Estimation Of 3D Object Structure, Motion And Rotation Based On 4D Affine Optical Flow Using A Multi-Camera Array

Tobias Schuchert[1,2] and Hanno Scharr[2]

[1] Fraunhofer Institute of Optronics, System Technologies and Image Exploitation,
Karlsruhe, Germany
tobias.schuchert@iosb.fraunhofer.de

[2] Institute for Chemistry and Dynamics of the Geosphere, ICG-3: Phytosphere,
Forschungszentrum Jülich, Germany
h.scharr@fz-juelich.de

Abstract. In this paper we extend a standard affine optical flow model to 4D and present how affine parameters can be used for estimation of 3D object structure, 3D motion and rotation using a 1D camera grid. Local changes of the projected motion vector field are modelled not only on the image plane as usual for affine optical flow, but also in camera displacement direction, and in time. We identify all parameters of this 4D fully affine model with terms depending on scene structure, scene motion, and camera displacement. We model the scene by planar, translating, and rotating surface patches and project them with a pinhole camera grid model. Imaged intensities of the projected surface points are then modelled by a brightness change model handling illumination changes. Experiments demonstrate the accuracy of the new model. It outperforms not only 2D affine optical flow models but range flow for varying illumination. Moreover we are able to estimate surface normals and rotation parameters. Experiments on real data of a plant physiology experiment confirm the applicability of our model.

1 Introduction

Object structure and motion estimation from camera image sequences is a typical and well explored computer vision topic and many different solutions exist for different application prerequisites. We target at a typical plant physiology lab situation (see e.g. [1]), where e.g. growth, i.e. divergence of the motion vector field or curvature production in terms of spatial derivatives of the rotation vector field, of plant organs are parameters of interest. In order to analyse derivatives of the motion field, motion and structure of plant organs – here leaves of seedlings and small plants – need to be measured in high spatial (sub-millimeter) and temporal resolution (several minutes). Highest accuracy is thus a prerequisite here, as derivatives of the motion field are the final signal of interest and rigid motion of leaves is much larger than motion due to e.g. growth. Such measurements help unravelling bio-chemical processes underlying plant growth (see e.g. [2]) and thus give hints for seed, feed, and food production or plant breeding. However, calculation time is less of an issue, as analyses may be calculated off-line.

A typical lab setup uses a single camera on a moving stage looking downward on the plant, instead of using multiple cameras. The advantage of such a setup is that the moving stage allows to take images from many equidistant camera positions, typically at several mm or even sub-mm distances depending on object size. Further calibration needs only be done for a single camera and the camera may be moved away when not needed as plants should not be shaded by measurement equipment. One loop through all camera positions takes much shorter (seconds) than time between two acquisitions at the same position (minutes). Consequently we may regard the acquired data as if it came from a synchronized, fine spaced 1D grid of cameras. This camera grid equidistantly samples a 4D space spanned by the sensor coordinates x and y, camera position s and time t.

Such 4D data has already been exploited in literature (e.g. [3, 4]). There 4D optical flow with affine components is calculated and all components are interpreted in terms of 3D translation, 3D position and surface normals of the imaged object. Good performance is reported for translating objects, however rotating objects lead to severe errors. Here we present a solution to this problem by modelling rotation and extending the affine flow to s- and t-derivatives of the flow-field. The model from [3, 4] is valid for instantaneously moving cameras observing moving surfaces. This is unlike preceding work (e.g. [5–8]) where either an observed surface moves, or a camera, but not both. The patch-based affine model also differs from other frameworks for motion and stereo analysis, where point-based models are applied (e.g. [9–12]) or global minimization in 3D scene space is addressed (e.g. [13, 14]).

1.1 Approach

The standard 2D affine optical flow model (cmp. e.g. [15])

$$\nabla I \left[\begin{pmatrix} u_x \\ u_y \end{pmatrix} + \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} \right] + I_t = 0 \qquad (1)$$

defines parameters in image coordinates, i.e. flow parameters. Here, the meaning of the flow parameters $u_.$ and $a_{..}$ will be explained in world coordinates and parameters of imaged moving surface patches.

Following [4] an image sequence may be interpreted as data in a 3D space where a brightness change constraint defines a linear model for intensity changes due to apparent local object motion. This motion is called optical flow. When the data is acquired by a single fixed camera, i.e. x-y-t-space, visible motion may be explained by object motion. When acquired by a moving camera looking at a fixed scene, i.e. x-y-s-space, displacements (then usually called disparities) are anti-proportional to local depth. This is known as structure from camera motion (e.g. [16]). Here, we interpret the camera position s as additional dimension of the data. Hence all image sequences acquired by a 1D camera grid can be combined to sample a 4D-volume in x-y-s-t-space. Brightness changes in this space are modelled as total differential of the intensity data.

The presented 4D fully affine optical flow model can be seen as an extended version of (1). But here, affine modelling not only covers linear changes in local pixel coordinates Δx, and Δy, but also in camera motion direction and time, i.e. additional Δs and Δt terms. In order to explain 3D structure and 3D motion in world coordinates by the estimated flow parameters, 3D dynamic surfaces patches are projected into the image by a pinhole camera (cmp. Sec. 2). A crucial point here is the correct handling of neighbor locations. We model it by back-projection of the pixel grid to the surface in the scene (see Secs. 2.4 and 2.5). A detailed derivation can be found in Section 2.

In order to evaluate the model we use a parameter estimation procedure as proposed in [4]. It is a total least squares (TLS) estimation scheme ideally suited for estimation when Gaussian noise is present. No robust statistics or regularization terms are applied. Such terms may conceal model errors and therefore are not suitable for model evaluation. Adaptations needed here are presented in Section 3.

Quantitative experiments (Section 4) use synthetic data with ground truth available. For systematic evaluation of accuracies we use pinhole-projected 32bit-float sinusoidal patterns suppressing otherwise unavoidable quantization noise. For more realistic scenes with ground truth available we use simple geometric structures moving in a known way rendered by POV-Ray [17] in 8bit-integer accuracy. We compare motion results to range flow [18, 3] in order to give an intuition of the accuracies achievable using a simple TLS estimator. In contrast to our model, range flow needs depth information as input and estimates 3D translation only.

An experiment with real data showing a small tobacco leaf visually demonstrates the increased accuracy compared to other methods. Only the new method yields plausible results.

1.2 Contribution

The current paper is an extension to a series of papers [3, 4]. Our contributions are the following: (1) Derivation of 4D affine flow parameters (Δs- and Δt-terms). (2) Back-projection of the pixel grid to an imaged surface respecting camera position and time. (3) Modelling rotational motion of surface patches. (4) A thorough model evaluation focusing on rotation effects.

Although our scheme can readily be used to estimate motion and shape deformations of plant leaves, we do not aim at presenting a final *method* yet. In our view a method not only needs accurate modelling but also needs well adapted discretization, estimation schemes, regularization etc. Here we focus on modelling, only.

2 Model derivation

2.1 Surface patch model

Following [4] we model a surface patch at world coordinate position (X_0, Y_0, Z_0) as a function of time t by

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{pmatrix} X_0 + U_X t + \Delta X \\ Y_0 + U_Y t + \Delta Y \\ Z_0 + U_Z t + Z_X \Delta X + Z_Y \Delta Y \end{pmatrix} \tag{2}$$

where Z_X and Z_Y are surface slopes in X- and Y-direction for time $t = 0$ and $\boldsymbol{U} = (U_X, U_Y, U_Z)$ is the velocity of the patch. The surface normal is then $\boldsymbol{n} = (Z_X, Z_Y, -1)$.

2.2 Rotation

We define rotation of a surface patch by angular velocity vector $\boldsymbol{\Omega} = (\Omega_X, \Omega_Y, \Omega_Z)^{\mathrm{T}}$ located at its central point $\boldsymbol{X}_0 = (X_0, Y_0, Z_0)^{\mathrm{T}}$. Velocity $\boldsymbol{U}$ of points on the surface patch is then determined by

$$\boldsymbol{U} = \boldsymbol{N} + \boldsymbol{\Omega} \times \Delta \boldsymbol{X} \tag{3}$$

with translational velocity $\boldsymbol{N} = (N_X, N_Y, N_Z)^T$, distance to the rotation center $\Delta \boldsymbol{X}$ and angular velocity $\boldsymbol{\Omega}$.

Equation (3) defines rotation around the surface patch center. For general rotational motion this is not sufficient, as the true center of rotation may not coincide with the patch center. This leads to accelerated motion of the patch center

$$\boldsymbol{U} = \boldsymbol{N} + \boldsymbol{A} t + \boldsymbol{\Omega} \times \Delta \boldsymbol{X}. \tag{4}$$

with acceleration $\boldsymbol{A}$. We address constant acceleration only, whereas acceleration coming from rotation is non-constant. This could be modelled by estimation of the true rotation center introducing 3 additional parameters analogue to (3).

2.3 Projective Camera Model

We use pinhole cameras at world coordinate positions $(s, 0, 0)$, looking into Z-direction

$$\begin{pmatrix} x \\ y \end{pmatrix} = \frac{f}{Z} \begin{pmatrix} X - s \\ Y \end{pmatrix} \tag{5}$$

Sensor coordinates x, y are aligned with world coordinates X, Y. Camera position space is sampled equidistantly using a 1D camera grid. We combine data acquired by all cameras into one 4D data set equidistantly sampling the continuous intensity function $I(x, y, s, t)$.

2.4 Pixel-centered view

Parameter estimation at a 4D pixel $\boldsymbol{x_0} = (x_0, y_0, s_0, t_0)$ is done using the acquired image data $I(\boldsymbol{x}) := I(x, y, s, t)$ in a local neighborhood Λ, with $\boldsymbol{x} := (x, y, s, t)^T$. Consequently we need to know surface position $\boldsymbol{X}$ for each 4D pixel, i.e. $\boldsymbol{X}(\boldsymbol{x})$. Using (5) we know

$$\begin{pmatrix} X(\boldsymbol{x}) \\ Y(\boldsymbol{x}) \end{pmatrix} = \frac{Z(\boldsymbol{x})}{f} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} s \\ 0 \end{pmatrix} \tag{6}$$

In order to derive an expression for $Z(\boldsymbol{x})$ we fit a tangent plane with surface normal $\boldsymbol{n} = (Z_X, Z_Y, -1)^T$ to the point $\boldsymbol{X}(\boldsymbol{x})$. The intersection of this tangent plane with the Z-axis is then $Z(0,0,0,t)$, and $Z(0,0,0,t) = Z(\mathbf{0}) + Z_t t$ for a constantly translating plane, where $\mathbf{0} := (0,0,0,0)^T$. Consequently $Z(\boldsymbol{x})$ can be expressed as

$$Z(\boldsymbol{x}) = Z(\mathbf{0}) + Z_X X(\boldsymbol{x}) + Z_Y Y(\boldsymbol{x}) + Z_t t \Leftrightarrow Z(\boldsymbol{x}) = \frac{Z(\mathbf{0}) + Z_X s + Z_t t}{1 - Z_X \frac{x}{f} - Z_Y \frac{y}{f}} \tag{7}$$

where we used (6) to substitute $X(\boldsymbol{x})$ and $Y(\boldsymbol{x})$. Combining (6) and (7) yields

$$\boldsymbol{X}(\boldsymbol{x}) = \tfrac{Z(\mathbf{0}) + Z_X s + Z_t t}{f - Z_X x - Z_Y y} \begin{pmatrix} x \\ y \\ f \end{pmatrix} + \begin{pmatrix} s \\ 0 \\ 0 \end{pmatrix} . \tag{8}$$

Equation (8) extends the formulation in [4], where $\boldsymbol{X}$ only depends on local image coordinates x and y.

2.5 Projecting the pixel grid to the surface

A pixel $\boldsymbol{x}$ in the local neighborhood Λ used for parameter estimation is given by $\boldsymbol{x} = \boldsymbol{x_0} + \Delta\boldsymbol{x} = (x_0 + \Delta x, y_0 + \Delta y, s_0 + \Delta s, t_0 + \Delta t)^T$. The surface patch center $\boldsymbol{X_0}$ by definition is the projection of the neighborhood center point $\boldsymbol{x_0}$ to the surface. Thus neighbor points of $\boldsymbol{X_0}$ on the surface given by $\Delta\boldsymbol{X} = (\Delta X, \Delta Y, \Delta Z)$ are projections of the cameras pixel grids to the surface. We need to derive $\Delta\boldsymbol{X}(\Delta\boldsymbol{x})$. To stay on the surface, we model $\Delta Z = Z_X \Delta X + Z_Y \Delta Y$, cmp. (2). From (2) we know

$$\Delta\boldsymbol{X} = \boldsymbol{X} - \boldsymbol{X}_0 - \boldsymbol{U} t \tag{9}$$

where $\boldsymbol{X}$ is a point on the surface at a given point in time t, $\boldsymbol{X}_0$ is the surface patch center at time $t_0 = 0$, and $\Delta\boldsymbol{X}$ is the distance between $\boldsymbol{X}$ and the point $\boldsymbol{X}_0 + \boldsymbol{U} t$ where the patch center moved to. The distance between $\boldsymbol{X}$ and $\boldsymbol{X_0}$ can expressed by the linearisation

$$\boldsymbol{X} - \boldsymbol{X}_0 = \frac{\partial \boldsymbol{X}}{\partial x} \Delta x + \frac{\partial \boldsymbol{X}}{\partial y} \Delta y + \frac{\partial \boldsymbol{X}}{\partial s} \Delta s + \frac{\partial \boldsymbol{X}}{\partial t} \Delta t \, . \tag{10}$$

Partial derivatives of $\boldsymbol{X}(\boldsymbol{x})$ can be derived from (8).

2.6 Brightness Change Model

Point correspondences in our multi-dimensional data set are derived via an estimation analogue to common structure-from-camera-motion or optical-flow methods. Thus we employ a differential brightness constraint modelling intensity changes $\mathrm{d}I$ of a surface element for the 4D data set $I(x, y, s, t)$. $\mathrm{d}I$ equals

$$I_x\mathrm{d}x + I_y\mathrm{d}y + I_s\mathrm{d}s + I_t\mathrm{d}t = I(g_1 + g_{1,x}\Delta x + g_{1,y}\Delta y + g_2 t)\mathrm{d}t \tag{11}$$

We denote $I_* = \frac{\partial I}{\partial *}$ for derivatives of the image intensities I. In the following we use notation $g = (g_1 + g_{1,x}\Delta x + g_{1,y}\Delta y + g_2 t)$. The left hand side of Equation (11) is derived from dI by chain rule. The right hand side of (11) models spatially varying brightness changes. It boils down to a local spatio-temporal series expansion of varying illumination times bidirectional reflectance distribution function (BRDF). We refer to [3] for a detailed derivation.

2.7 A 4D-Affine Model

We combine the above equations in order to derive the new 4D optical flow model and a geometric interpretation of its parameters. Again following [4] we project the moving surface patch model to the sensor plane by substituting (2) in (5) and calculate the differentials $\mathrm{d}x$ and $\mathrm{d}y$ for a *given surface location* (i.e. for constant ΔX and ΔY).

$$\begin{pmatrix} \mathrm{d}x \\ \mathrm{d}y \end{pmatrix} = \frac{f}{Z} \begin{pmatrix} (U_X - U_Z\frac{x}{f})\mathrm{d}t - \mathrm{d}s \\ (U_Y - U_Z\frac{y}{f})\mathrm{d}t \end{pmatrix} \tag{12}$$

From (2) we know that Z depends on the unknown ΔX and ΔY: $Z = Z_0 + U_Z\Delta t + Z_X\Delta X + Z_Y\Delta Y$. We therefore rephrase f/Z using (9) and (10)

$$f/Z = -\nu - b_1\Delta x - b_2\Delta y - b_3\Delta s - b_4\Delta t \tag{13}$$

$$\begin{aligned} \text{with} \quad & \nu = -\frac{f}{Z_0}, \quad b_1 = \frac{Z_X}{Z_0 c}, \quad b_2 = \frac{Z_Y}{Z_0 c}, \quad b_3 = \frac{f}{Z_0}\frac{Z_X}{Zc}, \quad b_4 = \frac{f}{Z_0}\frac{Z_t}{Zc} \\ \text{and} \quad & c = 1 - Z_X\frac{x}{f} - Z_Y\frac{y}{f}\ . \end{aligned} \tag{14}$$

The remaining substitution steps are then as follows: first in (4) substitute $\boldsymbol{\Delta X}$ by (9)–(10), then in (12) substitute $\boldsymbol{U}$ by (4) and $\frac{f}{Z}$ by (13). Finally substitute in the brightness change model (11) $\mathrm{d}x$ and $\mathrm{d}y$ by (12). Ignoring higher order terms yields representations of the elements of the 4 dimensional optical flow model

$$\begin{aligned} &\nabla I \left[\begin{pmatrix} u_x\mathrm{dt} + \nu\mathrm{ds} \\ u_y\mathrm{dt} \end{pmatrix} + \begin{pmatrix} a_{11}\mathrm{dt} + b_1\mathrm{ds} & a_{12}\mathrm{dt} + b_2\mathrm{ds} & a_{13}\mathrm{dt} + b_3\mathrm{ds} & a_{14}\mathrm{dt} + b_4\mathrm{ds} \\ a_{21}\mathrm{dt} & a_{22}\mathrm{dt} & a_{23}\mathrm{dt} & a_{24}\mathrm{dt} \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta y \\ \Delta s \\ \Delta t \end{pmatrix} \right] \\ &+I_s\mathrm{ds} + I_t\mathrm{dt} = Ig\mathrm{dt} \end{aligned} \tag{15}$$

with parameters

$$\begin{aligned}
u_x &= -\nu \left(N_X - \tfrac{x_0}{f} N_Z\right) \\
u_y &= -\nu \left(N_Y - \tfrac{y_0}{f} N_Z\right) \\
a_{11} &= -\nu \left[\Omega_Y \tfrac{\partial Z}{\partial x} - \Omega_Z \tfrac{\partial Y}{\partial x} - \tfrac{x_0}{f} \left(\Omega_X \tfrac{\partial Y}{\partial x} - \Omega_Y \tfrac{\partial X}{\partial x}\right) - \tfrac{N_Z}{Z_0}\right] & -b_1 \left(N_X - \tfrac{x_0}{f} N_Z\right) \\
a_{12} &= -\nu \left[\Omega_Y \tfrac{\partial Z}{\partial y} - \Omega_Z \tfrac{\partial Y}{\partial y} - \tfrac{x_0}{f} \left(\Omega_X \tfrac{\partial Y}{\partial y} - \Omega_Y \tfrac{\partial X}{\partial y}\right)\right] & -b_2 \left(N_X - \tfrac{x_0}{f} N_Z\right) \\
a_{13} &= -\nu \left[\Omega_Y \tfrac{\partial Z}{\partial s} - \Omega_Z \tfrac{\partial Y}{\partial s} - \tfrac{x_0}{f} \left(\Omega_X \tfrac{\partial Y}{\partial s} - \Omega_Y \tfrac{\partial X}{\partial s}\right)\right] & -b_3 \left(N_X - \tfrac{x_0}{f} N_Z\right) \\
a_{14} &= -\nu \left[\Omega_Y \tfrac{\partial Z}{\partial t} - \Omega_Z \tfrac{\partial Y}{\partial t} - \tfrac{x_0}{f} \left(\Omega_X \tfrac{\partial Y}{\partial t} - \Omega_Y \tfrac{\partial X}{\partial t}\right) + \left(A_X - \tfrac{x_0}{f} A_Z\right)\right] & -b_4 \left(N_X - \tfrac{x_0}{f} N_Z\right) \\
a_{21} &= -\nu \left[\Omega_Z \tfrac{\partial X}{\partial x} - \Omega_X \tfrac{\partial Z}{\partial x} - \tfrac{y_0}{f} \left(\Omega_X \tfrac{\partial Y}{\partial x} - \Omega_Y \tfrac{\partial X}{\partial x}\right)\right] & -b_1 \left(N_Y - \tfrac{y_0}{f} N_Z\right) \\
a_{22} &= -\nu \left[\Omega_Z \tfrac{\partial X}{\partial y} - \Omega_X \tfrac{\partial Z}{\partial y} - \tfrac{y_0}{f} \left(\Omega_X \tfrac{\partial Y}{\partial y} - \Omega_Y \tfrac{\partial X}{\partial y}\right) - \tfrac{N_Z}{Z_0}\right] & -b_2 \left(N_Y - \tfrac{y_0}{f} N_Z\right) \\
a_{23} &= -\nu \left[\Omega_Z \tfrac{\partial X}{\partial s} - \Omega_X \tfrac{\partial Z}{\partial s} - \tfrac{y_0}{f} \left(\Omega_X \tfrac{\partial Y}{\partial s} - \Omega_Y \tfrac{\partial X}{\partial s}\right)\right] & -b_3 \left(N_Y - \tfrac{y_0}{f} N_Z\right) \\
a_{24} &= -\nu \left[\Omega_Z \tfrac{\partial X}{\partial t} - \Omega_X \tfrac{\partial Z}{\partial t} - \tfrac{y_0}{f} \left(\Omega_X \tfrac{\partial Y}{\partial t} - \Omega_Y \tfrac{\partial X}{\partial t}\right) + \left(A_Y - \tfrac{y_0}{f} A_Z\right)\right] & -b_4 \left(N_Y - \tfrac{y_0}{f} N_Z\right)
\end{aligned} \tag{16}$$

The partial derivatives of world coordinates in (16) can be derived from (8), $b_.$ and ν are given in (14).

2.8 The Range Constraint, Z_t, b_4, and why (8) still holds under rotation

Flow parameter b_4 (Eq.(14)) depends on Z_t, the partial t-derivative of Z. We are not explicitly interested in Z_t, thus we want to express it using parameters we are interested in. We know that $U_Z := dZ/dt$ and thus that the time derivative of the first equation in (7) yields the *range constraint* known from [18]

$$Z_t = U_Z - Z_X U_X - Z_Y U_Y \tag{17}$$

valid for *translating* planes, i.e. for constant surface slopes Z_X and Z_Y. Obviously surface slopes change when a surface rotates and the range constraint becomes

$$Z_t = U_Z - Z_X U_X - Z_Y U_Y - X Z_{X,t} - Y Z_{Y,t}\,. \tag{18}$$

$Z_{X,t}$ and $Z_{Y,t}$ are t-derivatives of Z_X and Z_Y.

Equation (7) was derived for constant Z_X and Z_Y. For rotating surfaces we approximate them via first order Taylor expansions $Z_X(t) = Z_X(0) + Z_{X,t}t$ and $Z_Y(t) = Z_Y(0) + Z_{Y,t}t$ and derive for (7)

$$\begin{aligned}
Z(\boldsymbol{x}) \quad &= Z(\mathbf{0}) + Z_X(t)X(\boldsymbol{x}) + Z_Y(t)Y(\boldsymbol{x}) + Z_t t \\
\Leftrightarrow Z(\boldsymbol{x}) &= \tfrac{1}{c}\left(Z(\mathbf{0}) + Z_X(\mathbf{0})s + (Z_t + Z_{X,t}X(\boldsymbol{x}) + Z_{Y,t}Y(\boldsymbol{x}))t\right)
\end{aligned} \tag{19}$$

Substituting Z_t using (18) yields

$$Z(\boldsymbol{x}) = \frac{1}{c}\left(Z(\mathbf{0}) + Z_X s + (U_Z - Z_X U_X - Z_Y U_Y)t\right) = \frac{Z(\mathbf{0}) + Z_X s + \tilde{Z}_t t}{c} \tag{20}$$

where now $\tilde{Z}_t$ is *defined* by the standard range constraint (17). We conclude that (8) still holds for a first order model of rotational motion, if $\tilde{Z}_t$ ignores surface slope changes due to rotation. Consequently Z_t in (14) also becomes $\tilde{Z}_t$.

3 Parameter Estimation

We calculate image derivatives by optimized 5-tab derivative filter sets presented in [19] and then estimate parameters in three steps. **1.** We solve for 4D affine optical flow parameters ν, $b_1, \ldots, b_4$, u_x, u_y, $a_{11}, \ldots, a_{24}$ and brightness change parameters $g_1, g_{1,x}, g_{1,y}, g_2$ using a usual local total least squares estimator (see [4] for details). **2.** We solve for depth Z_0, and surface normals Z_X and Z_Y, and Z_t using (14), where focal length f has to be known e.g. from a calibration step. This allows to calculate c from (14) and partial derivatives of world coordinates from (8). **3.** We solve for translation $\boldsymbol{N}$, and rotation $\boldsymbol{\Omega}$ if desired, using the equations in (16) or – as reference methods – using the range flow method from [3]. Acceleration $\boldsymbol{A}$ can only be estimated up to 1 degree of freedom as we have only 2 equations (the ones for a_{14} and a_{24}) for 3 parameters A_X, A_Y, A_Z. (16) and (14) being an overdetermined system of equations, there are several ways to solve for $\boldsymbol{N}$ and $\boldsymbol{\Omega}$ using a standard least squares estimation scheme. For our experiments we select the following submodels by selecting some or all equations from (16) and (14), or by removing terms when parameters like rotation or acceleration are not estimated. This is equivalent to not modelling these parameters or setting them to zero. We use the following submodels

2D OF trans. estimates $\boldsymbol{N}$ only, using equations for $u_x, u_y, a_{11}, a_{12}, a_{21}, a_{22}$ (i.e. the method from [3]).
2D OF rot. estimates $\boldsymbol{N}$ and $\boldsymbol{\Omega}$ using equations for $u_x, u_y, a_{11}, a_{12}, a_{21}, a_{22}$.
2D OF trans. and ... and *2D OF rot. and ...* using additional equations indicated by ...
4D OF trans. estimates $\boldsymbol{N}$ only, using all 11 equations containing motion information, 10 from (16) and 1 from (14), i.e. the one for b_4.
4D OF rot. estimates $\boldsymbol{N}$ and $\boldsymbol{\Omega}$ only, using the 11 equations.
4D OF estimates $\boldsymbol{N}$, $\boldsymbol{\Omega}$, and $\boldsymbol{A}$ using the 11 equations.

Parameters that are not solved for are set to zero. *4D OF* uses two equations more than *2D OF rot. and* a_{13}, a_{23}, b_4 but estimates $\boldsymbol{A}$ in addition. We therefore get identical results for $\boldsymbol{N}$ and $\boldsymbol{\Omega}$ using the two models. Thus we do not show results for *4D OF* in the experiments below.

4 Experiments

In a first experiment we use synthetic sinusoidal sequences for systematic error analysis. Then the different models are compared on more realistic data with ground truth, i.e. , a moving cube rendered with POV-Ray [17]. Finally we show results for a rotating plant leaf.

4.1 Sinusoidal Pattern

For systematic error analysis we render a surface patch with sinusoidal pattern, where geometry and intensities mimic typical settings used in our actual lab experiments with plants. The 32-bit float intensity values are in the

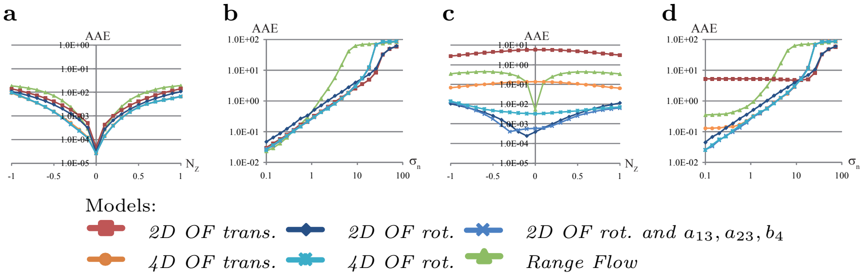

Fig. 1. Average angular error (AAE) versus increasing N_Z (left) and σ_n (right). **a**, **b**: data without rotation, **c**, **d**: with rotation.

range $[50; 150]$. Input sequences are generated with surface patch parameters $Z_0 = 100$ mm, $Z_X = 0.6$, and $Z_Y = -0.5$, and motion parameters $\boldsymbol{N} \approx (0.0073, -0.0037, -0.3)^{\mathrm{T}}$ mm/frame and $\omega = 0.003$ degree/frame around rotational axis $\boldsymbol{v} = (2, 3, 2)^{\mathrm{T}}$, i.e., $\Omega = \omega\boldsymbol{v}$. In each experiment we vary only one of these parameters. The synthetic sensor contains 501×501 pixels with width and height 0.0044 mm. The focal length of the projective camera is set to $f = 12$ mm. We generate data for 9 cameras, positioned horizontal as a 1D, equidistantly spaced camera grid with displacement of 0.5 mm. In order to keep optical flow in camera displacement direction below 1 pixel/displacement, we *preshift* the data by 13 pixel/displacement. The effective image size shrinks to 301×301 pixel due to border effects. Neighborhood Λ is implemented by a Gaussian filter with size $65 \times 65 \times 5 \times 5$ and standard deviations $19 \times 19 \times 1 \times 1$ in x, y, s, t-directions. In order to compare performance of models, we use the average angular error [20]

$$AAE = \frac{1}{N} \sum_{i=1}^{N} \arccos\left(\boldsymbol{r}_t(i)^{\mathrm{T}} \boldsymbol{r}_e(i)\right) \tag{21}$$

for N pixel with a minimum border distance of 60 pixel, true motion $\boldsymbol{r}_t$ and estimated motion $\boldsymbol{r}_e$, with $\boldsymbol{r} = (\boldsymbol{N}^{\mathrm{T}}, 1)^{\mathrm{T}}$ for translation and $\boldsymbol{r} = (\boldsymbol{\Omega}^{\mathrm{T}}, 1)^{\mathrm{T}}$ for rotation. Figure 1 shows average angular errors of translational motion estimates for sequences without (**a**,**b**) and with (**c**,**d**) rotation. We show errors for increasing translational motion N_Z in Figs. 1**a** and **c** and for increasing standard deviation of noise σ_n in Figs. 1**b** and **d**. Figures 1**a** and **b** demonstrate that all models perform almost equally well for sequences without rotation. Models using more affine parameters (*4D OF trans./rot.*, and *2D OF rot. and* a_{13}, a_{23}, b_4) perform best. Range flow performs only slightly better for low noise sequences. In case of rotation (Figs. 1**c** and **d**) *range flow* and the translational models yield high errors compared to rotational models. However, comparing rotational models, *4D OF rot.* performs worst. This indicates that modelling $\boldsymbol{A}$ in the equations for a_{14} and a_{24} ((16)) or not using a_{14} and a_{24} (i.e. *2D OF rot. and* a_{13}, a_{23}, b_4) is beneficial. In case of noise rotational models using 4D affine terms (*2D OF rot. and* a_{13}, a_{23}, b_4 and *4D OF rot.*) show best performance up to $\sigma_n = 10$. *4D OF trans.* shows considerable better performance than *Range Flow*, despite for $N_Z = 0$ and

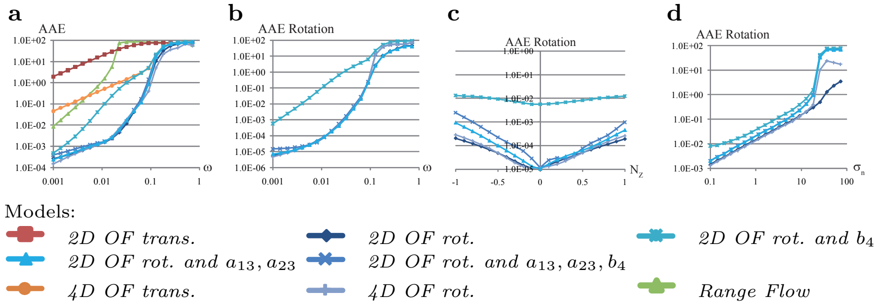

Fig. 2. a: AAE of $\boldsymbol{N}$ versus ω and b–d: AAE of $\boldsymbol{\Omega}$ versus **b**: ω, **c**: N_Z and **d**: σ_n.

no noise, and performs as good as the best rotational models for $1 < \sigma_n < 10$. In Fig. 2 we compare average angular errors of translational (Fig. 2**a**) and rotational (Fig. 2**b**–**d**) motion parameters for different rotation models. Translational models and Range Flow are shown for reference in Fig. 2**a**. Figures 2**c** and **d** show angular errors of rotational parameters for a sequence with rotation and increasing N_Z and σ_n, respectively, i.e. , the $\boldsymbol{\Omega}$ counterparts of Figs. 1**c** and **d**.

The figures demonstrate that incorporating the affine parameters a_{14} and a_{24} in *4D OF rot.* without modelling of acceleration significantly increases errors. Figures 2**a** and **b** show average angular errors of translational and rotational parameters for sequences with increasing ω. Rotational models without a_{14} and a_{24} perform similar and up to three orders of magnitude better than *range flow* and the translational models.

We conclude that modelling rotation yields almost always significantly lower or at least similar errors as the translational models and *range flow*. Using a_{14} and a_{24} without modelling acceleration $\boldsymbol{A}$ should be avoided.

4.2 Synthetic Cube

The synthetic cube sequence allows us to compare models on more realistic data with ground truth available. The cube center is at $Z = 600$mm, moves with $\boldsymbol{N} = (-0.2, 0, -1)^T$ mm/frame, and rotates around its Y-axis with $\omega = 0.4$ degrees/frame. It is covered with a noise pattern in order to make local estimation reliable. Neighborhood Λ is the same as for the sinusoidal sequences. The 1D camera grid contains 9 cameras with a displacement of 5 mm. Figures 3**a**–**d** show first and last frame of the central camera, two regions where errors are evaluated, and ground truth motion, respectively. Structure estimation accuracies for different choices of Λ and optical flow types are given in Tab.1. We see that using surface normals, i.e. affine terms $b_.$, and data from more than one point in time in the estimation improves accuracy by more than 1 order of magnitude. Accuracy is then comparable to typical laser scanning range sensors, e.g. Sick IVP Ruler E600 with 0.2mm resolution. Motion estimates of two translational models, one rotational model and *Range Flow* are shown in Figs. 3**e**–**h**. The errors are amplified by a factor of 5 for better comparison of the models. The estimates of *2D*

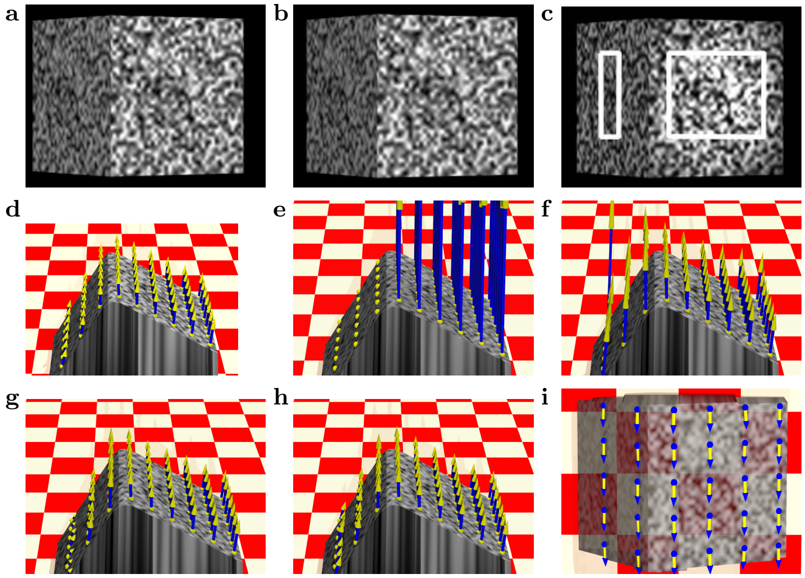

Fig. 3. Cube moving towards camera with rotation. Top row: First (**a**) and last (**b**) input frame, and central frame with evaluation areas (**c**). **d**: ground truth motion. Motion estimates $\boldsymbol{U}$ with amplified errors. **e**: *2D OF trans.*, **f**: *Range Flow*, **g**: *4D OF trans.*, and **h**: *2D OF rot. and* a_{13}, a_{23}, b_4. **i**: Rotational motion $\boldsymbol{\Omega}$ estimated via *2D OF rot. and* a_{13}, a_{23}, b_4.

Table 1. Average surface distances in mm (mean ± std. deviation). 'Left' and 'right' refer to the areas of interest depicted in Fig.3**c**.

Neighborhood Λ	flow type	Error 'left'	Error 'right'
$65 \times 65 \times 1 \times 1$	not affine	2.77 ± 0.91	3.10 ± 1.34
$65 \times 65 \times 1 \times 1$	affine	0.15 ± 0.07	0.29 ± 0.16
$65 \times 65 \times 5 \times 5$	affine	0.08 ± 0.03	0.21 ± 0.11

OF trans. clearly show large errors, where estimates on the right side of the cube point in Z-direction, estimates on the left side of the cube are not visible because they point inwards the cube. Estimates of *Range Flow* are more accurate, but distorted near borders of the cube. Models *4D OF trans.* and *2D OF rot. and* a_{13}, a_{23}, b_4 yield more accurate results. Estimates of the translational model are still distorted, mainly on the left side of the cube. Estimation results of the rotational model best match the ground truth. Fig. 3**i** shows a rendered top view of the cube with estimation results of rotational motion using the model *2D OF rot. and* a_{13}, a_{23}, b_4. The estimates clearly recover the true motion.

Table 2 shows angular errors for the regions depicted in Fig. 3**c** which quantitatively confirm the visual impression of the rendered results. *2D OF trans.* performs better when b_4, a_{13} and a_{23}, or all three terms are additionally used for estimation. Otherwise estimates are heavily distorted. The same is true for

Table 2. Average angular error (AAE) and standard deviations in degrees of translational and rotational motion parameters of regions on left and right side of the cube (see Fig. 3**c**). Errors or standard deviations above 1° (AAE) are indicated in red, below 0.1° (AAE) in green.

motion model	affine parameters	AAE left region		AAE right region	
		translation	rotation	translation	rotation
Translation	2D OF	112 ± 1.01	n/a	19.8 ± 0.67	n/a
	2D OF + b_4	1.22 ± 0.43	n/a	0.32 ± 0.20	n/a
	2D OF + a_{13},a_{23}	1.72 ± 1.28	n/a	1.06 ± 0.57	n/a
	2D OF + a_{13},a_{23},b_4	0.91 ± 0.64	n/a	0.58 ± 0.33	n/a
	4D OF	0.91 ± 0.64	n/a	0.58 ± 0.33	n/a
Translation and rotation	2D OF	6.85 ± 6.24	0.018 ± 0.009	1.73 ± 1.37	0.004 ± 0.011
	2D OF + b_4	0.52 ± 0.19	0.017 ± 0.009	0.27 ± 0.17	0.004 ± 0.011
	2D OF + a_{13},a_{23}	0.93 ± 0.37	0.017 ± 0.009	0.19 ± 0.11	0.004 ± 0.011
	2D OF + a_{13},a_{23},b_4	0.67 ± 0.29	0.017 ± 0.009	0.22 ± 0.13	0.005 ± 0.011
Range Flow		8.86 ± 0.69	n/a	1.88 ± 0.31	n/a

translation estimates with models also estimating rotation. Rotation estimates are equally well for all rotation models. Errors of *Range Flow* are lower than for *2D OF trans.*, but significantly higher than for models incorporating more affine terms.

4.3 Plant Leaf

Figure 4**a** shows one frame of a tobacco plant leaf input sequence. The leaf is textured with watercolour to reduce errors coming from the aperture problem (cmp. [1]). The scene is illuminated by directed infrared light emitting diodes from top causing illumination changes on the leaves. The maximal width of the leaf is approximately 20 mm. Images are taken by a movingstage-based 1D camera grid with 9 positions at 1 mm distance (see Sec. 1). Sampling rate of the camera per position is one image every 2 minutes. Sensor size is 1600 × 1200 pixel. Neighborhood Λ is implemented using a Gaussian filter with size 121 × 121 × 5 × 5 and standard deviation 41 × 41 × 1 × 1 in x, y, s, t-direction.

The big leaf on the right rotates upward around its node where it is attached to the stem (i.e. approx. around the Y-axis). This results in a visible motion towards the camera and to the left. Moreover the leaf unrolls along its midvein and folds its sides up. Figure 4**b** shows estimated structure and surface normals. Visibly the true structure is well recovered. Translation estimates for the presented models are shown in Fig. 4**c**–**e**. *Range Flow* [21] significantly overestimates the motion (Fig. 4**c**). With the purely translational motion model *2D OF trans.* [3] estimation results are heavily corrupted (Fig. 4**d**). This model apparently interprets shrinkage of the projected leave length in x-direction due to rotation as being caused by motion away from the camera. The rotational model *2D OF rot*

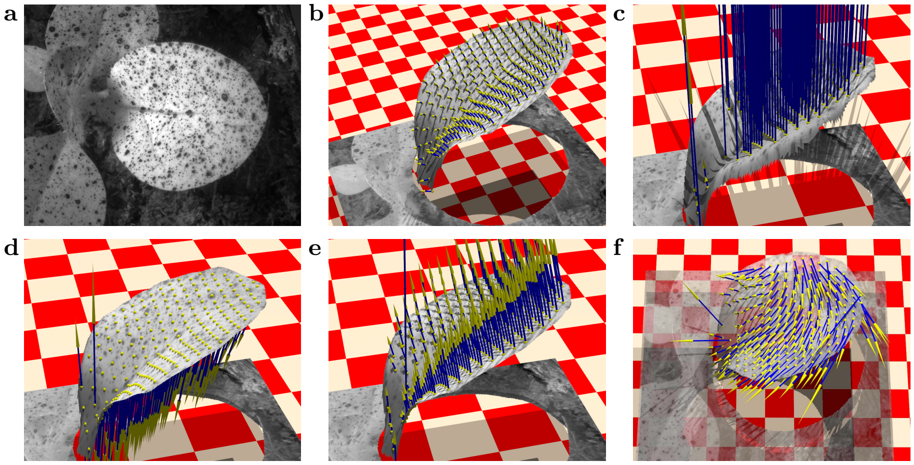

Fig. 4. Plant Leaf Sequence. **a**: Central frame of central camera. **b**: Estimated structure and surface normals. Motion estimates for **c**: *Range Flow* for Varying Illumination, **d**: *2D OF trans.* and **e**: proposed new model *2D OF rot and* a_{13},a_{23} *and* b_4. **f**: estimated rotational velocity.

and a_{13},a_{23} *and* b_4 yields a severely improved motion vector field, even though motion still seems to be overestimated. Figure 4**f** shows estimated rotational motion vectors. Rotation around the node is well visible. Unrolling and folding of the leaf can be recovered by analysing changes in the rotation vector field. Making this possible was the main goal of the presented work (cmp. Sec.1).

5 Summary and Conclusions

In this paper we presented a 4D affine optical flow model and how the parameters of this model can be explained by real world parameters. Based on a rigid surface patch we modelled translation, acceleration and rotation. The rotational model improves estimation results in almost all cases and additionally allows to estimate rotational parameters which is of high interest for understanding plant physiology. Synthetic experiments showed that modelling acceleration is not sufficient to estimate rotation reliably and should therefore not be used if rotation occurs in the sequence. The 4D affine model and its explanation of real world parameters improved accuracy of motion estimates on synthetic and real data compared to *Range Flow* and previous *2D OF* affine models. In order to increase accuracy further and cope with different scenarios than plant leaf estimation, the main focus in future research will be on developing a more sophisticated estimator. Furthermore the estimator should be able to handle large motions coming from camera displacement. This is a prerequisite for using full camera arrays, like e.g. [22], instead of moving stages, and adapting the affine optical flow model to other application areas.

References

1. Biskup, B., Küsters, R., Scharr, H., Walter, A., Rascher, U.: Quantification of plant surface structures from small baseline stereo images to measure the three-dimensional surface from the leaf to the canopy scale. In: Nova Acta Leopoldina. Number 357 in 96 (2009) 31–47
2. Walter, A., Christ, M.M., Barron-Gafford, G.A., Grieve, K.A., Paige, T., Murthy, R., Rascher, U.: The effect of elevated co2 on diel leaf growth cycle, leaf carbohydrate content and canopy growth performance of populus deltoides. Global Change Biology **8** (2005) 1207 – 1219
3. Schuchert, T., Scharr, H.: Simultaneous estimation of surface motion, depth and slopes under changing illumination. In: DAGM. (2007) 184–193
4. Schuchert, T., Scharr, H.: An affine optical flow model for dynamic surface reconstruction. In: Statistical and Geometrical Approaches to Visual Motion Analysis. Number 5604 in LNCS (2009) 70–90
5. Longuet-Higgins, H., Prazdny, K.: The interpretation of a moving retinal image. Proceedings of The Royal Society of London B, 208 (1980) 385–397
6. Kanatani, K.: Structure from motion without correspondence: general principle. In: Proc. Image Understanding Workshop. (1985) 10711–6
7. Adiv, G.: Determining 3-d motion and structure from optical flow generated by several moving objects. PAMI **7** (1985) 384–401
8. Subbarao, M., Waxman, A.: Closed form solutions to image flow equations for planar surfaces in motion. CVGIP: Graphical Models and Image Processing **36** (1986) 208–228
9. Szeliski, R.: A multi-view approach to motion and stereo. In: CVPR. (1999)
10. Vedula, S., Baker, S., Rander, P., Collins, R., Kanade, T.: Three-dimensional scene flow. PAMI **27** (2005) 475–480
11. Carceroni, R., Kutulakos, K.: Multi-view 3d shape and motion recovery on the spatio-temporal curve manifold. In: ICCV (1). (1999) 520–527
12. Wedel, A., Rabe, C., Vaudrey, T., Brox, T., Franke, U., Cremers, D.: Efficient dense scene flow from sparse or dense stereo data. In: ECCV, Marseille, France (2008)
13. Pons, J.P., Keriven, R., Faugeras, O.: Multi-view stereo reconstruction and scene flow estimation with a global image-based matching score. IJCV **72(2)** (2007) 179–193
14. Kolev, K., Klodt, M., Brox, T., Cremers, D.: Continuous global optimization in multiview 3d reconstruction. IJCV **84** (2009) 80–96
15. Fleet, D., Weiss, Y.: Optical flow estimation. In: Mathematical models for Computer Vision: The Handbook. Springer (2005)
16. L.H.Matthies, R.Szeliski, T.Kanade: Kalman filter-based algorithms for estimating depth from image sequences. IJCV **3** (1989) 209–236
17. Cason, C.: Persistence of vision ray tracer (POV-Ray), version 3.6, Windows (2005)
18. Spies, H., Jähne, B., Barron, J.: Range flow estimation. CVIU **85** (2002) 209–231
19. Scharr, H.: Optimal filters for extended optical flow. In: Complex Motion, LNCS 3417. (2004) 14–29
20. Barron, J., Fleet, D., Beauchemin, S.: Performance of optical flow techniques. In: IJCV. (1994) 43–77 12(1).
21. Schuchert, T., Aach, T., Scharr, H.: Range flow for varying illumination. In: ECCV. Volume 5302/2008., Springer Berlin / Heidelberg (2008) 509–522
22. ViewPLUS: ProFUSION 25 (2008) http://www.viewplus.co.jp/products/profusion25/ProFUSION25-e.pdf.