

AN APPLICATION OF SURFACE RECONSTRUCTION FROM ROTATIONAL MOTION

Reinhard Klette¹, Dirk Mehren² and Volker Rodehorst¹

Abstract. For surface reconstruction using motion, objects are placed on a rotating disc in front of a single camera. For camera calibration the method by [Tsai 1986] was implemented, extended (calculation of distorted from undisturbed coordinates) and optimized (e.g. with respect to the number of calibration planes and points in each plane). It is described how the calibration results can be used for this special case of surface reconstruction of objects on a rotating disc. Motion vectors calculated from point correspondences are used as input for this calculation of 3-D point positions. In two theorems, new reconstruction formulas are given. Experimentally, accurate depth values could be obtained for sparse object surface points. It is suggested to combine these exact values with "surface drafts" calculated by approaches based on reflectance properties.

Key words: surface reconstruction, shape from motion, camera calibration, rotating disc

1 INTRODUCTION

Shape from motion is often cited as one of the basic computer vision approaches [Horn 1986], and even some books in computer vision are focusing on that issue, e.g. [Kanatani 1990, Maybank 1993]. Rigid objects are projected into the image plane assuming a certain camera model. For a time sequence of such projections, motion vectors have to be computed. Based on these vectors, certain shape values of the objects may be determined. Here, "shape" is defined to be a set of gradient values of the object surface, cp. [Horn 1986]. Then, from shape values some depth information may be derived leading to a certain 3-D representation of the projected objects. Altogether, this process may be applied, e.g., to support geometric modeling of 3-D objects, or to generate depth maps.

Perhaps the first problem of "surface from motion" is computation of dense motion vector fields, cp. [Barron/Fleet/Beauchemin 1994, Handschack/Klette 1994]. Because these motion vector fields will be "quite erroneous in

¹ Fachgebiet Computer Vision, Institut für Technische Informatik im FB 13, Technische Universität Berlin, Franklinstr. 28/29, FR 3-11, D-10587 Berlin

² Daimler-Benz AG, Image Understanding Systems (F3B), Research and Technology, P.O. Box 2360, D-89013 Ulm

general", the approach may be realistic only in specific image regions with high-accuracy motion vector fields.

A second problem of shape from motion concerns the calculation of shape, i.e. surface normals, based on motion vector fields. Certain integrative methods, integrating also further features as, e.g. the light source direction, will work for correct motion vector fields, cp. [Aloimonos/Shulman 1990, Klette/Rodehorst 1993], and lead to unique shape values. Because differences of neighboring motion vectors are used, this integrative approach is even more sensitive to incorrect motion vector results as a direct method will be. A good collection of direct methods for shape from motion may be found in [Kanatani 1990], but these methods do not lead to unique shape values (if no further techniques, e.g. regularization, are applied), and these methods are numerically complicated.

A third problem is connected with going from shape (gradients) to depth, i.e. exact surface point positions in 3-D space. There are several suggestions in literature, e.g. [Aloimonos/Shulman 1990, Frankot/Chellappa 1988]. Such methods should be refined to react on local gradient distributions. But, e.g. for a gradient image of stairs it will always remain to be impossible to transform this into a surface reconstruction.

By using dynamic stereo based on the rotating disc and including the modification of the epipolar constraint of static stereo, corresponding points in consecutive images may be used to calculate depth (i.e. problems two and three will not appear in this approach).

In this paper, controlled object motion (objects on a rotating disc in front of a single static camera) is considered for calculating depth from motion vectors. The proposed dynamic stereo method is based on accurate camera calibration which has to be realized as preprocessing. The method by [Tsai 1986, Tsai 1987] was selected and extended in a single subprocess (calculation of distorted from undisturbed coordinates). Then, during object surface reconstruction, calibration results may be used for efficient and robust depth calculations. Therefore, two new theorems will be given specifying ways of reconstruction for the specific situation of a rotating disc.

For the first problem of correct motion vector fields, no new solutions could be found. Thus, if (!) motion vectors are correct, then depth may be calculated with very high accuracy and good time efficiency. This holds for dense vector fields, leading to dense depth maps, as well as for sparse motion vectors, leading to singular reconstructed points on the object surface, which are called *fixation points*. Because reflectance based methods may be used for calculating "rough drafts" of object surfaces, cp. [Schlüns/Wittig 1993], it is suggested to use fixation points for further enhancement of reconstructed object surfaces.

The paper is structured as follows: In Section 2 the calibration parameters are sketched for later use in the reconstruction process. The proposed extension of Tsai's method is described. In Section 3 the basic mathematics is given for

reconstructing surface points of objects placed on the rotating disc. Single motion vectors are used as input and the calibration parameters are assumed to be known. In Section 4, a few experimental results are sketched and the real-time aspect is discussed. Conclusions are given in Section 5.

2 CAMERA CALIBRATION

For camera calibration, internal camera parameters as well as geometric relations between camera coordinates and world coordinates have to be pre-calculated, and these data have essential influence on the accuracy obtainable in surface reconstruction. A good review on calibration techniques was given by [Tsai 1986]. The selected method may be classified to be a non-linear optimization technique. Here, only a brief definition of used calibration parameters will be given. The calibration method itself may be found in [Tsai 1986, Tsai 1987].

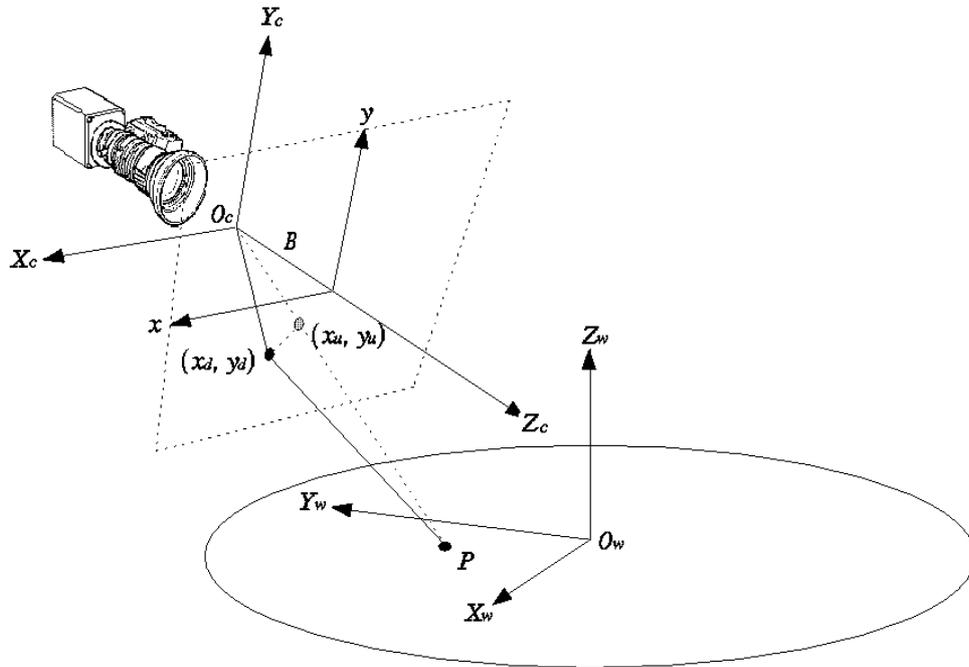


Figure 1: Camera geometry with perspective projection (world coordinates $X_w Y_w Z_w$ with respect to rotating disc, camera coordinates $X_c Y_c Z_c$, and ideal projected coordinates $x_u y_u$) and radial lens distortion (distorted image coordinates $x_d y_d$).

For the camera model, perspective projection and radial lens distortion was assumed. In Fig. 1, the different (left handed) coordinate systems are illustrated.

In this paper,

- (X_w, Y_w, Z_w) denote the 3-D coordinates of object points P in the *world coordinate system* **WCS**,
- (X_c, Y_c, Z_c) denote the 3-D coordinates of P in the *camera coordinate system* **CCS**,
- B is the distance of image plane to projection center (*focal length*),
- (x_u, y_u) are *undistorted* image coordinates of (X_c, Y_c, Z_c) assuming an ideal pinhole camera,
- (x_d, y_d) are *distorted* image coordinates, differing from (x_u, y_u) by radial lens distortion, and
- (x_f, y_f) are *device-dependent* coordinates of (x_d, y_d) in the digitized image (not illustrated in Fig. 1).

The Z-axis Z_c of the camera coordinate system coincides with the optical axis. The Z-axis Z_w of the world coordinate system coincides with the rotation axis of the rotating disc. These conditions are not of essential importance for calibration, but will simplify the used approach to shape analysis.

All coordinates and parameters will be measured at the same scale, e.g. μm . The only exception are coordinates (x_f, y_f) for the digitized image which are given in (sub-) pixels. Besides the device dependent scaling they differ from the remaining image coordinates by its representation in form of row and column positions.

Basically, an object point P may be in arbitrary position in 3-D space. Four steps are considered for mapping such a point (X_w, Y_w, Z_w) onto device dependent coordinates (x_f, y_f) .

2.1 From WCS to CCS

At first, an affine transform has to be considered from world coordinates (X_w, Y_w, Z_w) into camera coordinates (X_c, Y_c, Z_c) . Let

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} = R \cdot \begin{bmatrix} X_w \\ Y_w \\ Z_w \end{bmatrix} + T \quad (2.1)$$

with

$$R = \begin{bmatrix} r_1 & r_2 & r_3 \\ r_4 & r_5 & r_6 \\ r_7 & r_8 & r_9 \end{bmatrix} \quad \text{and} \quad T = [T_x, T_y, T_z]^T. \quad (2.2)$$

The transform of a rigid body from a Cartesian coordinate system into an other remains to be unique if the 3-D rotation around the origin is performed before the 3-D translation. *Rotation matrix* R and *translation vector* T have to be calibrated.

2.2 From CCS to Undistorted Image Coordinates

The 3-D camera coordinates (X_c, Y_c, Z_c) are transformed in ideal, undistorted image coordinates (x_u, y_u) by perspective projection. According to the pinhole camera model it holds

$$x_u = \frac{B \cdot X_c}{Z_c} \quad \text{and} \quad y_u = \frac{B \cdot Y_c}{Z_c} . \quad (2.3)$$

The *focal length* B has to be calibrated.

2.3 From Undistorted Coordinates to Distorted Ones

For the calculation of undistorted image coordinates (x_u, y_u) from real, distorted image coordinates (x_d, y_d) the method proposed by [Tsai 1986] was selected. The equations

$$x_d + D_x = x_u, \quad y_d + D_y = y_u \quad (2.4)$$

are based on the following abbreviations:

$$D_x = x_d \cdot (\kappa_1 r^2 + \kappa_2 r^4), \quad D_y = y_d \cdot (\kappa_1 r^2 + \kappa_2 r^4)$$

and $r = \sqrt{x_d^2 + y_d^2} . \quad (2.5)$

Here, radial distortion is modeled. The *distortion coefficients* κ_1 and κ_2 have to be calibrated. A positive value of κ_1 or κ_2 means that some stretching has to be performed for going from distorted to undistorted coordinates.

Different lens aberrations or distortions may be characterized by an infinite sequence of coefficients. Here, only the first two coefficients are considered. Otherwise, numeric instability would influence the result. Equations (2.4) may be used for the restoration of images if values of (x_d, y_d) are known. But, for going from undistorted to distorted coordinates, non-linear equations will result. This direction is not described in [Tsai 1986]. Applying numeric methods for solving non-linear equational systems leads to non-acceptable inefficiency, e.g. if a complete image has to be transformed. For that reason, for distorting ideal image points (x_u, y_u) the following approximation was used:

$$x_{di} = \frac{x_u}{1 + \kappa_1 r_{(i-1)}^2 + \kappa_2 r_{(i-1)}^4}, \quad y_{di} = \frac{y_u}{1 + \kappa_1 r_{(i-1)}^2 + \kappa_2 r_{(i-1)}^4}$$

$$\text{with } r_i = \sqrt{x_{di}^2 + y_{di}^2} \quad \text{for } i \in \{1, \dots, n\} . \quad (2.6)$$

With initial value $r_0 = \sqrt{x_u^2 + y_u^2}$, (2.6) leads to a first approximation of the desired solution (x_d, y_d) . By iteration, improved radii r_i may be calculated.⁴ This iterative procedure represents an extension of the original *Tsai* method.

2.4 From Distorted Coordinates to Device Dependent Ones

Finally, real or distorted image coordinates (x_d, y_d) are transformed into device dependent image coordinates (x_f, y_f) . Besides a specific scaling, centralized coordinates have to be transformed into row and column position values. Also a variable is introduced for possible digitization errors. Let

$$x_f = \frac{s_x x_d}{d'_x} + c_x , \quad y_f = \frac{y_d}{d_y} + c_y , \quad (2.7)$$

where

(c_x, c_y) denotes the origin of the image coordinates,
 $d'_x = d_x \frac{N_{cx}}{N_{fx}}$ is the distance of neighboring pixels in an image row,
 d_x is the distance between neighboring sensor elements in X-direction,
 d_y is the distance between neighboring sensor elements in Y-direction,
 N_{cx} is the number of sensor elements in X-direction, and
 N_{fx} is the number of pixels in each image row.

During scanning and digitization by CCD cameras and frame grabbers, the hardware timing is not always perfect. [Tsai 1986] was reporting about up to 25 pixel horizontal aberration. The scaling parameter s_x and the image origin (c_x, c_y) have to be calibrated. For sensor parameters, the data sheet of the producer has to be used.

3 SURFACE RECONSTRUCTION

Assume that during taking images of the object placed on the rotating disc, projections C_1 and C_2 in the image plane of the **CCS** are given for the same visible surface point W in the **WCS**, at time slots t_1 and t_2 . The task consists in calculating the coordinates of W , where the Z -coordinate of W in the **CCS** may be identified with the notion *depth*.

⁴ In our experiments it was typical that some stabilization of the calculated radii did appear after about 8 iterations.

At first, assume that the rotation may be under control, i.e. the rotation angle between time t_1 and t_2 is known. Based on the calibration results, the defined task can be solved. But, then it will be shown that knowledge of rotation angle is not necessary at all: The given task can be solved just by knowing that C_1 and C_2 are corresponding to the same surface point W .

3.1 World Point Position for Given Rotation

For transforming a point W of a world coordinate system **WCS** into a point C of a camera coordinate system **CCS**, the linear equation is given by

$$RW + T = C \quad (3.1)$$

where R denotes the 3 x 3 rotation matrix, and T denotes the translation vector⁵. For a rotation R_Δ of the disc and a point W in the **WCS**, assume

$$RW + T = C_1 \quad (3.2)$$

at time t_1 before rotation, and

$$RR_\Delta W + T = C_2 \quad (3.3)$$

at time t_2 after rotation. It follows that

$$\begin{aligned} R^T(C_1 - T) &= W \\ &= (RR_\Delta)^T(C_2 - T). \end{aligned} \quad (3.4)$$

Thus, it holds that

$$(RR_\Delta)^T T - R^T T = (RR_\Delta)^T C_2 - R^T C_1. \quad (3.5)$$

Assuming an ideal pinhole camera with focal length B and camera-centered perspective projection, for the *ideal* image point (x_{P_i}, y_{P_i}) at time t_i , $i \geq 1$, it holds that

$$x_{P_i} = \frac{X_{C_i} B}{Z_{C_i}}, \quad \text{and} \quad y_{P_i} = \frac{Y_{C_i} B}{Z_{C_i}}. \quad (3.6)$$

These equations hold for the ideal, non-distorted points (x_{P_i}, y_{P_i}) in the image plane. For distorted image coordinates (x_{D_i}, y_{D_i}) , see Section 2, the following equations have to be applied:

$$\begin{aligned} x_{P_i} &= x_{D_i}(1 + \kappa_1 r^2 + \kappa_2 r^4), & y_{P_i} &= y_{D_i}(1 + \kappa_1 r^2 + \kappa_2 r^4), \\ \text{and} \quad r &= \sqrt{X_{D_i}^2 + Y_{D_i}^2}. \end{aligned} \quad (3.7)$$

For distortion coefficients κ_1 and κ_2 , based on measurements of the distorted image coordinates, at first the ideal image points P_i can be computed, and secondly these ideal points may be used for determining points C_i in the **CCS**:

⁵ R as well as T was determined during camera calibration.

$$C_i = \begin{pmatrix} X_{C_i} \\ Y_{C_i} \\ Z_{C_i} \end{pmatrix} = \begin{pmatrix} \frac{x_{P_i} Z_{C_i}}{B} \\ \frac{y_{P_i} Z_{C_i}}{B} \\ Z_{C_i} \end{pmatrix} = Z_{C_i} \begin{pmatrix} \frac{x_{P_i}}{B} \\ \frac{y_{P_i}}{B} \\ 1 \end{pmatrix} = Z_{C_i} E_i . \quad (3.8)$$

E_i is introduced as abbreviation of the given vector. Furthermore, let

$$\vec{b} = (RR_\Delta)^T C_2 - R^T C_1 . \quad (3.9)$$

For determining \vec{b} , an equational system with unknowns Z_{C_1} and Z_{C_2} has to be solved:

$$\begin{aligned} \vec{b} &= (RR_\Delta)^T Z_{C_2} E_2 - R^T Z_{C_1} E_1 \\ &= Z_{C_2} (RR_\Delta)^T E_2 - Z_{C_1} R^T E_1 \\ &= Z_{C_2} \vec{c} - Z_{C_1} \vec{d} \\ &= \begin{pmatrix} \vec{c} \\ -\vec{d} \end{pmatrix} \cdot \begin{pmatrix} Z_{C_2} \\ Z_{C_1} \end{pmatrix} . \end{aligned} \quad (3.10)$$

This linear equational system is denoted by

$$\vec{b} = \mathbf{A} \cdot \vec{z} . \quad (3.11)$$

It is over-determined because three equations are available for two unknowns. Numerically stable it can be solved using pseudo inverse and pivoting,

$$(\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \vec{b} = \vec{z} . \quad (3.12)$$

For calibration data R , T and a given disc rotation R_Δ , at first the values Z_{C_1} and Z_{C_2} of vector \vec{z} may be calculated from image points E_1 and E_2 . Here, the calibration parameter B , κ_1 , κ_2 are used for this computation. The resulting depth values in the camera coordinate system have to be transformed into the world coordinate system. Both points $C_1 = Z_{C_1} E_1$ and $C_2 = Z_{C_2} E_2$ are representations of the same world point W , but at different times. Thus, they may be projected, and results may be compared for controlling correctness.

Theorem 3.1: For E_i as defined in (3.8) it holds

$$W = R^T (Z_{C_1} E_1 - T) = (RR_\Delta)^T (Z_{C_2} E_2 - T) . \quad (3.13)$$

Thus, surface point W is reconstructed in 3-D space. There are three equations and two unknowns. In fact, the disc rotation angle φ_Δ may be taken as third unknown. But, the equation system "loses its linearity" if considered also for unknown φ_Δ .

3.2 World Point Position for Unknown Rotation

Originated from the fact that depth computation is over-determined in Section 3.1, now the idea is to calculate further information from such an equational system. As third unknown, the disc rotation angle φ_Δ is considered, defining a rotation R_Δ . The system (3.11) "loses its linearity" if considered for three unknowns Z_{C_1} , Z_{C_2} and φ_Δ . For dealing with this problem, two new approaches will be considered.

Again, for a point W of the world coordinate system **WCS**, the equations (3.2) and (3.3) are considered at consecutive discrete time slots t_1 and t_2 . Again, from these equations it follows that

$$R^T C_1 - R^T T = R_\Delta^T (R^T C_2 - R^T T) \quad (3.14)$$

Also, the equations (3.8) are used as above. For unknown rotation, lens distortion may be integrated into the solution as already described in Section 3.1. It follows that

$$Z_{C_1} R^T E_1 - R^T T = R_\Delta^T (Z_{C_2} R^T E_2 - R^T T), \quad (3.15)$$

what can be abbreviated by

$$z_1 \vec{a} - \vec{c} = R_\Delta^T (z_2 \vec{b} - \vec{c}). \quad (3.16)$$

By rotating the disc, the corresponding rotation matrix R_Δ specifies (only) a rotation around the Z-axis. The unknown angle φ_Δ appears in this matrix as argument of different trigonometric functions.

3.2.1 Straightforward Solution

The inverse rotation matrix around the Z-axis may be represented as

$$R_\Delta^T = \begin{pmatrix} \cos(\varphi_\Delta) & -\sin(\varphi_\Delta) & 0 \\ \sin(\varphi_\Delta) & \cos(\varphi_\Delta) & 0 \\ 0 & 0 & 1 \end{pmatrix}. \quad (3.17)$$

With equation (3.16) it follows that

$$z_1 a_x - c_x = (z_2 b_x - c_x) \cos(\varphi_\Delta) - (z_2 b_y - c_y) \sin(\varphi_\Delta) \quad (3.18.1)$$

$$z_1 a_y - c_y = (z_2 b_x - c_x) \sin(\varphi_\Delta) + (z_2 b_y - c_y) \cos(\varphi_\Delta) \quad (3.18.2)$$

$$z_1 a_z - c_z = z_2 b_z - c_z \quad (3.18.3)$$

Solving (3.18.3) for z_2 , and using this in (3.18.1) and (3.18.2), it follows that

$$\begin{aligned} z_1 a_x - c_x &= \left(\frac{z_1 a_z b_x}{b_z} - c_x \right) \cos(\varphi_\Delta) - \left(\frac{z_1 a_z b_y}{b_z} - c_y \right) \sin(\varphi_\Delta) \\ z_1 a_y - c_y &= \left(\frac{z_1 a_z b_x}{b_z} - c_x \right) \sin(\varphi_\Delta) + \left(\frac{z_1 a_z b_y}{b_z} - c_y \right) \cos(\varphi_\Delta). \end{aligned} \quad (3.19)$$

This equational system with two unknowns φ_Δ and z_1 has a unique solution⁶, and by applying (3.18.3), altogether it follows

Theorem 3.2: It holds

$$\begin{aligned} \varphi_\Delta &= 2 \arctan \left(\frac{c_2(a_y b_z - b_y a_z) + c_1(a_x b_z - b_x a_z)}{c_2(a_x b_z + b_x a_z) - c_1(a_y b_z + b_y a_z)} \right), \\ z_1 &= \frac{b_z(c_x - c_x \cos(\varphi_\Delta) + c_y \sin(\varphi_\Delta))}{a_x b_z - a_z b_x \cos(\varphi_\Delta) + a_z b_y \sin(\varphi_\Delta)}, \\ z_2 &= z_1 \frac{a_z - c_z}{b_z - c_z}. \end{aligned} \quad (3.20)$$

With these solutions, the world coordinates of point W may be calculated as in Section 3.1. in (3.13).

3.2.2 Using Cylinder Coordinates

A different solution algorithm may be derived by following cylinder coordinate representation. The specific rotation of the disc around the Z-axis can be easily described using these coordinates. Here, a point is represented by coordinates (φ, ρ, h) with h as height, φ as angle between X-axis and projected straight line of length ρ . It holds

$$\begin{aligned} X &= \rho \cos \varphi, & Y &= \rho \sin \varphi, & Z &= h \\ \rho &= \sqrt{X^2 + Y^2}, & \varphi &= \arctan \left(\frac{Y}{X} \right). \end{aligned} \quad (3.21)$$

Here, a rotation R_Δ around the Z-axis is defined by a vector addition. Let P be a point. It follows

⁶ Solutions were calculated using *MapleV*[®], a mathematical symbol manipulation program.

$$R_{\Delta} \cdot P_{cartesian} = P_{cylindrical} + \begin{pmatrix} \varphi_{\Delta} \\ 0 \\ 0 \end{pmatrix}. \quad (3.22)$$

The "Cartesian equation" (3.16) results into its "cylindrical pendant"

$$\begin{pmatrix} \arctan\left(\frac{Z_1 a_y - c_y}{Z_1 a_x - c_x}\right) \\ \sqrt{(Z_1 a_x - c_x)^2 + (Z_1 a_y - c_y)^2} \\ (Z_1 a_z - c_z) \end{pmatrix} = \begin{pmatrix} \arctan\left(\frac{Z_2 b_y - c_y}{Z_2 b_x - c_x}\right) \\ \sqrt{(Z_2 b_x - c_x)^2 + (Z_2 b_y - c_y)^2} \\ (Z_2 b_z - c_z) \end{pmatrix} + \begin{pmatrix} -\varphi_{\Delta} \\ 0 \\ 0 \end{pmatrix}. \quad (3.23)$$

Theorem 3.2*: For reconstruction, the following equational system can be used,

$$Z_1 a_z = Z_2 b_z, \quad (3.24.1)$$

$$(Z_1 a_x - c_x)^2 + (Z_1 a_y - c_y)^2 = (Z_2 b_x - c_x)^2 + (Z_2 b_y - c_y)^2, \quad (3.24.2)$$

$$\varphi_{\Delta} = \arctan\left(\frac{Z_2 b_y - c_y}{Z_2 b_x - c_x}\right) - \arctan\left(\frac{Z_1 a_y - c_y}{Z_1 a_x - c_x}\right). \quad (3.24.3)$$

Unknowns Z_1 and Z_2 may be calculated from (3.24.1-2). Therefore, (3.24.1) may be solved for Z_2 ,

$$Z_2 = Z_1 \frac{a_z}{b_z}, \quad (3.25)$$

this will be used in (3.24.2), leading to

$$Z_1 = 2 \frac{\left(a_x c_x + a_y c_y - b_x \frac{a_z}{b_z} c_x - b_y \frac{a_z}{b_z} c_y\right)}{\left(a_x^2 + a_y^2 - \left(b_x \frac{a_z}{b_z}\right)^2 - \left(b_y \frac{a_z}{b_z}\right)^2\right)}. \quad (3.26)$$

By these results for Z_1 and Z_2 , angle φ_{Δ} follows from (3.24.3). World point W will be calculated as before using (3.13).

4 EXPERIMENTAL RESULTS

The complete surface reconstruction process has been implemented and tested on Sun Sparc Stations. The tasks are separated into several modules like camera calibration, motion vector calculation and depth recognition.

4.1 Real-Time Aspects

The non-recurring calibration takes less than one second for the used set of 75 calibration points, including the optimization for lens distortion.

The reconstruction algorithms were implemented straightforward, following Theorems 3.1, 3.2 or 3.2*, and having time complexity in class $\mathbf{O}(N)$, where N denotes the number of motion vectors (u, v) which are not equal to zero. For each vector, the same process is repeated, i.e. vectors are considered to be independent features. Assume that a vector field is already given for 512×512 pixels. Then, if φ_{Δ} is known, the computation of depth for all 512×512 pixels takes less than one second. If φ_{Δ} is unknown it takes twice as much.

For a sequence of several images, for all vectors describing motion of the same surface point, theoretically only one vector would be sufficient for calculating the depth of this surface point. Because of digitized disparities and erroneous vector fields, the average of depth results for all given vectors would improve the reconstruction quality, e.g. by finer depth resolution. But, for improved algorithmic complexity, all vectors related to the same surface point may be combined sequentially, and depth may be calculated for this resulting vector at once reducing the runtime.

Both algorithms, with known rotation angle and without known angle, could be highly optimized by lookup tables and a massive parallel implementation would be possible. Thus, starting with motion vectors, reconstruction of depth can be realized in real time.

Computation of corresponding points, features, motion fields etc. will depend upon image data and selected methods. For real-time realization of correspondence mappings, parallel implementations of matching strategies may be the solution, see [Koschan/Rodehorst 1995].

4.2 Optimization of Calibration Object

A *calibration object* has to be used for performing camera calibration which physically represents a specific geometric configuration of a finite set of *calibration points* (dots on the calibration object). Following [Tsai 1986], two cases were considered: all calibration points are coplanar, or may be in arbitrary position in several *calibration planes*. For improving time complexity of

calibration processes, the number of calibration points has to be minimized such that calibration is still "close to optimum".

The non-coplanar case did prove to be practically "more suitable". Therefore, synthetic calibration points were assumed and used for a quantitative evaluation of calibration, see Tab. 1. Real camera data from the product data sheet were used. For measuring errors, the distances were calculated between assumed calibration point positions and positions after backprojection, using calibration results. Then, these distances may be used to define different error measures. A normal distribution of errors in positioning synthetic calibration points was used during the repeated process of calculating calibration errors, see Tab. 2.

Calibration parameter	Synthetic dataset	Calibration results
Translation	T_x	80.0 mm
	T_y	100.0 mm
	T_z	8000.0 mm
Rotation	Yaw	60.0 deg
	Pitch	20.0 deg
	Roll	-5.0 deg
Focal length B	32.0 mm	32.016 mm
Scaling factor s_x	1.020	1.020
Distortion coefficients	κ_1	$-1.0 \cdot 10^{-5} \frac{1}{\text{mm}^2}$
	κ_2	$-1.0 \cdot 10^{-5} \frac{1}{\text{mm}^2}$
		$-1.5 \cdot 10^{-7} \frac{1}{\text{mm}^2}$
		$-2.7 \cdot 10^{-6} \frac{1}{\text{mm}^2}$

Table 1: Calibration results of a synthetic dataset under noisy conditions. The 3D world coordinates varies 0.5 mm and the image coordinates 0.3 pix.

Error measures	Fig. 3a	Fig. 3b
Mean of Projection	0.564 pix	0.459 pix
Mean of Backprojection	0.268 mm	0.232 mm
Standard deviation of Projection	0.337 pix	0.260 pix
Standard deviation of Backprojection	0.141 mm	0.163 mm

Table 2: Error measures for real camera calibration

In general, increasing the number of calibration points leads to improved calibration results. But, about 20 calibration points in a single calibration plane are already close to optimum. For the number of calibration planes, minimum error was measured at about 4 calibration planes. Here, error increases slightly again for more than 4 planes.

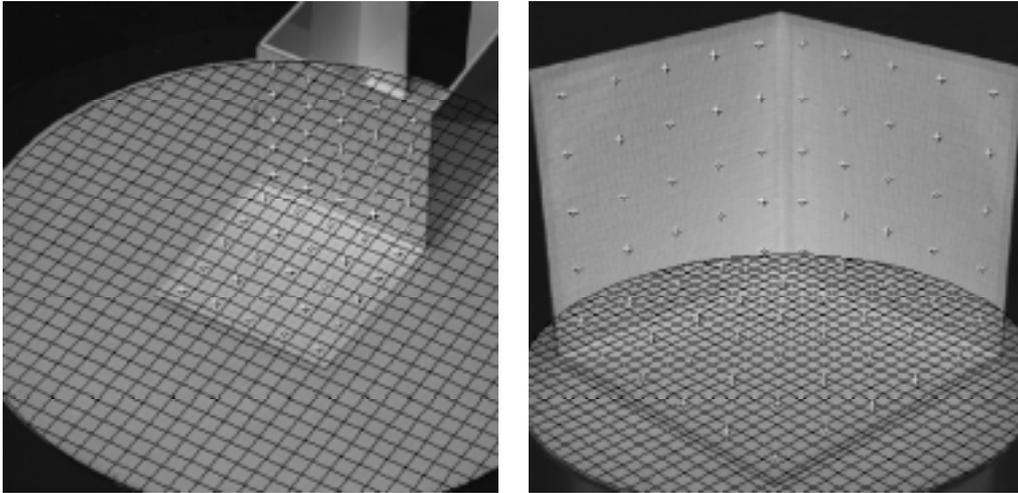


Figure 2: Calibration object with two faces not covering the image (left). Object with three faces with good covering of the image (right). Calibration points were detected, calibration was performed and then these points were backprojected into the image (crosses). Based on calibration, also the rotating disc was backprojected (grid texture). In the left image, an error is visible, where the grid does not match the border of the rotating disc.

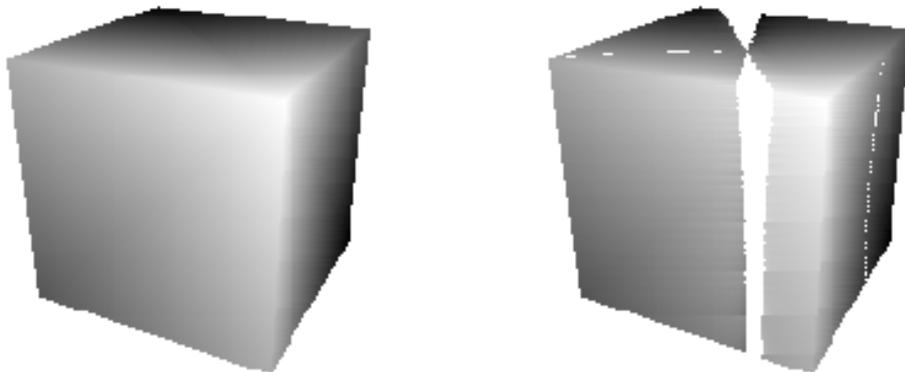


Figure 3: Ideal depth map (left) of a cube and depth map as calculated using the straightforward solution method described in Section 3.2.1.

For the calibration of real objects an "open cube" was used consisting of three square faces, orthogonal by pairs, with 25 calibration points on each face. For measuring positions of calibration points in digitized images, at first each calibration point could be detected as a circular shaped region with diameter of about 4 pixel, then moments were used to calculate the centroid of such a region.

Such centroids were used as subpixel accurate positions of projected calibration points in the image plane. For obtaining best calibration results, the calibration object should cover the complete image, see Fig. 2.

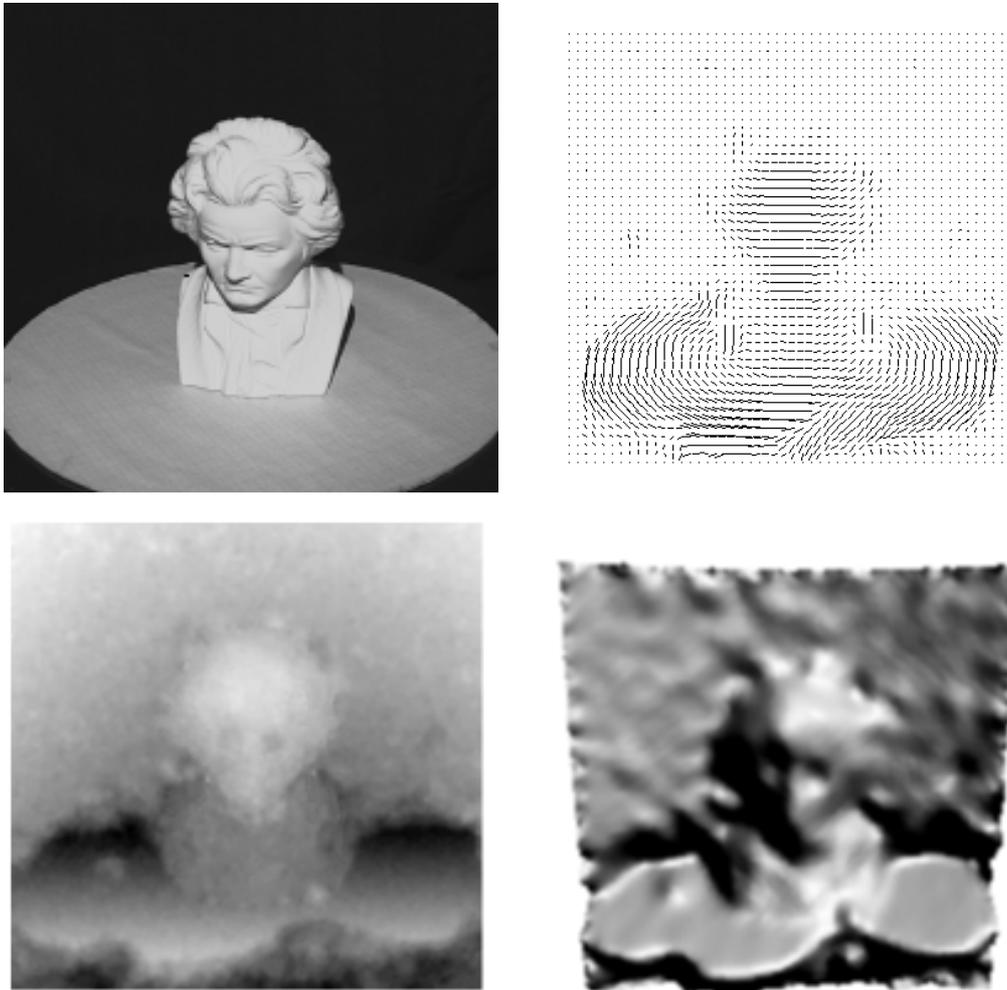


Figure 4: A plaster statue on the rotating disc, optical flow field calculated by the method of [Anandan 1987], reconstructed depth map using these vectors, and 3-D visualization of this depth map.

4.3 Results of Surface Reconstruction

In experiments, the method based on knowledge of rotation angle did prove to be robust for any 2-D motion of point C_1 into C_2 within the image plane. But, both approaches dealing with the case of unknown rotation angle did not work if the direction of the motion vector (u, v) of point C_1 into C_2 is "nearly parallel" to image rows or image columns, i.e.

$$\frac{u}{v} \gg 1 \text{ or } \frac{u}{v} \ll 1. \quad (4.1)$$

This is illustrated by the missing depth values (white stripes) in the right picture of Fig. 3. So far, no mathematical explanation is available for this "bad behavior" of methods using no knowledge about rotation angle.

As final conclusion, a two-step procedure is proposed. First the method with unknown rotation angle is used for calculating the unique (!) rotation angle:

For all corresponding pairs C_1 and C_2 in the motion vector field the rotation angle is calculated. Then, for all these resulting angles a certain mean value is derived as unique rotation angle φ_Δ .

Then, for this angle the method with known rotation angle may be used to calculate depth values for all pairs of corresponding points C_1 and C_2 .

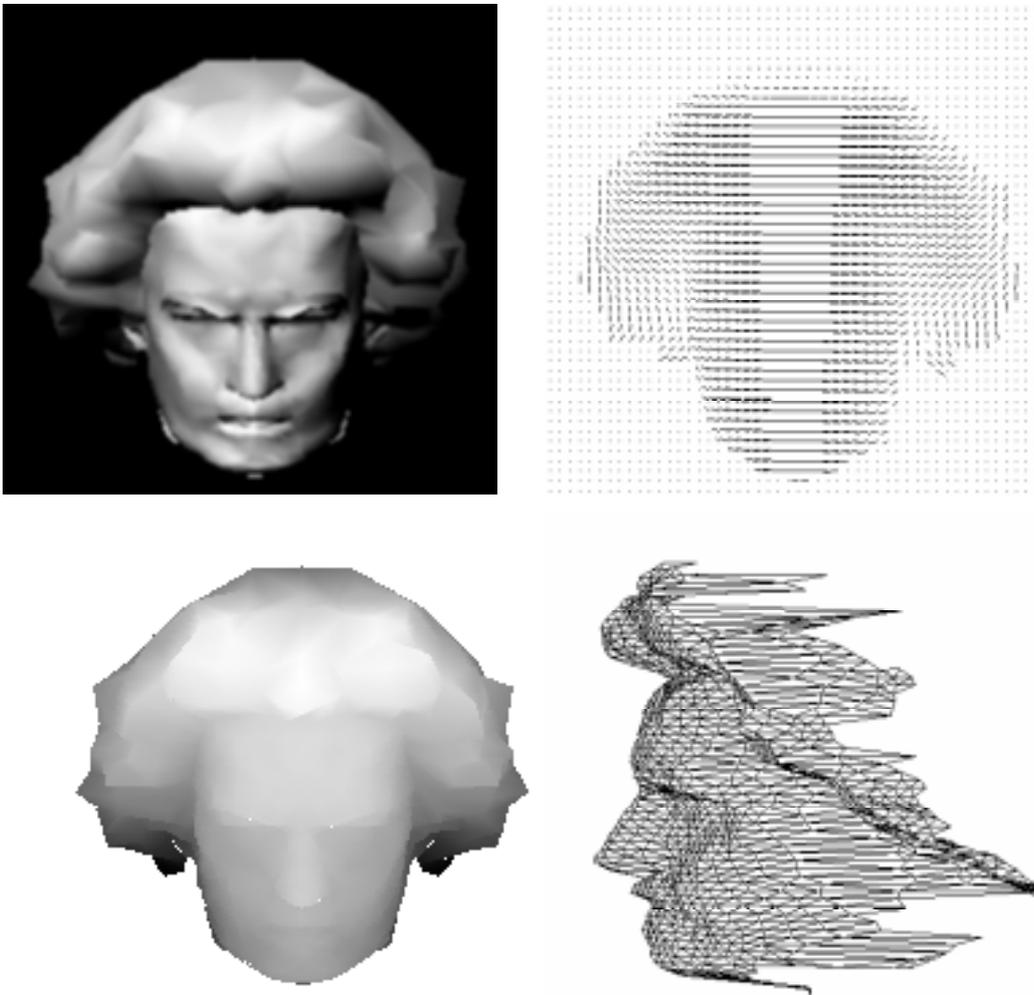


Figure 5: Synthetic 3-D object, its motion vector field simulating the rotating disc, reconstructed depth map, and 3-D visualization of these depth values.

In Fig.4 it is illustrated what quality of shape or depth calculation was obtainable in our experiments. Comparing many point-based differential methods for computing optical flow, the method by [Anandan 1987] was chosen to behave best under the given (!) conditions. For 3-D visualization of the depth map, depth values were smoothed.

If (!) motion vector fields would be correct, then the reconstruction method described in Section 3 will lead to exact object surfaces. Fig. 5 illustrates this for a synthetic object where the motion vector field (simulating the rotating disc) was available. By adding noise to these ideal motion vector fields it becomes clear that small distortions will have great influence on the reconstructed surfaces. Altogether, optical flow computation will not lead to complete surface reconstruction within acceptable limits of quality. But, for computing singular point positions in 3-D or for generating some "rough drafts", this method might be of some practical value.

5 CONCLUSIONS

The used method for camera calibration may be highly recommended because of its accuracy and robustness. It was used also for surface reconstruction based on structured light (where reconstruction is possible within acceptable limits of quality). It is suggested to extend this method for estimating directions to point light sources (this is necessary for photometric procedures).

Viewing polyhedral objects (boxes etc.) on the disc, for sparse surface points as corners or points on edges which are about orthogonal to the direction of rotation, very accurate measurements of 3-D positions could be obtained.

For combining sparse but accurate surface point positions (*fixation points*) with rough surface drafts, as resulting, e.g. from photometric methods, combinations in Fourier space of high frequencies of photometric results, and low frequencies of motion analysis results may be suggested.

REFERENCES

1. J. Aloimonos, D. Shulman: *Integration of Visual Modules: An Extension of the Marr Paradigm*. Academic Press, Boston, 1990.
2. P. Anandan: Ph.D., COINS TR 87-21, Univ. of Massachusetts, Amherst, 1987.
3. J.L. Barron, D.J. Fleet, S.S. Beauchemin: Performance of optical flow techniques. *Int. J. Computer Vision* **12** (1994), pp. 43 - 77.

4. R.T. Frankot, R. Chellappa: A method for enforcing integrability in shape from shading algorithms. *IEEE Trans. Patt. Anal. Mach. Intell.* **10** (1988), pp. 439-451.
5. P. Handschack, R. Klette: Evaluation of differential methods for image velocity measurement. *Computers & Artificial Intelligence* (1994), to appear.
6. B.K.P. Horn: *Robot Vision*. McGraw-Hill, New York 1986.
7. K. Kanatani: *Group-Theoretical Methods in Image Understanding*. Springer, Berlin 1990.
8. R. Klette, V. Rodehorst: Algorithms for shape from shading, lighting direction and motion. *Proceed. 5th Int. Conf. CAIP'93, Budapest, September 1993* (eds.: D. Chetverikov, W.G. Kropatsch), Springer, Lecture Notes in Computer Science **719**, pp. 420 - 427.
9. A. Koschan, V. Rodehorst: Towards real-time stereo employing parallel algorithms for edge-based and dense stereo matching (1995, submitted).
10. S. Maybank: *Theory of Reconstruction from Image Motion*. Springer, Berlin 1993.
11. K. Schlüns, O. Wittig: Photometric stereo for non-Lambertian surfaces using color information, *Proc. 7th Int. Conference on Image Analysis and Processing, Monopoli, Italy, Sept. 20-22, 1993*, pp. 505-512.
12. R.Y. Tsai: An efficient and accurate camera calibration technique for 3D machine vision. *Proc. IEEE Conf. Computer Vision and Pattern Rec.* 1986, pp. 364 - 374.
13. R.Y. Tsai: A versatile camera calibration technique for high accuracy 3D machine vision metrology using off-the-shelf TV camera and lenses. *IEEE J. Rob. and Automat.* (1987), pp. 323 - 344.