PROCEEDINGS OF SPIE

The effects of pattern screen surface deformation on deflectometric measurements: a simulation study

Allgeier, Stephan, Gengenbach, Ulrich, Köhler, Bernd, Reichert, Klaus-Martin, Hagenmeyer, Veit

SPIE.

The effects of pattern screen surface deformation on deflectometric measurements – a simulation study

Stephan Allgeier, Ulrich Gengenbach, Bernd Köhler, Klaus-Martin Reichert, and Veit Hagenmeyer

Institute for Automation and Applied Informatics, Karlsruhe Institute of Technology (KIT), Postfach 3640, 76021 Karlsruhe, Germany

ABSTRACT

Phase-measuring deflectometry (PMD) is an optical inspection technique for full-field topography measurements of reflective sample surfaces. The measurement principle relies on the analysis of specific patterns, reflected at the sample surface. Evaluation algorithms often model the respective pattern screen as a planar light source. However, the 32″ pattern screen in our inspection setup exhibits a central bulge of its surface of about 2–3 mm. This paper presents a simulation framework for PMD to evaluate the effects of a deformed screen surface. The idea is to simulate image data acquired with screen surface deformations and to examine the effects on the PMD evaluation results. The simulated setup consists of a 32″ pattern screen with an adjustable central bulge height of 0–3 mm and two cameras with a field of view (FOV) of approximately 225 mm by 172 mm on the sample surface. A first experiment examines the reconstruction errors for a planar sample surface if the reconstruction algorithm uses perfect calibration data (i.e. the same parameters used for the simulated image acquisition). The reconstructed surfaces exhibit a tilt with a maximum height difference of 174 µm across the FOV. A second experiment repeats the reconstruction process of the same sample surface, using camera parameters determined in a simulated calibration process. The resulting surfaces possess irregular, wave-like errors with amplitudes of up to 9 µm in the FOV. The presented simulation results reveal the accuracy limits if a deformation model of the pattern screen is not explicitly included in the reconstruction process.

Keywords: surface inspection, deflectometry, simulation, printed electronics, calibration

1. INTRODUCTION

Printing of functional inks is a technology with great potential to realize innovative smart electronic devices. In particular, flexible systems based on inks made of novel materials e.g. nanoparticles on foil substrates offer new opportunities in many applications.[1] The printing process however depends on a number of parameters related to the ink properties (e.g. ink viscosity, surface energy, nanoparticle load), substrate properties (surface texture, surface energy) and to the printing equipment proper. Aerosol-jet printers have a focal length of up to 5 mm and are thus very tolerant with respect to nonplanar substrates. Ink-jet printers on the other hand require the nozzle stand-off distance to be kept constant, typically at about 0.5 mm, to minimize droplet placement errors. A significant effort goes into the clamping of the substrate to keep variations in the nozzle stand-off distance low.[2] While pristine substrates usually are planar, this state frequently changes in multi-step printing processes where printing steps and thermal sintering or curing steps alternate. These thermal processes, e.g. oven processes at 130 °C to sinter silver nanoparticle inks, can lead to substrate shrinkage and warpage.[3] This results on the one hand from internal strain in the foil substrates due to their fabrication process. On the other hand the printed structures may also lead to local strain and thus to local warpage. These local or global substrate warpages may alter the stand-off distance of the printing nozzle. Thus after the first thermal process the substrate cannot be regarded as planar any more. Hence, to ensure a high printing quality by adapting the clamping of the substrate or the nozzle stand-off distance, the substrate warpage has to be measured with high precision after each thermal process.

Scanning processes such as optical profilometry allow characterization of substrate warpage with high precision but are time consuming if high lateral resolution is required. Deflectometry on the other hand is much more

---

Corresponding author: Stephan Allgeier, stephan.allgeier@kit.edu

time-efficient for this task, as it takes only few camera images for full-field topography inspection of reflective surfaces. The measurement principle relies on imaging known patterns as they are reflected at the sample surface. Qualitative information on the sample surface topography can be readily inferred by a visual analysis of the reflected pattern, for instance by identifying deviations from an expected pattern image.[4,5] If the relative geometrical arrangement of the measurement setup, i.e. the pattern and the camera, is accurately known, it is also possible to reconstruct a topographical representation of the sample surface. For this, the pattern – or more often the series of patterns that are presented and imaged in temporal succession – must be specifically designed to encode information of the point of origin of the light that is finally measured at a certain camera pixel. In phase-measuring deflectometry (PMD), horizontal and vertical sinusoidal fringe patterns that are shifted in several discrete steps serve this purpose; the phase information effectively encodes the pattern coordinates.[6–9] Being able to associate a pattern coordinate to every camera pixel allows the inference of potential light paths from the points of origin of the light on the pattern to the corresponding camera pixels. The information of potential light paths is equivalent to a field of potential normal vectors of the surface in the measurement space. Finally, with some additional constraints, e.g. the assumption of a differentiable and intergrable surface[10] or the requirement to conform to the measurement of a second camera in stereo deflectometry,[6] the surface topography can be reconstructed from this gradient field.[11]

The light path analysis requires a mathematical model of the measurement setup, specifically of the pattern generator and the cameras, which are often modeled as pinhole cameras. The model parameters need to be determined by a priori knowledge or a calibration process.[6,12–16]

The majority of PMD measurement setups described in the scientific literature use a computer screen to present the patterns.[7,9,14,15,17,18] For the purpose of determining the origin of the signal measured in a camera pixel, they are usually modeled in the PMD evaluation algorithms as a planar light source with a very regular grid of light emitting square pixels. However, the assumption of a planar screen surface may not always be appropriate, particularly when considering measurement setups with the screen mounted at an inclined orientation. For instance, a $32''$ screen in our in-house setup, which is mounted at an angle of approx. 45° with respect to the horizontal base plane, manifests a central outward bulge of the screen surface of approx. 2–3 mm. The screen surface topography strongly depends on the mounting angle. It approaches a planar configuration when tilted towards an upright orientation.

In this contribution, we present a simulation framework for PMD with deformable surface representations for both the pattern screen surface and the sample surface, and a camera model to simulate the image acquisition process by tracing the camera "view rays" to the pattern screen. The aim is to create simulated image data of a setup with screen surface deformations and to examine the effects on the PMD evaluation results when evaluation algorithms are applied that generally assume a planar screen.

Section 2 provides a detailed overview of the simulation framework, including the sample and screen surface representations, the used camera model and the raytracing process. Sections 3 and 4 present the setup and results of two experiments that use simulated image data. They both examine the effects of a non-planar pattern screen surface on the reconstruction results, using either perfect a-priori knowledge of the simulated imaging setup (Sec. 3) or simulated calibration results (Sec. 4) for the PMD evaluation process. In Sec. 5, we discuss the limitations of the simple planar screen model with respect to the evaluation results and derive conclusions for further developments.

2. SIMULATION FRAMEWORK

The purpose of the simulation framework is to simulate the image acquisition process during a deflectometric measurement, i.e. to generate image datasets that correspond to the image data acquired during such a measurement. The framework supports the following components (cf. Fig. 1):

- One or more cameras, modeled as pinhole cameras with additional lens distortion terms,

- a pattern screen, modeled as a spline-based surface, and

- a sample surface, also modeled as a spline-based surface.
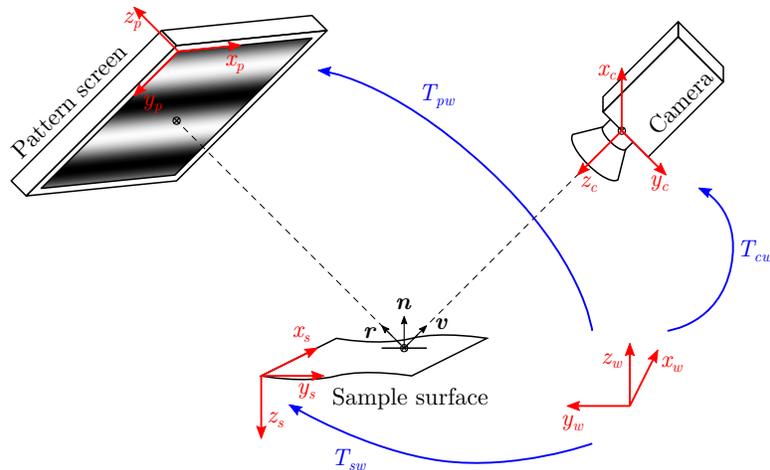
Figure 1. The simulation framework implements models for the camera and the pattern screen of the simulated deflecto-metric measurement system and for the sample surface. Each component model is described with regard to an individual coordinate system (shown in red). The geometric relation between all components is specified by the coordinate transformations (shown in blue) between the individual component coordinate systems and a common world coordinate system (x_w, y_w, z_w).

The simulation of the imaging process focuses entirely on the effects of screen surface deformations. Its function can be summarized as follows: from the implemented camera model, each camera pixel possesses a well-defined corresponding view ray. The view ray is extended towards the reflective sample surface, where the incident light ray is calculated according to the law of specular reflection, i.e. the surface normal bisects the angle between the incident light ray and the camera view ray. The incident light ray is then traced back onto the screen surface. The pattern intensity at this screen location is then simply used as the intensity value measured at the camera pixel. Other real-world effects such as non-specular reflection components, limited depth of field, measurement noise, angle-dependent light emission characteristics of the screen etc. are not taken into account in the present examinations. The following paragraphs provide a brief mathematical description of the simulation process, including the model parameters that fully define the simulated measurement setup.

2.1 Geometric Model

The geometrical relations of the setup components are defined by their respective three-dimensional Cartesian coordinate systems. They are described uniquely by the 4×4 coordinate transformation matrices T_{cw} (or $T_{c_i w}$ in case of several cameras), T_{pw} and T_{sw} that transform a point or vector given in a common reference or world coordinate system (in homogeneous coordinates) into the respective component coordinate system (cf. Fig. 1). Each coordinates transformation is defined by six parameters: three for the translation vector and three rotation components (we use three Givens rotation angles for that purpose[19,20]). The transformation matrices are inverted and combined to provide the transformation matrix from any origin coordinate system (index o) to any destination coordinate system (index d) by

$$T_{do} = T_{dw} \cdot T_{ow}^{-1}. \tag{1}$$

2.2 Pinhole Camera Model

The pinhole camera model is well established and frequently used in deflectometric applications. An exhaustive introduction can be found in Ref. 19. In short, the projection of a point $\vec{p}_{(c)} = (x_c, y_c, z_y)^T_{(c)}$ given in the camera coordinate system onto the corresponding image pixel coordinates $\vec{p}_{(i)} = (x_i, y_i)^T_{(i)}$ of the resulting image is modeled as a three-step process.[20] Note that throughout the text, subscripts in parentheses are meant to denote the coordinate system in which a vector is given; (c) and (i) denote the camera and image coordinate systems, respectively.
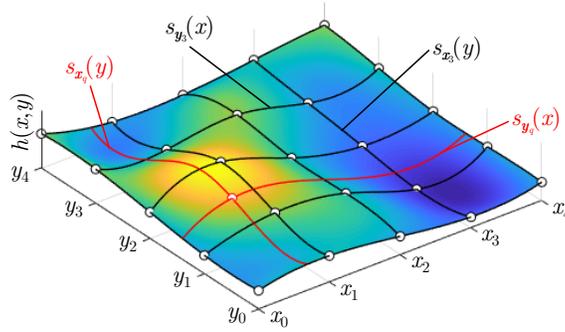
Figure 2. Model for a non-planar surface, based on interpolating cubic spline functions. The height values $h(x, y)$ at the grid of support points (at coordinates $x_0 \ldots x_4$ and $y_0 \ldots y_4$) are provided. Surface points along the grid lines are directly interpolated by the pre-calculated base spline functions s_{x_i} and s_{y_i} (black), surface points at arbitrary coordinates (x_q, y_q) are interpolated in a two-step process by intermediate spline functions s_{x_q} and s_{y_q} (red, see text).

The first step projects $\vec{p}_{(c)}$ onto the image plane (i.e. the plane perpendicular to the optical axis and a distance equal to the focal length fl in front of the pinhole):

$$\vec{p}_{ip} = \left(fl\frac{x_c}{z_c}, fl\frac{y_c}{z_c}, fl \right)^T = fl \left(x_{ip}, y_{ip}, 1 \right)^T . \tag{2}$$

The second step applies lens distortion effects[21] to the normalized image coordinates x_{ip} and y_{ip}, parameterized by the radial lens distortion coefficients k_1 and k_2 and the tangential lens distortion coefficients p_1 and p_2:

$$\left(\begin{array}{c} x'_{ip} \\ y'_{ip} \end{array} \right) = \left(\begin{array}{c} x_{ip} \\ y_{ip} \end{array} \right) + \left(\begin{array}{c} x_{ip} \left(k_1 r^2 + k_2 r^4 \right) \\ y_{ip} \left(k_1 r^2 + k_2 r^4 \right) \end{array} \right) + \left(\begin{array}{c} 2p_1 x_{ip} y_{ip} + p_2 \left(r^2 + 2x_{ip}^2 \right) \\ 2p_2 x_{ip} y_{ip} + p_1 \left(r^2 + 2y_{ip}^2 \right) \end{array} \right) \tag{3}$$

where $r^2 = x_{ip}^2 + y_{ip}^2$ is the squared distance (in normalized coordinates) of the projected point on the image plane from the optical axis. The third step is the transformation of the distorted coordinates to pixel coordinates. The transformation is described by the camera intrinsics matrix K that contains the focal length parameters fl_x and fl_y (the focal length described in direction-dependent image pixel units) and the principal point coordinates pp_x and pp_y (also specified in image pixel units):

$$\vec{p}_{(i)} = \left(\begin{array}{ccc} fl_x & 0 & pp_x \\ 0 & fl_y & pp_y \end{array} \right) \left(\begin{array}{c} x'_{ip} \\ y'_{ip} \\ 1 \end{array} \right) = K \left(\begin{array}{c} x'_{ip} \\ y'_{ip} \\ 1 \end{array} \right) . \tag{4}$$

Finally, the view ray associated with a given image pixel $\vec{p}_{(i)}$ is defined as the line that passes through the pinhole (i.e. the origin of the camera coordinate system) and has the direction vector $(x_{ip}, y_{ip}, 1)_{(c)}^T$. It is readily calculated by inverting the last two steps of the imaging process, the second one numerically. As the simulation only requires the view rays associated with the integer-valued pixel coordinates given by the image sensor size, they can be pre-calculated and stored for a given simulated camera.

2.3 Pattern Screen and Sample Surface Model

The mathematical representations for both the screen surface and the reflective sample surface support non-planar geometries. The simulation framework models them with standard (one-dimensional) interpolating cubic spline functions[22] over a two-dimensional grid of support points (cf. Fig. 2). The surface representations are defined in their respective coordinate systems, i.e. the pattern screen surface in the pattern screen coordinate system and the sample surface in the sample coordinate system. The following description of a surface model is applicable to both the pattern screen and the sample without changes. Later sections will amend the resulting

height function h with subscripts (s) or (p), to differentiate between the sample and the pattern screen surface, respectively.

The support points of a surface are given as height values over a regular, rectangular, axis-parallel grid of points in the x-y-plane of their respective coordinate system. They are therefore fully specified by the sets X_s and Y_s of (ordered) x- and y-coordinates of the support points, respectively,

$$X_s = \{x_0, \ldots, x_m \mid x_0 < \ldots < x_m\} \tag{5}$$
$$Y_s = \{y_0, \ldots, y_n \mid y_0 < \ldots < y_n\} \tag{6}$$

and by a matrix

$$Z_s = \begin{bmatrix} z_{0,0} & \cdots & z_{0,m} \\ \vdots & \ddots & \vdots \\ z_{n,0} & \cdots & z_{n,m} \end{bmatrix} \tag{7}$$

containing the height values of the surface at the control points. Note that, of course, positive and negative height values are interpreted with respect to the orientation of the z-axis, so the orientation of the respective coordinate system has to be kept in mind when specifying concave or convex surfaces. In order to avoid the need for extrapolation outside of the grid, the smallest and largest coordinate values in X_s and Y_s implicitly define the bounds of the (rectangular) surface.

The regular grid of control points decomposes the surface into separate surface patches, with the patch borders of each patch forming a rectangle when projected at the x-y-plane. The surface sections along each vertical or horizontal grid line, i.e. at each of the x- or y-coordinates in X_s and Y_s, respectively, are given by the one-dimensional interpolation cubic spline function uniquely defined by the control points along the respective grid line. Let $s_{x_0}(y), \ldots s_{x_m}(y)$ be the (vertical) spline functions interpolating the surface sections at the x-coordinates in X_s and $s_{y_0}(x), \ldots s_{y_n}(x)$ the (horizontal) spline functions interpolating the surface sections at the y-coordinates in Y_s (all using the "not-a-knot" end condition, i.e. forcing the third derivative at the second and second-to-last control points to be continuous[23]). These spline functions provide a (sparse) set of surface sections, as well as the first and second partial derivatives in the direction of the spline functions at every point on these sections (cf. Fig. 2).

The interpolated surface height $h(x, y)$ at arbitrary query coordinates (x_q, y_q) with $x_0 \leq x_q \leq x_m$ and $y_0 \leq y_q \leq y_n$ is derived from the given support grid as follows. If the coordinates lie on one of the grid lines, i.e. $x_q = x_i$ for any $x_i \in X_s$ or $y_q = y_i$ for any $y_i \in Y_s$, the surface height $h(x_q, y_q)$ is given by the function value of the respective spline function along the grid line. In all other cases, we calculate the one-dimensional cubic spline function $s_{y_q}(x)$ to interpolate the surface section at y_q, using the height values $s_{x_0}(y_q), \ldots, s_{x_m}(y_q)$ as control points. The surface height is then given by the function value

$$h(x_q, y_q) = s_{y_q}(x_q). \tag{8}$$

In addition to the surface position, we will also require the surface gradient later on. In order to obtain both partial derivatives $\frac{\partial h}{\partial x}$ and $\frac{\partial h}{\partial y}$ of the surface function at (x_q, y_q), we further determine the cubic spline function $s_{x_q}(y)$ (that interpolates the surface section at x_q) using the height values $s_{y_0}(x_q), \ldots, s_{y_n}(x_q)$ as control points.

2.4 Simulated Imaging Process

Based on the models described in Sec. 2.1–2.3, the imaging process can be simulated. As mentioned above, the idea is to trace a camera view ray to the pattern screen – or more physically correct, to back-trace the incident light ray to its origin – in order to find the resulting intensity values for each camera pixel.

Given a camera pixel of interest, let $\vec{v}_{(c)}$ be the direction vector and $\vec{o}_{(c)} = (0, 0, 0)_{(c)}^T$ the origin of the associated view ray, both given in the camera coordinate system. Applying the transformation matrix T_{sc} transforms them into the sample coordinate system, yielding $\vec{v}_{(s)} = (x_v, y_v, z_v)_{(s)}^T$ and $\vec{o}_{(s)} = (x_o, y_o, z_o)_{(s)}^T$. The

point of intersection $\vec{p}_{(s)} = (x_p, y_p, z_p)_{(s)}^T$ of the view ray with the sample surface is calculated iteratively. The initial step is to find the intersection of the view ray with the x-y-plane,

$$z_p^{(0)} = 0, \tag{9}$$

$$t^{(0)} = \frac{z_p^{(0)} - z_o}{z_v}, \tag{10}$$

$$x_p^{(0)} = x_o + t^{(0)} x_v, \tag{11}$$

$$y_p^{(0)} = y_o + t^{(0)} y_v. \tag{12}$$

The iteration then proceeds with the following steps until convergence, using the spline-based surface function $h_{(s)}(x, y)$ from Sec. 2.3:

$$z_p^{(i)} = h_{(s)} \left(x_p^{(i-1)}, y_p^{(i-1)} \right), \tag{13}$$

$$t^{(i)} = \frac{z_p^{(i)} - z_o}{z_v}, \tag{14}$$

$$x_p^{(i)} = x_o + t^{(i)} x_v, \tag{15}$$

$$y_p^{(i)} = y_o + t^{(i)} y_v. \tag{16}$$

Once the point $\vec{p}_{(s)}$ is known, the raytracing process continues along the reflected view ray $\vec{r}_{(s)}$, which is calculated as

$$\vec{r}_{(s)} = \vec{v}_{(s)} - 2 \frac{\vec{v}_{(s)} \vec{n}_{(s)}}{\vec{n}_{(s)} \vec{n}_{(s)}} \vec{n}_{(s)} \tag{17}$$

according to the law of reflection. Here, $\vec{n}_{(s)}$ is the surface normal vector at the point of intersection. It is defined as $\vec{n}_{(s)} = (-d_x, -d_y, 1)^T$, where d_x and d_y are the partial derivatives of the surface height function $h_{(s)}(x, y)$ at $\vec{p}_{(s)}$. Note that $\vec{r}_{(s)}$ points away from the surface if $\vec{v}_{(s)}$ points towards it.

A coordinate system transformation of $\vec{r}_{(s)}$ and $\vec{p}_{(s)}$ using T_{ps} provides the direction vector of the reflected view ray $\vec{r}_{(p)}$ and its origin point $\vec{p}_{(p)}$ with respect to the pattern screen coordinate system. Finding the point of intersection $\vec{q}_{(p)} = (x_q, y_q, z_q)_{(p)}^T$ of the ray with the pattern screen surface is entirely analogous to the intersection of the original view ray with the sample surface.

The coordinates (x_q, y_q) are interpreted as pattern coordinates. Assuming that the pattern intensity function uses the same domain as the simulated camera image, for example both using the value range $[0, 255]$, the pattern intensity at those coordinates can be directly used as the intensity value measured at the corresponding camera pixel. Note that the simulated imaging process of an entire pattern sequence does not require the view ray tracing to be repeated multiple times. As the result of the ray tracing is a mapping of camera pixels and pattern screen locations, this mapping can be reused as long as the specification of the camera, screen and sample setup remains unchanged.

The simulated images are finally saved as image files, which can be used for deflectometric analyses in place of images from real measurements.

2.5 Experimental Setup

Table 1 provides the parameters of the simulated deflectometric measurement setup (cf. Fig. 3) that is used for the experiments described in the following two sections. It roughly resembles our real measurement setup and comprises two ideal pinhole cameras (i.e. all lens distortion coefficients are 0), arranged symmetrically with a distance of 180 mm in between. Both are directed at an angle of 45° at the sample surface, where they capture a field of view (FOV) of approximately 225 mm by 172 mm. A 32″ screen at the opposite side of the simulated setup is also mounted at a 45° angle with respect to the sample surface.

In order to quantify the influence of screen surface deformations on the calibration and surface reconstruction processes, the screen surface is modeled by a 3×3 grid of control points. The eight border points are all given a

Table 1. Model parameters for the simulation experiments

Camera internal parameters (both cameras)								
Focal length [image pixels]	fl_x	fl_y			4400	4400		
Principal point [image pixels]	pp_x	pp_y			960	600		
Lens distortion coefficients	k_1	k_2	p_1	p_2	0	0	0	0
Image size [image pixels]					1920	1200		
Left camera external parameters								
Coordinate system origin [mm]	$x_{0,(w)}$	$y_{0,(w)}$	$z_{0,(w)}$		−90	295	−295	
Coordinate system rotation [°]	$\alpha_{x,(w)}$	$\beta_{x,(w)}$	$\gamma_{x,(w)}$		45	10	10	
Right camera external parameters								
Coordinate system origin [mm]	$x_{0,(w)}$	$y_{0,(w)}$	$z_{0,(w)}$		90	295	−295	
Coordinate system rotation [°]	$\alpha_{x,(w)}$	$\beta_{x,(w)}$	$\gamma_{x,(w)}$		45	−10	−10	
Pattern screen surface parameters								
Support grid coordinates (x) [mm]	$X_{s,(p)}$				0	354	708	
Support grid coordinates (y) [mm]	$Y_{s,(p)}$				0	199	398	
Support grid central height [mm]	Δh				variable (0, 1, 2, 3)			
Pattern screen external parameters								
Coordinate system origin [mm]	$x_{0,(w)}$	$y_{0,(w)}$	$z_{0,(w)}$		−354	−78	−400	
Coordinate system rotation [°]	$\alpha_{x,(w)}$	$\beta_{x,(w)}$	$\gamma_{x,(w)}$		135	0	0	
Sample surface external parameters								
Coordinate system origin [mm]	$x_{0,(w)}$	$y_{0,(w)}$	$z_{0,(w)}$		−200	−200	0	
Coordinate system rotation [°]	$\alpha_{x,(w)}$	$\beta_{x,(w)}$	$\gamma_{x,(w)}$		0	0	0	
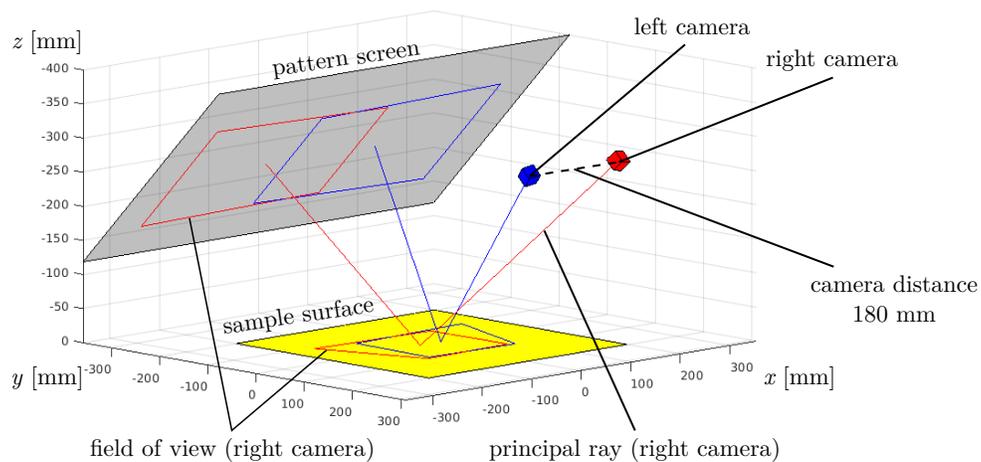
Figure 3. Draft of the simulated deflectometric inspection system that is used for the experiments in Sec. 3 and 4.

Table 2. Characteristics of surface reconstruction results in experiment 1

Screen bulge height Δh	0 mm		1 mm		2 mm		3 mm	
Camera (L = left, R = right)	L	R	L	R	L	R	L	R
Max. height difference [µm]	0.0	0.0	49.7	57.8	99.6	115.8	149.5	174.1
Max. deviation from best-fit-plane [µm]	0.0	0.0	14.6	10.8	29.1	21.7	43.7	32.6
Best-fit plane tilt angle [µrad]	0.0	0.1	278.5	259.8	557.4	520.3	836.6	781.5

height value of 0 to model a rigid screen frame that coincides with the x-y-plane of the screen coordinate system. The height value Δh of the central control point is varied in the experiments. Although the simulation framework supports non-planar sample surfaces, as described in Sec. 2.3, the experiments presented in the following two sections uniformly use a planar mirror model.

The surface reconstruction is performed using a simple stepwise integration approach.[12] Starting from a (given) initial surface point, the surface is iteratively expanded in all directions, driven by the surface normal at the borders of the current reconstruction. Let $\vec{s}_{(p)}$ be a sample surface point, given in the pattern screen coordinate system. The surface normal vector at $\vec{s}_{(p)}$ is the bisection vector between the camera view ray vector $\vec{v}_{(p)} = \vec{s}_{(p)} - \vec{o}_{(p)}$ associated with that point and the incident light ray vector $\vec{r}_{(p)} = \vec{s}_{(p)} - \vec{p}_{(p)}$. The incident light ray is calculated as follows: If (x_i, y_i) are the camera image coordinates onto which $\vec{s}_{(p)}$ is projected (cf. Sec. 2.2) and (x_p, y_p) are the corresponding pattern coordinates (according to the deflectometric registration), then $\vec{p}_{(p)} = (x_p, y_p, 0)^T$. It is precisely here, that the reconstruction algorithm implicitly assumes the x-y-plane of the pattern screen coordinate system to be the planar light source.

3. EXPERIMENT 1

The first experiment is designed to examine the errors in the reconstruction of a planar sample surface if the reconstruction algorithm uses the same camera intrinsic and extrinsic parameters as specified in the simulated setup (cf. Tab. 1) as a priori knowledge. Since the reconstruction algorithms and the simulation framework use the same camera model, the only difference between both in this experiment is their implicit assumption or explicit model, respectively, regarding the screen surface.

Four different height values Δh of the central screen bulge are examined in this experiment: 0 mm, 1 mm, 2 mm, and 3 mm. The simulated imaging process is performed for each value. The first value specifies a planar screen as a control experiment. In this case, the expected reconstruction result is a surface identical to the sample model used for the simulation.

For $\Delta h = 0$ mm, the reconstructed surface exhibits minimal deviations from the plane in which it was measured, in the order of magnitude of approx. 0.03 µm (cf. Fig. 4). These errors result from the digitization of the pixel measurements in the simulated cameras, when the measured intensities are saved as 8-bit images. For $\Delta h = 3$ mm, the reconstructed surfaces exhibit a tilt with a maximum height difference of 174 µm across the examined FOV. Notably, the two surfaces reconstructed from the two camera data, respectively, are tilted in opposing directions (cf. Fig. 4). Table 2 lists the maximum height difference of the reconstructed surfaces inside the FOV, their maximum deviation from the best-fit plane and the tilt angle of the best-fit plane for all reconstruction results.

4. EXPERIMENT 2

For real deflectometric inspection systems, perfect a priori knowledge of the true setup parameters – as assumed in the first experiment – is generally not available. Estimating these parameters is the purpose of calibration procedures. A second experiment therefore simulates a holistic calibration procedure[13] and examines the surface reconstructions using the camera internal and external parameters determined by the simulated calibration process.
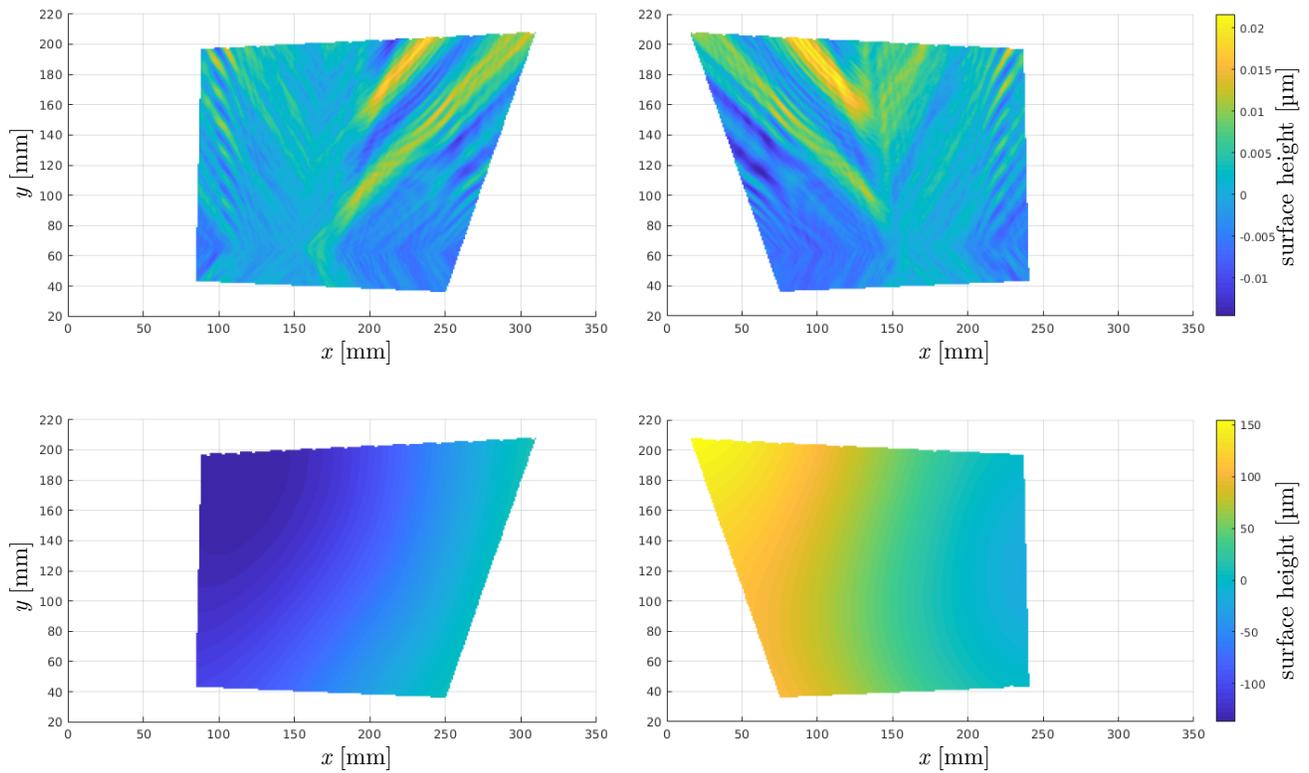
Figure 4. Reconstructed topography of a planar sample surface from deflectometric measurements in experiment 1. The first row shows the results from simulated measurements with a planar pattern screen. The results in the second row are from a simulation with a concave bulge of $3\,\mathrm{mm}$ height in the center of the pattern screen surface (note the difference of a factor of roughly 10^4 regarding the color scales in the top and bottom row). The left and right columns show the reconstructions from the left and right camera data, respectively.
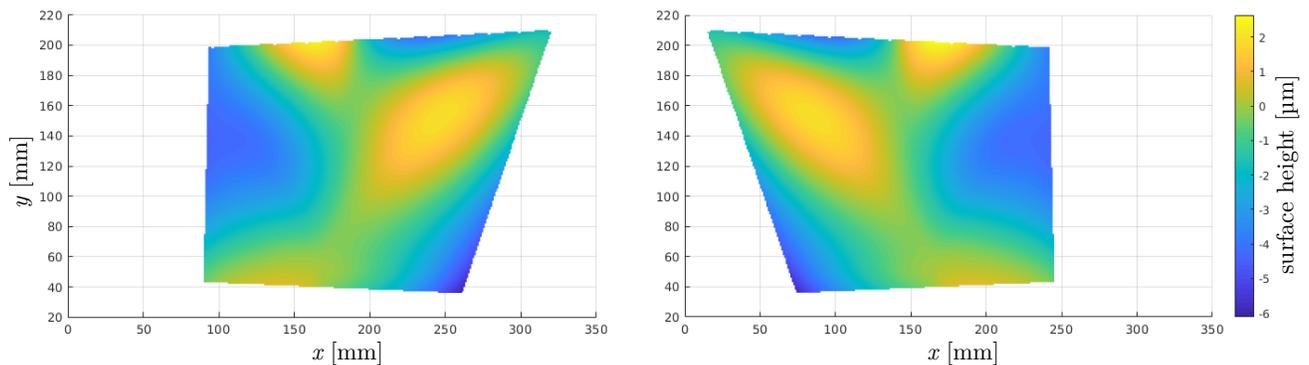
Figure 5. Reconstructed topography of a planar sample surface from deflectometric measurements in experiment 2. The results are reconstructions from the left and right camera data from a simulation with a concave bulge of $3\,\mathrm{mm}$ height in the center of the pattern screen surface. The results from simulated measurements with a planar pattern screen are identical to experiment 1 (cf. Fig. 4).

Table 3. Characteristics of surface reconstruction results in experiment 2

Screen bulge height Δh	0 mm		1 mm		2 mm		3 mm	
Camera (L = left, R = right)	L	R	L	R	L	R	L	R
Max. height difference [µm]	0.0	0.0	2.6	2.6	7.6	7.2	8.5	8.6
Max. deviation from best-fit-plane [µm]	0.0	0.0	1.6	1.6	4.3	4.3	5.3	5.2
Best-fit plane tilt angle [µrad]	0.0	0.1	6.3	6.5	20.1	16.7	10.5	11.0

The holistic calibration with a planar mirror requires deflectometric measurements with several different mirror poses.[13] These are obtained by simulating the imaging process for 16 different poses of a planar front-surface mirror. They follow the scheme we use for real calibrations: one measurement with the mirror lying on the base table; three with the mirror parallel to the table but elevated by different amounts; and a total of twelve measurements with the mirror rotated around the x- or y-axis by different angles (three positive and three negative rotation angles per axis).[20]

For $\Delta h = 0$ mm, the calibration procedure yields the correct internal and external camera parameters, i.e. they are identical with the parameters used for the simulated measurements. The reconstruction results are therefore also identical to those in experiment 1. For all other screen bulge heights Δh, the surfaces reconstructed in this experiment are not tilted, but possess irregular, wave-like errors with an amplitude of up to 9 µm in the examined FOV (cf. Fig. 5). Table 3 lists the maximum height difference of the reconstructed surfaces inside the FOV, their maximum deviation from the best-fit plane and the tilt angle of the best-fit plane for all reconstruction results.

5. CONCLUSIONS

In comparison with real physical experiments, simulation approaches in general often offer easier, more precise and potentially more wide-ranged control over the examined parameters. Simulated experiments also can be repeated arbitrarily. With regard to the current contribution, it would be very challenging to perform the simulated examinations in real-world experiments. Even accurately measuring the screen surface deformation is a difficult task, considering in particular that it has to be performed in the inclined pose used for the measurements. Manipulating the screen surface deformation in a controlled manner for experiments does not seem feasible with a reasonable amount of resources.

The incentive for the initial development of the simulation framework has been to explore and explain reconstruction results from real-world measurements of a mirror surface (cf. Fig. 6) that were not reproduced
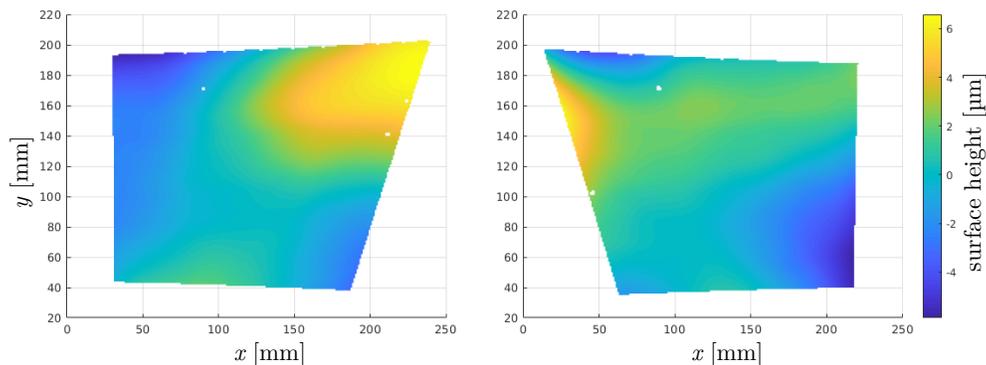
Figure 6. Reconstructed topography of a (nominally planar) front-surface mirror from real deflectometric measurements. The left and right plots show the reconstructions from two different cameras. The setup of the inspection system resembles the simulation model used in the simulation experiments.

with scans of the same mirror surface using an interferometer. Even if the simulation results (in experiment 2) do not exactly reproduce these real surface reconstructions, they manifest a similar tendency to create warped surface topographies, as a result from pattern screen deformation. The presented simulation results show that a deformation of the pattern screen with a central $\Delta h = 3\,\text{mm}$ limits the accuracy of surface reconstruction from PMD measurements to about $10\,\mu\text{m}$ with respect to the analyzed FOV, if a planar screen model is assumed in the reconstruction process.

If this limitation is not tolerable for a given application, there are in principle two potential solutions. One approach is to change the setup properties to reduce or prevent the pattern screen deformation. This may be achieved by using a different pattern screen with a stiffer, more stable surface. Alternatively, the entire inspection setup may be altered towards a vertical arrangement of the screen and, in consequence, a non-horizontal arrangement of the sample surface. The second category of solutions for the presented accuracy limitation is to enhance to measurement processes on the algorithmic side by including the screen surface topography into the reconstruction algorithms as part of the measurement setup model.[24] Of course, this information then also needs to be determined in the calibration process. This is the focus of current research efforts. Some major questions in this respect are how much flexibility (i.e. how many free variables) the screen model requires for good results, and how to determine them in a robust calibration process.

It may further be of interest to examine the effects of a non-planar calibration mirror on the holistic calibration process, given that the current implementations assume the mirror surface to be a perfect plane.[20] Finally, future developments could extend the simulation framework with features such as alternative camera models or depth-of-field effects.

ACKNOWLEDGMENTS

REFERENCES

[1] Nayak, L., Mohanty, S., Nayak, S. K., and Ramadoss, A., "A review on inkjet printing of nanoparticle inks for flexible electronics," *J. Mater. Chem. C* **7**, 8771–8795 (2019).

[2] Kwon, K.-S., Rahman, M. K., Phung, T. H., Hoath, S. D., Jeong, S., and Kim, J. S., "Review of digital printing technologies for electronic materials," *Flex. Print. Electron.* **5**, 043003 (2020).

[3] Cummins, G. and Desmulliez, M. P., "Inkjet printing of conductive materials: a review," *Circuit World* **38**, 193–213 (2012).

[4] Li, W., Bothe, T., Osten, W., and Kalms, M., "Object adapted pattern projection—Part I: generation of inverse patterns," *Opt. Lasers Eng.* **41**, 31–50 (2004).

[5] Beyerer, J., Puente León, F., and Frese, C., [*Machine Vision: Automated Visual Inspection: Theory, Practice and Applications*], Springer, Berlin (2016).

[6] Knauer, M. C., Kaminski, J., and Häusler, G., "Phase Measuring Deflectometry: a new approach to measure specular free-form surfaces," in [*Optical Metrology in Production Engineering*], Osten, W. and Takeda, M., eds., *Proc. SPIE* **5457**, 366–376, SPIE (2004).

[7] Huang, L., Idir, M., Zuo, C., and Asundi, A., "Review of phase measuring deflectometry," *Opt. Lasers Eng.* **107**, 247–257 (2018).

[8] Allgeier, S., Gengenbach, U., Köhler, B., Reichert, K.-M., and Hagenmeyer, V., "Robust phase unwrapping based on non-coprime fringe pattern periods for deflectometry measurements," in [*Automated Visual Inspection and Machine Vision III*], Beyerer, J. and Puente León, F., eds., *Proc. SPIE* **11061**, 1106103, SPIE (2019).

[9] Zhang, Z., Chang, C., Liu, X., Li, Z., Shi, Y., Gao, N., and Meng, Z., "Phase measuring deflectometry for obtaining 3D shape of specular surface: a review of the state-of-the-art," *Opt. Eng.* **60**, 020903 (2021).

[10] Pak, A., "Stability of absolute depth reconstruction from deflectometric measurement data," in [*Interferometry XVII: Techniques and Analysis*], Creath, K., Burke, J., and Schmit, J., eds., *Proc. SPIE* **9203**, 92030E, SPIE (2014).

[11] Huang, L., Idir, M., Zuo, C., Kaznatcheev, K., Zhou, L., and Asundi, A., "Comparison of two-dimensional integration methods for shape reconstruction from gradient data," *Opt. Lasers Eng.* **64**, 1–11 (2015).

[12] Tarini, M., Lensch, H. P., Goesele, M., and Seidel, H.-P., "3D acquisition of mirroring objects using striped patterns," *Graph. Models* **67**, 233–259 (2005).

[13] Faber, C., [*New Methods and Advances in Deflectometry*], Erlangen Scientific Press, Erlangen (2012).

[14] Ren, H., Gao, F., and Jiang, X., "Iterative optimization calibration method for stereo deflectometry," *Opt. Express* **23**, 22060–22068 (2015).

[15] Zhou, T., Chen, K., Wei, H., and Li, Y., "Improved system calibration for specular surface measurement by using reflections from a plane mirror," *Appl. Opt.* **55**, 7018–7028 (2016).

[16] Xu, Y., Gao, F., Zhang, Z., and Jiang, X., "A holistic calibration method with iterative distortion compensation for stereo deflectometry," *Opt. Lasers Eng.* **106**, 111–118 (2018).

[17] Liu, M., Hartley, R., and Salzmann, M., "Mirror surface reconstruction from a single image," *IEEE Trans. Pattern Anal. Mach. Intell.* **37**, 760–773 (2015).

[18] Liu, Y., Huang, S., Zhang, Z., Gao, N., Gao, F., and Jiang, X., "Full-field 3D shape measurement of discontinuous specular objects by direct phase measuring deflectometry," *Sci. Rep.* **7**, 10293 (2017).

[19] Hartley, R. and Zisserman, A., [*Multiple View Geometry in Computer Vision*], Cambridge University Press, Cambridge, second ed. (2006).

[20] Allgeier, S., Gengenbach, U., Köhler, B., Reichert, K.-M., and Hagenmeyer, V., "Reproducibility of two calibration procedures for phase-measuring deflectometry," in [*Interferometry XX*], North Morris, M. B., Creath, K., and Porras-Aguilar, R., eds., *Proc. SPIE* **11490**, 114900G, SPIE (2020).

[21] Drap, P. and Lefèvre, J., "An exact formula for calculating inverse radial lens distortions," *Sensors* **16** (2016).

[22] Stoyan, G. and Baran, A., [*Elementary Numerical Mathematics for Programmers and Engineers*], Birkhäuser, Cham (2016).

[23] Ascher, U. M. and Greif, C., [*A First Course in Numerical Methods*], Society for Industrial and Applied Mathematics, Philadelphia, PA (2011).

[24] Bartsch, J., Kalms, M., and Bergmann, R. B., "Improving the calibration of phase measuring deflectometry by a polynomial representation of the display shape," *J. Eur. Opt. Soc. Rapid Publ.* **15** (2019).