represented by a point $\boldsymbol{x}_0$ a direction $\hat{\boldsymbol{l}}$ and a parameter s:

$$\boldsymbol{x}(s) = \boldsymbol{x}_0 + s\,\hat{\boldsymbol{l}}\ . \tag{D.1}$$

Given sensor coordinates x, y this line is given by:

$$\boldsymbol{x}_0 = [x\,y\,0]^T \quad \text{and} \quad \hat{\boldsymbol{l}} = [0\,0\,1]^T\ , \tag{D.2}$$

for orthogonal projection. For a perspective projection using focal length f we obtain:

$$\boldsymbol{x}_0 = [0\,0\,0]^T \quad \text{and} \quad \hat{\boldsymbol{l}} = \frac{1}{\sqrt{x^2+y^2+f^2}}[-x\ -y\ f]^T\ . \tag{D.3}$$

Apart from the 3D structure we want the objects to have some intensity texture. This has to be modelled as a function of the 2D surface.

D.1 Plane

A plane can be described by its surface normal $\hat{\boldsymbol{n}}$ and the distance d to the origin:

$$\boldsymbol{x}^T\hat{\boldsymbol{n}} + d = 0\ . \tag{D.4}$$

In order to investigate the algorithms it is intuitive to describe the surface normal in spherical coordinates ϑ, φ. In particular the angle ϑ between Z-axis and the plane normal is of interest. From the angles we readily obtain the surface normal as:

$$\hat{\boldsymbol{n}} = [\sin\vartheta\cos\varphi,\ \sin\vartheta\sin\varphi,\ -\cos\vartheta]^T \ . \tag{D.5}$$

an intensity value has to be associated with each point $\boldsymbol{x}$ on the plane. Thus we introduce 2D plane coordinates x_p, y_p relative to a point $\boldsymbol{d}$ on the plane. Initially this point can be chosen to be $\boldsymbol{d} = [0\,0\,d]^T$. In order to define the axis we need a direction vector $\hat{\boldsymbol{y}}_p$ in the plane. This can be computed from the cross-product between surface normal and the X-axis[1]: $\boldsymbol{y}_p = \hat{\boldsymbol{n}} \times [1\,0\,0]^T = [0\,n_3\ -n_2]$. The other axis on the plane is then given by $\hat{\boldsymbol{x}}_p = \hat{\boldsymbol{y}}_p \times \hat{\boldsymbol{n}}$. Plane coordinates are determined by the projection of the point vector $\boldsymbol{r} = \boldsymbol{x} - \boldsymbol{d}$ in the plane unto the two axis:

$$x_p = \hat{\boldsymbol{x}}_p^T \boldsymbol{r} \ , \tag{D.6}$$

$$y_p = \hat{\boldsymbol{y}}_p^T \boldsymbol{r} \ . \tag{D.7}$$

Often one wants to describe the movement in the plane by polar coordinates r, θ defined by:

$$\theta = \arctan\frac{y_p}{x_p} \ , \tag{D.8}$$

$$r = \sqrt{x_p^2 + y_p^2} \ . \tag{D.9}$$

Movements on the plane do not change the 3D structure but can be modelled in terms of θ, r and $\boldsymbol{d}_p' = \boldsymbol{d} + \boldsymbol{f}_p$, with $\boldsymbol{f}_p \perp \hat{\boldsymbol{n}}$. In particular does an increase in r correspond to an expansion of the surface. Any motion component along the surface normal or a change of the surface normal results in a change in the distance d. For each time frame we then compute the range data as seen with the range sensor. For parallel projection (D.2) we have:

$$X(x,y) = x \ , \tag{D.10}$$

$$Y(x,y) = y \ , \tag{D.11}$$

$$Z(x,y) = -\frac{d + n_1 x + n_2 y}{n_3} \ . \tag{D.12}$$

For perspective projection we first insert (D.3) into (D.4) to obtain $s = \frac{d}{n_1 l_1 + n_2 l_2 - n_3 l_3}$. The range data arrays are then given by:

$$X(x,y) = \frac{-l_1 d}{n_1 l_1 + n_2 l_2 - n_3 l_3} \ , \tag{D.13}$$

$$Y(x,y) = \frac{-l_2 d}{n_1 l_1 + n_2 l_2 - n_3 l_3} \ , \tag{D.14}$$

$$Z(x,y) = \frac{l_3 d}{n_1 l_1 + n_2 l_2 - n_3 l_3} \ , \tag{D.15}$$

with $\boldsymbol{l}$ from (D.3). As stated before there is an intensity associated with each data point $I(x,y) = I(x_p, y_p) = I(r, \theta)$.

As we want to perform a quantitative performance analysis we need to know the correct range flow seen at each sensor element. Towards this end we compute the 3D location as described above

[1] Clearly the surface normal can not be parallel to the X-axis, as no sensor could measure this situation.

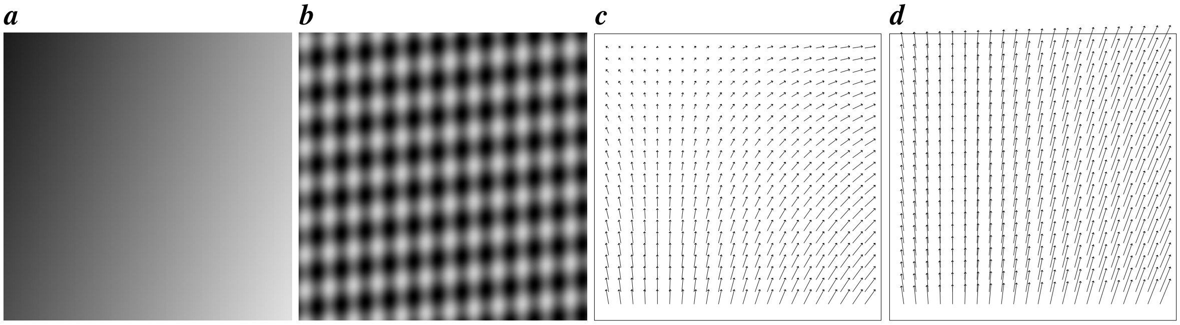


***Figure D.1**: Synthetic plane: **a** depth, **b** intensity, **c** XY-component and **d** XZ-component of the range flow.*

first, this yields plane coordinates. Then we perform the movement and determine where the point has moved based on its coordinates in the plane coordinate system. The two points are given by:

$$\begin{aligned} \boldsymbol{x} &= \boldsymbol{d} + x_p \hat{\boldsymbol{x}}_p + y_p \hat{\boldsymbol{y}}_p \,, & \text{(D.16)} \\ \boldsymbol{x}' &= \boldsymbol{d}' + x'_p \hat{\boldsymbol{x}}'_p + y'_p \hat{\boldsymbol{y}}'_p \,. & \text{(D.17)} \end{aligned}$$

The correct movement is then readily obtained from $\boldsymbol{f} = \boldsymbol{x}' - \boldsymbol{x}$. This velocity describes where a point in the current frame is moving. Range flow is typically evaluated for the central frame of a short sequence.

Figure D.1 shows an example of a synthetic plane using orthogonal projection. Sensor elements are spaced at $0.5\,mm$ in both directions. The plane parameters are $d = 150\,mm$, $\vartheta = 30^\circ$ and $\varphi = 20^\circ$. The thus obtained depth map is given in Fig. D.1a. In this case the intensity is chosen as a sinusoidal plaid pattern:

$$\begin{aligned} I(x_p, y_p) &= o + a_1 \sin(\frac{2\pi x'_p}{\lambda_1}) + a_2 \sin(\frac{2\pi y'_p}{\lambda_2}) \,, & \text{(D.18)} \\ x'_p &= x_p \cos(\alpha_1) + y_p \sin(\alpha_1) \,, & \text{(D.19)} \\ y'_p &= x_p \cos(\alpha_2) + y_p \sin(\alpha_2) \,. & \text{(D.20)} \end{aligned}$$

In Fig. D.1b we chose $o = 100,\ a_1 = a_2 = 50, \lambda_1 = 12\,mm,\ \lambda_2 = 16\,mm$ and $\alpha_1 = 0^\circ,\ \alpha_2 = 90^\circ$. There is an expansion rate of 1% on the plane and the complete plane moves with $(0.1\ 0.2\ 0.5)^T mm$. This results in a range flow field as shown in Fig. D.1 c,d. Of course one could use a different choice for the intensity, for example a radial sinus:

$$I(r, \theta) = o + a \sin(\frac{2\pi r}{\lambda}) \,. \qquad \text{(D.21)}$$

D.2 Sphere

The points on a sphere with radius R and center $(X_0\ Y_0\ Z_0)^T$ can be described by:

$$(X - X_0)^2 + (Y - Y_0)^2 + (Z - Z_0)^2 = R^2 \,. \qquad \text{(D.22)}$$

For orthogonal projection we directly obtain the 3D coordinates seen by a sensor element (x, y) as:

$$\begin{aligned} X(x, y) &= x \;, && \text{(D.23)} \\ Y(x, y) &= y \;, && \text{(D.24)} \\ Z(x, y) &= Z_0 - \sqrt{R^2 - (x - X_0)^2 - (y - Y_0)^2} \;. && \text{(D.25)} \end{aligned}$$

In case of perspective projection with focal length f we first recall that:

$$X = \frac{x}{f} Z \quad \text{and} \quad Y = \frac{y}{f} Z \;. \tag{D.26}$$

Insertion into (D.22) and sorting the terms yields:

$$\underbrace{(\frac{x^2}{f^2} + \frac{y^2}{f^2} + 1)}_{a} Z^2 + \underbrace{(-2\frac{x}{f}X_0 - 2\frac{y}{f}Y_0 - 2Z_0)}_{b} Z + \underbrace{(X_0^2 + Y_0^2 + Z_0^2 - R^2)}_{c} = 0 \;. \tag{D.27}$$

This is readily solved using the quadratic formula:

$$Z(x, y) = \frac{-b - \sqrt{b^2 - 4ac}}{2a} \;. \tag{D.28}$$

Where we chose the smaller solution as this corresponds to the side of the sphere the sensor sees. From the Z value X and Y can be recovered using (D.26).

We want to associate an intensity with the surface as well. This is best done using spherical coordinates defined by:

$$\begin{aligned} \theta &= \arccos(\frac{Z_0 - Z}{R}) \;, && \text{(D.29)} \\ \phi &= \arctan(\frac{Y_0 - Y}{X_0 - X}) \;. && \text{(D.30)} \end{aligned}$$

We can then describe the intensity at each sensor point by means of these angles $I = I(\theta, \phi)$. An example intensity function is given by:

$$I = \begin{cases} o & \text{if } \theta < 0.5^\circ \\ o + a\sin(\frac{2\pi\theta}{\lambda_\theta}) + a\sin(\frac{2\pi\phi}{\lambda_\phi}) & \text{else} \end{cases} \;. \tag{D.31}$$

Figure D.2 shows thus computed range data using an offset $o = 100$, an amplitude $a = 50$ and wavelengths of $\lambda_\theta = 1^\circ$ and $\lambda_\phi = 30^\circ$. In this example perspective projection with $f = 20\,mm$ and a sensor grid of $\Delta x = \Delta y = 0.05\,mm$ was used. The radius was $R = 150\,mm$ and the sphere centre $(X_0\,Y_0\,Z_0)^T = (0\,0\,300)^T mm$.

Growth is readily modelled by enlarging the radius of the sphere by a factor k. As the surface area is proportional to the squared radius the relative change in surface area is given by k^2. The expansion rate is then found to be:

$$e = (k^2 - 1) \cdot 100\ [\%] \;. \tag{D.32}$$

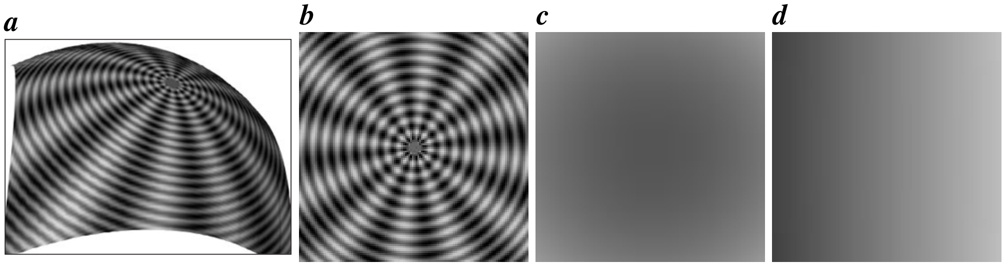


Figure D.2: *Synthetic sphere:* ***a*** *texture mapped range data,* ***b*** *intensity data,* ***c*** *depth (Z) data and* ***d*** *X data.*

Normally we specify the expansion rate and compute the radius change factor from that: $k = \sqrt{\frac{e}{100} + 1}$. For example does a growth of 1% result in a factor $k = \sqrt{1.01} \approx 1.004988$. Additionally to the expansion the centre of the sphere can be moved by a constant translation $(T_X\, T_Y\, T_Z)^T$. The resulting correct range flow field is then given as follows:

$$
\begin{aligned}
U &= T_X + (k-1)(X - X_0)\ , && \text{(D.33)}\\
V &= T_Y + (k-1)(Y - Y_0)\ , && \text{(D.34)}\\
W &= T_Z + (k-1)(Z - Z_0)\ . && \text{(D.35)}
\end{aligned}
$$

Bibliography

L. Alvarez, R. Deriche, J. Sánchez, and J. Weickert. Dense disparity map estimation respecting image derivatives: a pde and scale-space based approach. Technical report, INRIA, Sophia-Antipolis, France, January 2000a.

L. Alvarez, J. Weickert, and J. Sánchez. A scale-space approach to nonlocal optical flow calculations. In *Scale-Space Theories in Computer Vision*, pages 235–246, 1999.

L. Alvarez, J. Weickert, and J. Sánchez. Reliable estimation of dense optical flow fields with large displacements. *International Journal of Computer Vision*, 39:41–56, 2000b.

A. Bab-Hadiashar and D. Suter. Optic flow calculation using robust statistics. In *CVPR*, pages 988–993, Puerto Rico, 1997.

A. Bab-Hadiashar and D. Suter. Robust total least squares based optic flow computation. In *Asian Conference on Computer Vision*, Hong Kong, January 1998.

A. Bainbridge-Smith and R. G. Lane. Determining optical flow using a differential method. *Image and Vision Computing*, 15:11–22, 1997.

J. L. Barron, D. J. Fleet, and S. Beauchemin. Performance of optical flow techniques. *International Journal of Computer Vision*, 12(1):43–77, 1994.

J. L. Barron and A. Liptay. Measuring 3-d plant growth using optical flow. *Bioimaging*, 5:82–86, 1997.

J. L. Barron, A. Liptay, and H. Spies. Optical and range flow to measure 3d plant growth and motion. In *Image Vision Computing New Zealand*, pages 68–77, November 2000.

J. L. Barron and H. Spies. Quantitative regularized range flow. In *Vision Interface,*, pages 203–210, 2000.

E. Barth. The minors of the structure tensor. In *DAGM*, pages 221–228, Kiel, Germany, 2000.

S. Beauchemin and J. L. Barron. The frequency structure of one-dimensional occluding image signals. *PAMI*, 12(2):200–206, February 2000.

J. 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, Vienna, Austria, October 1995.

J. Bigün and G. H. Granlund. Optimal orientation detection of linear symmetry. In *ICCV*, pages 433–438, 1987.

M. J. Black. *Robust Incremental Optical Flow*. PhD thesis, Yale University, 1992.

M. J. Black. Explaining optical flow events with parameterized spatio-temporal models. In *CVPR'99*, Fort Collins, Colorado, June 1999.

M. J. Black and P. Anandan. A model for detection of motion over time. In *ICCV*, pages 33–37, Osaka, Japan, December 1990.

M. J. Black and P. Anandan. Robust dynamic motion estimation over time. In *CVPR*, pages 296–302, Maui, Hawaii, June 1991.

M. J. Black, D. J. Fleet, and Y. Yacoob. Robustly estimating changes in image appearance. *Computer Vision and Image Understanding*, 78(1):8–31, 2000.

M. J. Black and A. D. Jepson. Estimating optical flow in segmented images using variable-order parametric models with local deformations. *PAMI*, 18(10):972–986, October 1996.

F. Blais, J. Lecavalier, J. Domey, P. Boulanger, and J. Courteau. Application of the biris range sensor for wood volume measurement. Technical Report NRC 35033, National Research Council Canada, Ottawa, Canada, October 1992.

F. Blais and M. Rioux. Biris: A simple 3-d sensor. In *Optics, Illuminations, and Image Sensing for Machine Vision*, volume 728, pages 26–31, Cambridge, MA, October 1986.

I. N. Bronstein and K. A. Semendjajew. *Taschenbuch der Mathematik*. Verlag Harri Deutsch, Thun, 24 edition, 1989.

J. Casey. *Exploring Curvature*. Vieweg, Braunschweig, Germany, 1996.

K. Chaudhury, R. Mehrota, and C. Srinivasan. Detecting 3d flow. In *Conference on Robotics and Automation*, volume 2, pages 1073–1078, May 1994.

E. R. Davies. *Machine Vision: Theory, Algorithms, Practicalities*. Academic Press, 1990.

J. W. Demmel. The smallest perturbation of a submatrix which lowers the rank and constrained total least squares problems. *SIAM Journal on Numerical Analysis*, 24(1):199–206, February 1987.

S. Dineen. *Multivariate Calculus and Geometry*. Springer, London, 1994.

M. P. do Carmo. *Differential Geometry of Curves and Surfaces*. Prentice-Hall, Englewood Cliffs, NJ, 1976.

G. Farnebäck. Spatial domain methods for orientation and velocity estimation. Licentiate in Technology LIU-TEK-LIC-1999:13, Linköping University, March 1999.

G. Farnebäck. Fast and accurate motion estimation using orientation tensors and parametric motion models. In *ICPR*, volume 1, pages 135–139, Barcelona, Spain, September 2000a.

G. Farnebäck. Orientation estimation based on weighted projection onto quadratic polynomials. In G. Girod, G. Greiner, H. Niemann, and H.-P. Seidel, editors, *Vision, Modeling, and Visualization 2000*, pages 89–96, Saarbrücken, Germany, November 2000b.

O. Faugeras. *Three-Dimensional Computer Vision: A Geometric Viewpoint*. MIT Press, 1993.

C. Fermueller, R. Pless, and J. Aloimonos. Statistical biases in optic flow. In *CVPR'99*, Fort Collins, Colorado, June 1999.

G. Fischer. *Lineare Algebra*. Vieweg, Braunschweig, 11 edition, 1997.

D. J. Fleet. *Measurement of Image Velocity*. Kluwer Academic Publishers, Dordrecht, The Netherlands, 1992.

C. Garbe. Entwicklung eines Systems zur dreidimensionalen Particle Tracking Velocimetry mit Genauigkeitsuntersuchungen und Anwendung bei Messungen in einem Wind-Wellen Kanal. Master's thesis, University of Heidelberg, Heidelberg, Germany, 1998.

C. Garbe. *Heat Flow*. PhD thesis, University of Heidelberg, Heidelberg, Germany, 2001. in preparation.

C. Garbe, H. Haußecker, and B. Jähne. Measuring the sea surface heat flux and probability distribution of surface renewal events. In *Proc. Gas Transfer at Water Surfaces Symposium*, Miami Beach, Florida, 2000.

A. Gelb. *Applied Optimal Estimation*. M.I.T. Press, Cambridge, MA, 1974.

G. H. Golub and C. F. van Loan. An analysis of the total least squares problem. *SIAM Journal on Numerical Analysis*, 17(6):883–893, December 1980.

G. H. Golub and C. F. van Loan. *Matrix Computations*. The Johns Hopkins University Press, Baltimore and London, 3 edition, 1996.

G. H. Granlund and H. Knutsson. *Signal Processing for Computer Vision*. Kluwer Academic, Dordrecht, The Netherlands, 1995.

H. Gröning. *Monokulares 3D-Tracking und radiometrische Kalibrierung*. PhD thesis, University of Heidelberg, Heidelberg, Germany, 2001. in preparation.

C. Großmann and J. Terno. *Numerik der Optimierung*. Teubner, Stutgart, Germany, 2 edition, 1997.

W. Hackbusch. *Theorie und Numerik elliptischer Differentialgleichungen*. Teubner, Stuttgart, Germany, 1986.

F. R. Hampel, E. M. Ronchetti, P. J. Rousseeuw, and W. A. Stahel. *Robust Statistics: The Approach Based on Influence Functions*. John Wiley and Sons, New York, 1986.

R. M. Haralick. Digital step edges from zero crossing of second directional derivatives. *PAMI*, 6(1): 58–68, January 1984.

R. M. Haralick and L. Watson. A facet model for image data. *Computer Graphics and Image Processing*, 15(2):113–129, February 1981.

R. I. Hartley and P. F. Sturm. Triangulation. *Computer Vision and Image Understanding*, 68(2): 146–157, November 1997.

M. Harville, A. Rahimi, T. Darrell, G. Gordon, and J. Woodfill. 3d pose tracking with linear depth and brightness constraints. In *ICCV*, pages 206–213, 1999.

H. Haußecker and D. J. Fleet. Computing optical flow with physical models of brightness variation. In *CVPR'00*, volume 2, 2000.

H. Haußecker, C. Garbe, H. Spies, and B. Jähne. A total least squares for low-level analysis of dynamic scenes and processes. In *DAGM*, pages 240–249, Bonn, Germany, 1999. Springer.

H. Haußecker and B. Jähne. Ein Mehrgitterverfahren zur Bewegungssegmentierung in Bildfolgen. In *DAGM*, pages 24–31, 1993.

H. Haußecker and B. Jähne. A tensor approach for precise computation of dense displacement vector fields. In E. Paulus and F. M. Wahl, editors, *DAGM*, pages 199–208, Braunschweig, September 1997. Springer.

H. Haußecker and H. Spies. Motion. In B. Jähne, H. Haußecker, and P. Geißler, editors, *Handbook of Computer Vision and Applications*, volume 2, chapter 13. Academic Press, 1999.

G. E. Healey and R. Kondepudy. Radiometric ccd camera calibration and noise estimation. *PAMI*, 16 (3):267–276, March 1994.

B. K. P. Horn and J. Harris. Rigid body motion from range image sequences. *CVGIP*, 53(1):1–13, January 1991.

B. K. P. Horn and B. Schunk. Determining optical flow. *Artificial Intelligence*, 17:185–204, 1981.

P. J. Huber. *Robust Statistics*. John Wiley and Sons, New York, 1981.

G. Häusler. Three-dimensional sensors - potentials and limitations. In B. Jähne, H. Haußecker, and P. Geißler, editors, *Handbook of Computer Vision and Applications*, volume 1, pages 485–506. Academic Press, 1999.

B. Jähne. *Digital Image Processing*. Springer, Berlin, Germany, 3 edition, 1995.

B. Jähne. *Practical Handbook on Image Processing for Scientific Applications*. CRC Press, Boca Raton, Florida, 1996.

B. Jähne, H. Haußecker, H. Scharr, H. Spies, D. Schmundt, and U. Schurr. Study of dynamical processes with tensor-based spatiotemporal image processing techniques. In *ECCV*, pages 322–336. Springer, 1998.

B. Jähne, H. Scharr, and S. Körkel. Principles of filter design. In B. Jähne, H. Haußecker, and P. Geißler, editors, *Handbook of Computer Vision and Applications*, volume 2, pages 125–151. Academic Press, 1999.

W. Kühnel. *Differentialgeometrie*. Vieweg, Braunschweig/Wiesbaden, Germany, 1999.

N. Kirchgeßner. Voruntersuchungen zur 3d-Wuchsanalyse von Pflanzenblättern und die Modellierung der Blattadern. Master's thesis, University of Heidelberg, Heidelberg, Germany, 1999.

N. Kirchgeßner, H. Scharr, and U. Schurr. 3d-Modellierung von Pflanzenblättern mittels eines Depth-from-Motion Verfahrens. In *DAGM*, pages 381–388, Kiel, Germany, 2000.

N. Kirchgeßner, H. Spies, H. Scharr, and U. Schurr. Root growth analysis in physiological coordinates. In *International Conference on Image Analysis and Processing (ICIAP'01)*, Palermo, Italy, September 2001. accepted.

G. J. Klein and R. H. Huesman. A 3d optical approach to addition of deformable pet volumes. In *IEEE Nonrigid and Articulated Motion Workshop*, pages 136–143, June 1997.

R. Klette, A. Koschan, and K. Schlüns. *Computer Vision Räumliche Information aus digitalen Bildern*. Vieweg, Braunschweig, Germany, 1996.

H. Kälviäinen, E. Oja, and L. Xu. Randomized hough transform applied to translational and rotational motion analysis. In *ICPR*, pages 672–675, 1992.

H. Knutsson and C.-F. Westin. Normalized and differential convolution: Methods for interpolation and filtering of incomplete and uncertain data. In *CVPR*, pages 515–516, New York City, June 1993.

T. Lindeberg. *Scale-Space Theory in Computer Vision*. Kluwer Academic Publishers, Dordrecht, The Netherlands, 1994.

Y. Liu and M. A. Rodrigues. Correspondenceless motion estimation from range images. In *ICCV*, 1999.

F. Loranger, D. Laurendeau, and R. Houde. A fast and accurate 3-d rangefinder using the biris technology: the trid sensor. In *Recent Advances in 3-D Digital Imaging and Modeling*, pages 51–58, Ottawa, Ontario, 1997.

B. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *DARPA Image Understanding Workshop*, pages 121–130, 1981.

L. Lucchese, G. Doretto, and G. M. Cortelazzo. Frequency domain estimation of 3-d rigid motion based on range and intensity data. In *Recent Advances in 3-D Digital Imaging and Modeling*, pages 107–112, Ottawa, Ontario, 1997.

T. Luhmann. *Nahbereichsphotogrammetrie : Grundlagen, Methoden und Anwendungen*. Wichmann, Heidelberg, 2000.

L. Matthies, T. Kanade, and R. Szeliski. Kalman filter based algorithms for estimating depth from image sequences. *International Journal of Computer Vision*, 3(3):209–238, 1989.

A. Mc Ivor. Robust 3d surface property estimation. In *Asian Conference on Computer Vision*, volume 2, pages 275–279, December 1995.

P. Meer, D. Mintz, and A. Rosenfeld. Robust regression methods for computer vision. *International Journal of Computer Vision*, 6(1):59–70, 1991.

R. Mester. Orientation estimation: conventional techniques and a new non-differential method. In *European Signal Processing Conference (EUSIPCO'2000)*, Tampere, Finland, September 2000.

M. Mühlich and R. Mester. The role of total least squares in motion analysis. In *ECCV*, pages 305–321, Freiburg, Germany, 1998.

M. Mühlich and R. Mester. Subspace methods and equilibration in computer vision. Technical Report XP-TR-C-21, Institute for Applied Physics, Goethe-Universitaet, Frankfurt, Germany, November 1999.

A. Mitiche and P. Bouthemy. Computation and analysis of image motion: A synopsis of current problems and methods. *International Journal of Computer Vision*, 19(1):29–55, 1996.

H.-H. Nagel. Optical flow estimation and the interaction between measurement errors at adjacent pixel positions. *International Journal of Computer Vision*, 15:271–288, 1995.

H.-H. Nagel and W. Enkelmann. An investigation of smoothness constraints for the estimation of dispalcement vector fields from image sequences. *PAMI*, 8(5):565–593, September 1986.

H.-H. Nagel and A. Gehrke. Spatiotemporal adaptive estimation and seqmentation of of-fields. In *ECCV*, pages 87–102, Freiburg, Germany, 1998.

P. Nesi, A. Del Bimbo, and D. Ben-Tzi. A robust algorithm for optical flow estimation. *Computer Vision and Image Understanding*, 62(1):59–68, 1995.

O. Nestares, D. J. Fleet, and D. Heeger. Likelihood functions and confidence bounds for total-least-squares problems. In *CVPR'00*, volume 1, 2000.

L. Ng and V. Solo. Errors-in-variables modelling in optical flow problems. In *ICASSP*, pages 2772–2776, 1998.

E. P. Ong and M. Spann. Robust optical flow computation based on least-median-of-squares regression. *International Journal of Computer Vision*, 31(1):51–82, February 1999.

M. Otte and H.-H. Nagel. Optical flow estimation: Advances and comparisons. In J.-O. Eklundh, editor, *ECCV*, pages 51–60, 1994.

P. Perona and J. Malik. Scale space and edge detection using anisotropic diffusion. *PAMI*, 12:629–639, July 1990.

W. Press, S. Teukolsky, W. Vetterling, and B. Flannery. *Numerical Recipes in C*. Cambridge University Press, Cambridge, MA, 2 edition, 1992.

M. Proesmans, L. Van Gool, E. Pauwels, and A. Oosterlinck. Determination of optical flow and its discontinuities using non-linear diffusion. In *ECCV*, pages 295–304, 1994.

J. A. Rice. *Mathematical Statistics and Data Analysis*. Duxbury Press, 2 edition, 1995.

M. Rioux and F. Blais. Compact three-dimensional camera for robotic applications. *Journal of the Optical Society of America A*, 3(9):1518–1521, September 1986.

B. Sabata and J. K. Aggarwal. Estimation of motion from a pair of range images: A review. *CVGIP*, 54(3):309–324, 1991.

H. Scharr. *Optimale Operatoren in der Digitalen Bildverarbeitung*. PhD thesis, University of Heidelberg, Heidelberg, Germany, 2000.

H. Scharr, S. Körkel, and B. Jähne. Numerische Isotropieoptimierung von FIR-Filtern mittels Querglättung. In F. M. Wahl and E. Paulus, editors, *DAGM*, pages 199–208, 1997.

H. Scharr and J. Weickert. An anisotropic diffusion algorithm with optimized rotation invariance. In *DAGM*, pages 460–467, Kiel, Germany, September 2000.

Y. Y. Schechner and N. Kiryati. Depth from defocus vs. stereo: How different really are they? *International Journal of Computer Vision*, 39(2):141–162, 2000.

F. Scheck. *Mechanik: Von den Newtonschen Gesetzen zum deterministischen Chaos*. Springer, Berlin, Germany, 2 edition, 1990.

D. Schmundt. Voruntersuchung der Einsatzmöglichkeiten digitaler Bildverarbeitung zur Analyse von Transportvorgängen und Wachstumsprozessen in Pflanzen. Master's thesis, University of Heidelberg, 1995.

D. Schmundt. *Development of an Optical Flow Based System for the Precise Measurement of Plant Growth*. PhD thesis, University of Heidelberg, 1999.

D. Schmundt, M. Stitt, B. Jähne, and U. Schurr. Quantitative analysis of local growth rates of dicot leaves at high temporal and spatial resolution, using image sequence analysis. *Plant Journal*, 16: 505–514, 1998.

C. Schnörr. Determining optical flow for irregular domains by minimizing quadratic functionals of a certain class. *International Journal of Computer Vision*, 6(1):25–38, 1991.

C. Schnörr. On functionals with greyvalue-controlled smoothness terms for determining optical flow. *PAMI*, 15(10):1074–1079, October 1993.

C. Schnörr. Variational methods for adaptive image smoothing and segmentation. In B. Jähne, H. Haußecker, and P. Geißler, editors, *Handbook of Computer Vision and Applications*, volume 2, pages 451–484. Academic Press, 1999.

C. Schnörr and J. Weickert. Variational image motion computation: Theoretical framework, problems and perspectives. In *DAGM 2000*, pages 476–487, 2000.

B. Schunk. The image flow constraint equation. *Computer Vision, Graphics and Image Processing*, 35:20–46, 1986.

R. Schwarte, H. Heinol, B. Buxbaum, T. Ringbeck, Z. Xu, and K. Hartmann. Principles of three-dimensional imaging techniques. In B. Jähne, H. Haußecker, and P. Geißler, editors, *Handbook of Computer Vision and Applications*, volume 1, pages 463–484. Academic Press, 1999.

D.-G. Sim and R.-H. Park. Robust reweighted map motion estimation. *PAMI*, 20(4):353–365, April 1998.

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 Transactions on Medical Imaging*, 13(2):386–397, 1994.

H. Spies, H. Haußecker, B. Jähne, and J. L. Barron. Differential range flow estimation. In *DAGM*, pages 309–316, Bonn, Germany, September 1999.

H. Spies and B. Jähne. A general framework for image sequence analysis. In *Fachtagung Informationstechnik*, pages 125–132, Magdeburg, Germany, March 2001. Otto-von-Guericke-Universität Magdeburg.

H. Spies, B. Jähne, and J. L. Barron. Dense range flow from depth and intensity data. In *ICPR*, pages 131–134, Barcelona, Spain, September 2000a.

H. Spies, N. Kirchgeßner, H. Scharr, and B. Jähne. Dense structure estimation via regularised optical flow. In G. Girod, G. Greiner, H. Niemann, and H.-P. Seidel, editors, *VMV '2000,*, pages 57–64, Saarbrücken, Germany, November 2000b. Aka GmbH, Berlin.

H. Spies and H. Scharr. Accurate optical flow in noisy image sequences. In *ICCV*, Vancouver, Canada, July 2001. accepted.

C. V. Stewart. Bias in robust estimation caused by discontinuities and multiple structures. *PAMI*, 19 (8):818–833, August 1997.

J. Stoer. *Numerische Mathematik 1*. Springer, Berlin, Germany, 8 edition, 1999.

P. F. Sturm and S. J. Maybank. On plane-based camera calibration: A general algorithm, singularities, applications. In *CVPR'99*, Fort Collins, Colorado, June 1999.

R. Szeliski. Estimating motion from sparse range data without correspondence. In *ICCV*, pages 207–216, 1988.

D. Terzopoulos. Regularization of inverse visual problems involving discontinuities. *PAMI*, 8(4): 413–424, 1986.

D. Terzopoulos. The computation of visible-surface representations. *PAMI*, 10(4):417–438, July 1988.

P. Torr. *Motion Segmentation and Outlier Detection*. PhD thesis, University of Oxford, 1995.

L. Tsap, D. 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): 997–1007, 1999.

L. Tsap, D. Goldgof, and S. Sarkar. Multiscale combination of physically-based registration and deformation modeling. In *CVPR*, volume 2, pages 422–429, Hilton Head, SC, June 2000.

S. Van Huffel and J. Vandewalle. *The Total Least Squares Problem: Computational Aspects and Analysis*. Society for Industrial and Applied Mathematics, Philadelphia, 1991.

S. Vedula, S. Baker, P. Rander, R. Collins, and T. Kanade. Three-dimensional scene flow. In *ICCV*, pages 722–729, Pittsbrugh, PA, September 2000.

F. M. Wahl. A coded light approach for depth map acquisition. In *DAGM*, pages 12–17, Paderborn, September 1986.

A. Walter. *Räumliche und zeitliche Wachstumsmuster in Wurzeln und Blättern dikotyler Pflanzen*. PhD thesis, University of Heidelberg, Heidelberg, Germany, 2001.

A. Walter and U. Schurr. Diurnal variation of leaf growth in ricinus communis and its relationship to tensile forces in the midrib - an analysis with digital image processing. In *Conference of Plant Biomechanics*, 2000a.

A. Walter and U. Schurr. Spatial variability of leaf development, growth and function. In B. Marshall and J. Roberts, editors, *Leaf Development and Canopy Growth*. Sheffield Academic Press, 2000b.

A. Walter, H. Spies, S. Wilms, R. Küsters, N. Kirchgeßner, and U. Schurr. Growing straight? - growth dynamics in root tips is quantified automatically and with a high spatio-temporal resolution by means of digital image sequence processing. *Journal of Experimental Botany*, 2001. submitted.

J. Weber and J. Malik. Robust computation of optical flow in a multi-scale differential framework. In *ICCV*, pages 12–20, Berlin, Germany, May 1993.

J. Weber and J. Malik. Robust computation of optical flow in a multi-scale differential framework. *International Journal of Computer Vision*, 14(1):67–81, 1995.

J. Weickert. *Anisotropic Diffusion in Image Processing*. Teubner, Stuttgart, Germany, 1998.

C.-F. Westin. *A Tensor Framework for Multidimensional Signal Processing*. PhD thesis, Linköping University, Linköping, Sweden, 1994.

C. E. Willert and M. Gharib. Three-dimensional particle imaging with a single camera. *Experiments in Fluids*, 12:353–358, 1992.

G. Wiora. *Optische 3D-Messtechnik: Präzise Gestaltvermessung mit einem erweiteren Streifenprojektionsverfahren*. PhD thesis, University of Heidelberg, Heidelberg, Germany, 2001.

H. Wolf and K. Wolf. Neue Hard- und Softwareentwicklungen bei ABW. In *4. ABW-Workshop Optische 3D-Formerfassung*. Technische Akademie Esslingen, January 1997.

Y. Xiong and S. A. Shafer. Dense structure from a dense optical flow sequence. *Computer Vision and Image Understanding*, 69(2):222–245, 1998.

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):82–89, January 1993.

W. Yu, K. Daniilidis, S. Beauchemin, and G. Sommer. Detection and characterization of multiple motion points. In *CVPR'99*, Fort Collins, Colorado, June 1999.

Y. Zhang and C. Kambhamettu. Integrated 3d scene flow and structure recovery from multiview image sequences. In *CVPR*, volume 2, 2000.

Z. Zhang. A flexible new technique for camera calibration. *PAMI*, 22(11):1330–1334, November 2000.

Acknowledgements

Of the numerous people that contributed to the success of the presented thesis I can only name but a few. First of all my thanks go to Prof. Bernd Jähne for supervising this work. I also want to thank Prof. Josef Bille that he agreed to act as second referee.

I'm grateful to Prof. John Barron for the warm welcome at the University of Western Ontario for the first year of my PhD, not to mention all the Guinness. I acknowledge the support of Dr. Albert Liptay from Agriculture Canada for letting me use their Biris sensor.

My thanks go to Dr. Uli Schurr for supervising the botanical aspects of this work. He and his group at the Institute of Botany made a truly interdisciplinary project possible. Especially the cooperation with Achim and Stefan was very inspiring.

For providing a stimulating research environment I thank all the members of the image processing research group at the Interdisciplinary Center for Scientific Computing and the Institut of Environmental Physics at the University of Heidelberg. In particular I want to thank Christoph, Hanno, Hermann, Norbert and Tobias for our successful work together.

Finally I want to express my gratitude to my wife My for her patience and support during the sometimes stressful times.