Segmentation of Photovoltaic Module Cells in Electroluminescence Images

Sergiu Deitsch[a,b,d,*], Claudia Buerhop-Lutz[c], Andreas Maier[d], Florian Gallwitz[a,b], Christian Riess[d]

[a]*Energie Campus Nuremberg, Fürther Str. 250, 90429 Nuremberg, Germany*
[b]*Nuremberg Institute of Technology, Department of Computer Science, Hohfederstr. 40, 90489 Nuremberg, Germany*
[c]*ZAE Bayern, Immerwahrstr. 2, 91058 Erlangen, Germany*
[d]*Pattern Recognition Lab, University of Erlangen-Nuremberg, Martensstr. 3, 91058 Erlangen, Germany*

Abstract

High resolution electroluminescence (EL) images captured in the infrared spectrum allow to visually and non-destructively inspect the quality of photovoltaic (PV) modules. Currently, however, such a visual inspection requires trained experts to discern different kind of defects, which is time-consuming and expensive.

In this work, we make an important step towards improving the current state-of-the-art in solar module inspection. We propose a robust automated segmentation method to extract individual solar cells from EL images of PV modules. Automated segmentation of cells is a key step in automating the visual inspection workflow. It also enables controlled studies on large amounts of data to understanding the effects of module degradation over time—a process not yet fully understood.

The proposed method infers in several steps a high level solar module representation from low-level edge features. An important step in the algorithm is to formulate the segmentation problem in terms of lens calibration by exploiting the plumbline constraint. We evaluate our method on a dataset of various solar modules types containing a total of 408 solar cells with various defects. Our method robustly solves this task with a median weighted Jaccard index of 95.09 % and an F_1 score of 97.23 %.

Keywords: PV modules, EL imaging, visual inspection, lens distortion, solar cell extraction

1. Introduction

Visual inspection of solar modules using EL imaging allows to easily identify damage inflicted to solar panels either by environmental influences such as hail, during the assembly process, or due to prior material defects or material aging. The resulting defects can notably decrease the photoelectric conversion efficiency of the modules and thus their energy yield. This can be avoided by regularly inspecting the solar modules and depending on the type of damage, either by repairing or replacing the defective units.

An important step towards automated visual inspection of photovoltaic modules in EL images is the segmentation of individual cells from the module. However, the appearance of PV modules in EL images depends on a number of different factors. It varies with the type of semiconducting material used for the solar cells and with the shape of individual solar cell wafers. Additionally, cracks and defects introduce distracted streaks or, if a cell is disconnected from the electrical circuit, make them much darker than functional cells. This makes an automated segmentation for solar cells challenging. Additionally, solar modules vary in the number and layout of cells, and cells themselves are oftentimes subdivided by busbars into segments of different sizes. For a fully automated segmentation, it is desirable to infer also the layout and segmentation subdivision from the images.

Given these considerations, important cues for segmenting individual solar cells are provided by the inter-cell borders and the busbars. Corners of solar cells are less reliable due to varying shapes of the silicon wafer or due to the mechanical damage of the solar panel.

Inter-cell borders and busbars are straight line features that can be easily detected, for instance, using the linear Hough transform [1]. However, noise of the sensor for capturing the EL image due to long

---

*Corresponding author
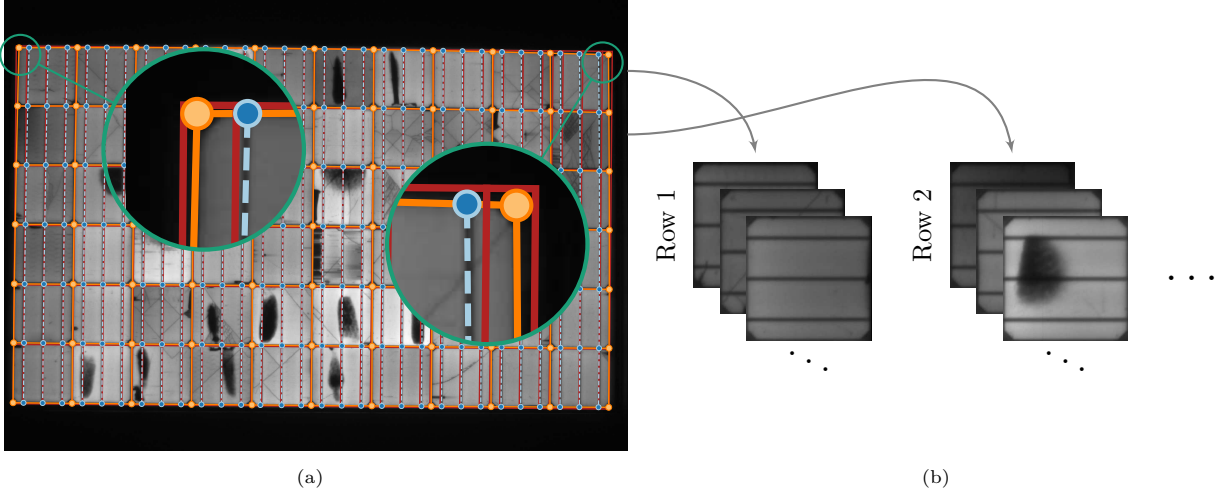   *Email address:* `sergiu.deitsch@fau.de` (Sergiu Deitsch)

Figure 1: (a) An EL image of a PV module overlaid by a rectangular grid (—) and parabolic curve grid (—) including the busbars (- - -) determined using our approach. Notice how both grids do not align properly due to (weak) lens distortion, which increases especially towards the image border. Without the curve grid, the alignment error will typically be even larger due to unknown true grid intersections which are used here for registering both grids. Using the curve grid, we estimate the lens distortion, rectify the image and finally extract the individual cells (b). The segmented solar cells can be used for further analysis, such as automatic defect classification or failure prediction in PV modules.

exposure in the dark chamber or blur due to incorrect focus lead to failure of such simple approaches. Moreover, lens distortion causes straight lines to be captured as curves. This problem could be solved by calibrating the EL camera in advance, but calibration in a dark room is not straightforward and requires experienced personnel. Also, calibration is not practicable in situations where EL images were postprocessed, e.g., by scaling, rotating or cropping the images, which is common in visual inspection of PV modules. Thus, we propose a robust, completely automatic segmentation of solar modules into solar cells from high resolution EL images of PV modules. We assume that the images are captured in a manufacturing setting. In this case, the EL irradiation of the solar module predominates the background irradiation, and the solar modules are captured facing the EL camera without major perspective distortion.

We particularly hope that the proposed segmentation algorithm facilitates new research allowing to understand the influence of module degradation on the efficiency of solar modules and correspondingly the power generation. A robust segmentation of solar cells allows to continuously and automatically monitor the degradation process, for instance, by observing the differences in a series of solar cell images captured over a certain period of time. The segmentation also allows to automatically create

training data for learning-based algorithms for defect classification and failure prediction.

1.1. Contributions

To best of our knowledge, the proposed segmentation pipeline is the first work to enable a fully automatic extraction of solar cells from EL images of solar modules. The segmentation works both on monocrystalline and polycrystalline type PV modules and is robust with respect to various defects typically occurring in solar modules such as (deep) cracks and disconnected cells. Within the segmentation pipeline, one particular contribution is a robust initialization with only few parameters for estimating lens distortion.

1.2. Outline

The remainder of this work is organized as follows. Section 2 discusses the related work. In Section 3, the individual stages of the segmentation pipeline are presented. In Section 4, we evaluate the presented segmentation approach on a number of different PV modules with respect to the segmentation accuracy. Finally, the conclusions are given in Section 5.

2. Related Work

The segmentation of PV modules into individual solar cells is related to the detection of calibration

patterns, such as checkerboard patterns typically used for calibrating intrinsic camera and lens parameters [2, 3, 4, 5, 6]. However, the appearance of calibration patterns is typically perfectly known, whereas detection of solar cells is encumbered by various (a-priori unknown) defects. Additionally, the number and layout of solar cells in PV modules is also assumed to be unknown. For the calculation of lens parameters, the lens distortion of EL images of PV modules may be too weak to apply existing approaches that rely on strong image deformations to unambiguously deduce the lens parameters.

To estimate the parameters of a lens distortion model, the plumbline constraint [7] is typically employed. The constraint exploits the fact that the projection of straight lines under radial and tangential distortion will not be truly straight. For example, under radial distortion straight lines are captured as curves. For typical visual inspection tasks, a single image is sufficient to estimate the lens distortion parameters [8, 9, 10, 11, 12]. This can be especially achieved by decoupling the intrinsic parameters of the camera from the parameters describing the lens distortion model.

Since few years, Convolutional Neural Network (CNN) are used for various segmentation tasks. Existing CNN-based segmentation tasks can be categorized as (1) object detection, (2) semantic segmentation, and (3) instance-aware segmentation. One of the first CNN object detection architectures is Regions with CNN features (R-CNN) Girshick et al. [13] to learn features that are subsequently classified using a class-specific linear Support Vector Machine (SVM) to generate region proposals. R-CNN learns to simultaneously classify object proposals and refine their spatial locations. The predicted regions, however, provide only a coarse estimation of object's location by means of bounding boxes. Girshick [14] proposed Fast Region-based Convolutional Neural Network (Fast R-CNN) by accelerating training and testing times while also increasing the detection accuracy. Ren et al. [15] introduced Region Proposal Network (RPN) that shares full-image convolutional features with the detection network enabling nearly cost-free region proposals. RPN is combined with Fast R-CNN into a single network that simultaneously predicts object bounds and estimates the probability of an object for each proposal. For semantic segmentation, Long et al. [16] introduced Fully Convolutional Networks (FCNs) allowing for pixelwise inference. The FCN is learned end-to-end and pixels-to-pixels requiring appropriately labeled training data. For instance segmentation, Li et al. [17] combined segment proposal and object detection for Fully Convolutional Instance Segmentation (FCIS) where the general idea is to predict the locations in a fully convolutional network. He et al. [18] proposed a Mask R-CNN which extends Faster R-CNN.

The main limitation for such learning-based approaches is that they require a considerable number of appropriately labeled images for training. However, pixelwise labeling is time-consuming and in absence of data not possible at all. Also, such learning-based approaches require training data that is statistically representative for the test data, which oftentimes requires to re-train a model on data with different properties. In contrast, the proposed approach can be deployed and robustly segments cell modules without notable requirements of (potentially manually labeled) training data.

3. Methodology

The general framework for segmenting the solar cells in EL images of PV modules is as follows. First, we locate the busbars and the inter solar cell borders by extracting the ridge edges. The ridge edges are extracted at subpixel accuracy and approximated by a set of smooth curves defined as second-degree polynomials. The parametric representation is used to construct an initial grid of perpendicularly arranged curves that identify the PV module. Using this curve grid, we estimate the initial lens distortion parameters and hypothesize the optimal set of curves by further excluding outliers in a RAndom Sample Consensus (RANSAC) scheme. Then we refine the lens distortion parameters that we eventually use to rectify the EL image. From the final set of curves we infer the PV module configuration and finally extract the size, perspective, and orientation normalized images of solar cells. Fig. 2 illustrates the individual pipeline stages.

3.1. Preprocessing

First, the contrast of an EL image is stretched to the full range of intensity values. Then, low-level edge processing is applied to attenuate structural variations that might stem from cracks or silicon wafer texture, with the goal of preserving larger lines or curves.
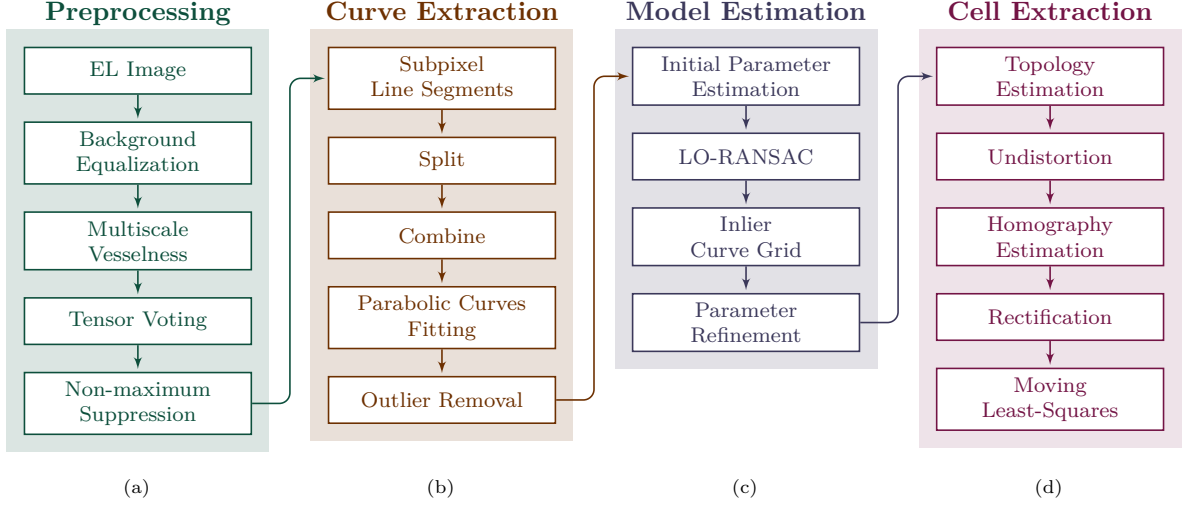
Preprocessing	Curve Extraction	Model Estimation	Cell Extraction
EL Image	Subpixel Line Segments	Initial Parameter Estimation	Topology Estimation
Background Equalization	Split	LO-RANSAC	Undistortion
Multiscale Vesselness	Combine	Inlier Curve Grid	Homography Estimation
Tensor Voting	Parabolic Curves Fitting	Parameter Refinement	Rectification
Non-maximum Suppression	Outlier Removal		Moving Least-Squares
(a)	(b)	(c)	(d)

Figure 2: The proposed PV module segmentation pipeline consists of four stages. In the preprocessing stage (a), all contours are extracted. In the curve extraction stage (b), combinations of line segment are determined that fit to parabolic curves. In the model estimation stage (c), the lens distortion and inlier parabolic curves are determined. In the cell extraction stage (d) the cell topology is determined and the cells are extracted.

3.1.1. Contrast Stretching

We follow the approach by Franken et al. [19]. A copy I_{bg} of the input EL image I is created. This copy is slightly blurred with a Gaussian kernel, and a morphological closing with a disk-shaped structure element is applied. Dividing each pixel of I by I_{bg} attenuates unwanted background noise while emphasizing high contrast regions. Then, histogram equalization is applied to increase its overall contrast. Fig. 4(b) shows the resulting image I.

3.1.2. Gaussian Scale-Space Vesselness

Applying the Hessian of the second derivative at multiple scales allows to robustly extract line and curve ridges. This can be performed, e.g., by the vesselness filter [20, 19].

Vesselness response in a Gaussian pyramid is computed as follows. Let $\boldsymbol{u} := (u, v)^\top$ denote discrete pixel coordinates, I, and O the number of octaves in the pyramid, and $P \in \mathbb{N}$ the number of sublevels in each octave. We set the standard deviation of the Gaussian at the finest resolution σ to the golden ratio $\sigma = 1 + \sqrt{5}/2 \approx 1.6$. For each octave $o \in \{0, \ldots, O - 1\}$ and sublevel $\ell \in \{0, \ldots, P - 1\}$, we convolve I with a Gaussian kernel of size $s := \gamma^{o + \ell/O}\sigma$ to obtain the image L, where $\gamma = 2$ is the downsampling factor from one octave to another. We then approximate the Hessian $\boldsymbol{H}$ by applying a 7×7 Sobel operator to the image

gradient ∇L of the downsampled image I:

$$\boldsymbol{H} := \begin{bmatrix} \frac{\partial^2 L}{\partial u^2} & \frac{\partial^2 L}{\partial u \partial v} \\ \frac{\partial^2 L}{\partial v \partial u} & \frac{\partial^2 L}{\partial v^2} \end{bmatrix} = \frac{\partial \nabla L}{\partial \boldsymbol{u}} \tag{1}$$

The eigenvalue decomposition of the symmetric Hessian provides information about line-like and blob-like structures. Let $\boldsymbol{H} = \boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^{-1}$ denote this eigenvalue decomposition, where $\boldsymbol{\Lambda} := \operatorname{diag}(\lambda_1, \lambda_2) \in \mathbb{R}^{2\times 2}$ is a diagonal matrix containing the eigenvalues $\lambda_1 > \lambda_2$. Local ridgeness $R(\boldsymbol{u})$ is obtained as

$$R(\boldsymbol{u}) = \max_{\substack{o \in \{0, \ldots, O-1\} \\ \ell \in \{0, \ldots, P-1\}}} \{\lambda_1(\boldsymbol{u}), 0\} . \tag{2}$$

The ridgeness specifies the likelihood of a line segment in the image at position $\boldsymbol{u}$. The angle $\beta(\boldsymbol{u})$ of the most likely line segment orientation at position $\boldsymbol{u}$ is obtained from orientation of the associated eigenvector.

The ridgeness $R(\boldsymbol{u})$ together with $\beta(\boldsymbol{u})$ provide initial cues for ridge edges in the EL image (see Fig. 4(c)).

3.1.3. Contextual Enhancement via Tensor Voting

Ridgeness can be very noisy. To discern noise and high curvatures from actual line and curve features, $R(\boldsymbol{u})$ is contextually enhanced using tensor voting [21].

Tensor voting uses a stick tensor voting field to model the likelihood that a feature in the neighborhood belongs to the same curve as the feature in the origin of the voting field [22]. The voting field is defined with a likelihood function

$$w(\boldsymbol{x}) = \exp\left(-\frac{r^2}{2\varsigma^2}\cos^{2\nu}\phi\right) \qquad (3)$$

that indicates whether $\boldsymbol{x} = (r\cos\phi, r\sin\phi)$ passes through a curve, and

$$\gamma(\boldsymbol{x}) = 2\phi \qquad (4)$$

specifies the most likely angle of that curve. $\varsigma > 0$ controls the proximity of the voting field, and ν determines the angular specificity that we set to $\nu = 2$ in our experiments. w and γ are combined to generate a tensorial filter kernel $\boldsymbol{V}: \mathbb{R}^2 \to \mathbb{R}^{2\times2}$ that assigns a square matrix to all spatial positions as

$$\boldsymbol{V}(\boldsymbol{x}) = w(\boldsymbol{x})\boldsymbol{c}(\boldsymbol{x})\boldsymbol{c}(\boldsymbol{x})^\top \qquad (5)$$

for $\boldsymbol{c}(\boldsymbol{x}) = (\cos\gamma(\boldsymbol{x}), \sin\gamma(\boldsymbol{x}))^\top$. The input is ridgeness R and orientation β, and the output is a tensor field

$$\boldsymbol{U}(\boldsymbol{x}) = \int_\Omega R(\boldsymbol{x}')\boldsymbol{V}^\beta(\boldsymbol{x}')(\boldsymbol{x} - \boldsymbol{x}')\,\mathrm{d}\boldsymbol{x}' \qquad (6)$$

where $\boldsymbol{V}^\beta(\boldsymbol{x})$ denotes the tensorial voting field rotated over β by a 2-D rotation matrix $\boldsymbol{R}_\beta \in \mathrm{SO}(2)$

$$\boldsymbol{V}^\beta(\boldsymbol{x}) = \boldsymbol{R}_\beta\boldsymbol{V}(\boldsymbol{R}_\beta^\top\boldsymbol{x})\boldsymbol{R}_\beta^\top \ . \qquad (7)$$

Following Franken et al. [22], stickness $\tilde{R}(\boldsymbol{u})$ is computed as the difference between the two eigenvalues of $\boldsymbol{U}$, and $\tilde{\beta}(\boldsymbol{u})$ is the angle of the largest eigenvalue, analogously to $\beta(\boldsymbol{u})$. Eq. (6) can be efficiently computed as a sum of eight convolutions in the Fourier domain using steerable tensor voting. We iterate tensor voting two times, since one pass is not always sufficient. Unlike [19], however, we do not thin out the stickness immediately after the first pass to avoid too many disconnected edges. Given the high resolution of the EL images in the our dataset of approximately 2500×2000 pixels, we use a fairly large proximity of $\varsigma_1 = 15$ in the first tensor voting step, and $\varsigma_2 = 10$ in the second. Fig. 4(d) shows a typical stickness output.

3.2. Curve Extraction

Centerline points of ridges are grouped by their curvature. Parabolic curves are fitted to these points, which yields a higher level representation, while simultaneously discarding point outliers.

3.2.1. Extraction of Ridges at Subpixel Accuracy

We perform non-maximum suppression by global Otsu thresholding [23] on the stickness $\tilde{R}(\boldsymbol{u})$ followed by skeletonization by Saeed et al. [24]. Afterwards, we collect the points that represent the centerline of the ridges through edge linking [25]. These coarse positions can be refined by setting the centerline to the mean of a Gaussian that is fitted to the edge profile [26] using the Gauss-Newton algorithm [27]. The window of the Gaussian is empirically set to 21 pixels, with four sample points per pixel that are computed via bilinear interpolation. The Gauss-Newton algorithm is initialized with the sample mean and standard deviation in the window, and multiplicatively scaled to the intensity at the mean. The mean of the fitted Gaussian is then reprojected along the edge profile to obtain the edge subpixel position.

3.2.2. Connecting Larger Curve Segments

A limitation of the edge linking method is that it does not prioritize curve pairs with similar orientation. To address this, we first reduce the set of points that constitute a curve to a sparser representation of line segments using the Douglas-Peucker algorithm [28], where we empirically set the error threshold to $\epsilon = 1/5$. That way, two linked edges are disconnected if the angle between them exceeds 5°. In a second pass through the line segments, two line segments are joined if they are nearby, of approximately the same length and pointing into the same direction (within a range of $\pm10°$).

3.2.3. Parametric Curve Representation

Projected lines are represented as second-degree polynomials to model radial distortion. The curve parameters are computed via linear regression on the curve points. More specifically, let

$$f(x) = a_2x^2 + a_1x + a_0 \qquad (8)$$

denote a second-degree polynomials in horizontal or vertical direction. The curve is fitted to line segment points $\mathcal{C}_k := \{(x_i, y_i)^\top \mid i = 1, \ldots, n\}$ by minimizing the Mean Squared Error (MSE)

$$\mathrm{MSE}(f) = \frac{1}{n}\sum_{i=1}^n \|f(x_i) - y_i\|_2^2 \qquad (9)$$

in a RANSAC [29] iteration. In one iteration, we randomly sample three points to fit Eq. (8), and then determine which of the remaining points support
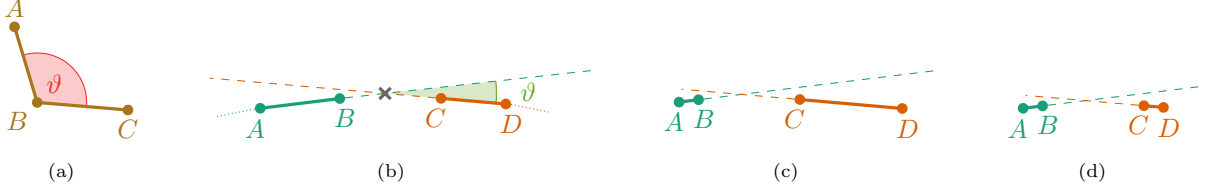
Figure 3: (a) The two adjacent line segments AB and BC are split at point B because $\vartheta = \angle ABC$ exceeds the threshold ϑ_s. (b) The disjoint line segments AB and CD are joined because they intersect and their tangent ϑ is below the threshold. While the extensions of line segments in (c) and (d) intersect and their tangent is below the threshold, the line segments are too short with respect to the opposite line segment ($\overrightarrow{AB}^2 < \overrightarrow{CD}$) or their distance $\overrightarrow{BC}$ is larger than the line segments, i.e., $\overrightarrow{AB}^2 < \overrightarrow{BC} \wedge \overrightarrow{CD}^2 < \overrightarrow{BC}$.

this curve model. The curve quality is evaluated using MSE (9). To keep the computational time low, RANSAC is limited to 100 iterations, and early aborted if many inliers are found [30].

3.3. Curve Grid Model Estimation

The individual curves are used to jointly form a grid, which allows to further discard outliers, and to estimate lens distortions. To estimate the distortions, we use the plumbline constraint, which uses the prior that straight lines are mapped onto curves. That way, it becomes possible to estimate distortion efficiently from a single image, which allows to use this approach also post-hoc on cropped, zoomed or similarly processed images.

3.3.1. Representation of Lens Distortion

We represent the radial lens distortion by a function $L \colon \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ that maps the distance of a pixel from the distortion center to a distortion factor. This factor can be used to radially displaces each normalized image coordinate $\tilde{\boldsymbol{x}}$.

Image coordinates are normalized by scaling down image coordinates $\boldsymbol{x} := (x, y)^\top$ horizontally by the distortion aspect ratio s_x followed by shifting the center of distortion $\boldsymbol{c} := (c_x, c_y)^\top$ to the origin and normalizing the resulting 2-D point to the unit range using the dimensions $N \times M$ of the image. Homogeneous coordinates allow to express the normalization with a matrix product. By defining the upper-triangular matrix

$$\boldsymbol{K} = \begin{bmatrix} s_x M & 0 & c_x \\ 0 & N & c_y \\ 0 & 0 & 1 \end{bmatrix} \qquad (10)$$

the normalization is performed using $\boldsymbol{n} \colon \Omega \to \mathcal{R}$

$$\boldsymbol{n}(\boldsymbol{x}) = \boldsymbol{\pi}\left(\boldsymbol{K}^{-1}\boldsymbol{\pi}^{-1}(\boldsymbol{x})\right) \qquad (11)$$

where $\boldsymbol{\pi} \colon \mathbb{R}^3 \to \mathbb{R}^2$ projects homogeneous to inhomogeneous coordinates,

$$\boldsymbol{\pi} \colon (x, y, z)^\top \mapsto \frac{1}{z}(x, y)^\top, \quad \text{for } z \neq 0 . \qquad (12)$$

and the inverse operation $\boldsymbol{\pi}^{-1} \colon \mathbb{R}^2 \to \mathbb{R}^3$ backprojects inhomogeneous to homogeneous coordinates:

$$\boldsymbol{\pi}^{-1} \colon (x, y)^\top \mapsto (x, y, 1)^\top \qquad (13)$$

Note that the inverse mapping $\boldsymbol{n}^{-1}$ converts normalized image coordinates to image plane coordinates.

3.3.2. The Field-of-View Lens Distortion Model

To describe the radial lens distortion, we use the first order Field-of-View (FOV) lens model by Devernay and Faugeras [8] that has a single distortion parameter ω. While images can also suffer from tangential distortion, this type of distortion is often negligible [31]. The parameter $0 < \omega \leq \pi$ denotes the opening angle of the lens. The corresponding radial displacement function L is defined in terms of the distortion radius $r \geq 0$ as

$$L(r) = \frac{1}{\omega} \arctan\left(2r \tan \frac{\omega}{2}\right) \qquad (14)$$

One advantage of the model is that its inversion has a closed-form solution with respect to the distortion radius r.

Similar to Devernay and Faugeras [8], we decouple the distortion from the projection onto the image plane, avoiding the need to calibrate for intrinsic camera parameters. Instead, the distortion parameter ω is coupled with the distortion center $\boldsymbol{c}$ and distortion aspect ratio s_x which are collected in a vector $\boldsymbol{\theta} := (\boldsymbol{c}, s_x, \omega)$.
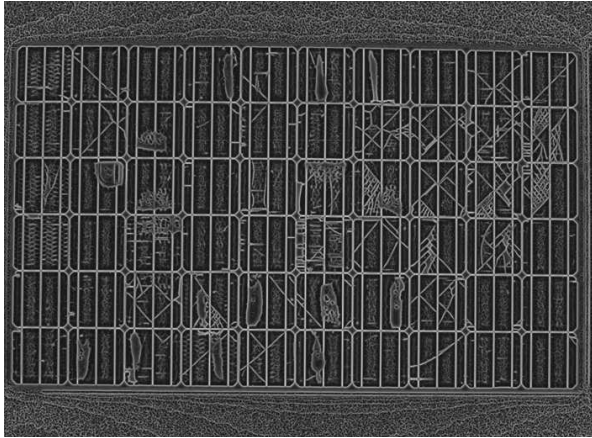
Normalized undistorted image coordinates $\tilde{\boldsymbol{x}}_u = \boldsymbol{\delta}^{-1}(\tilde{\boldsymbol{x}}_d)$ can be directly computed from distorted
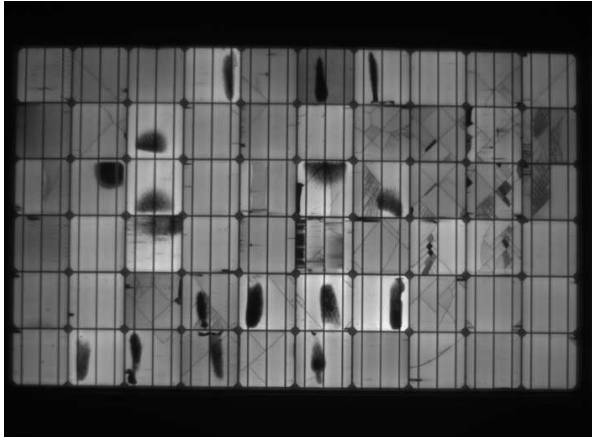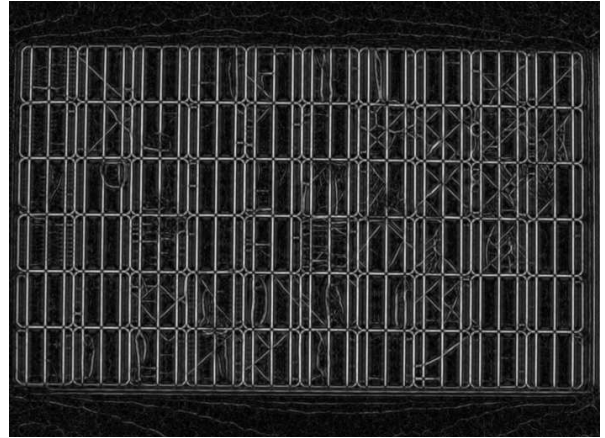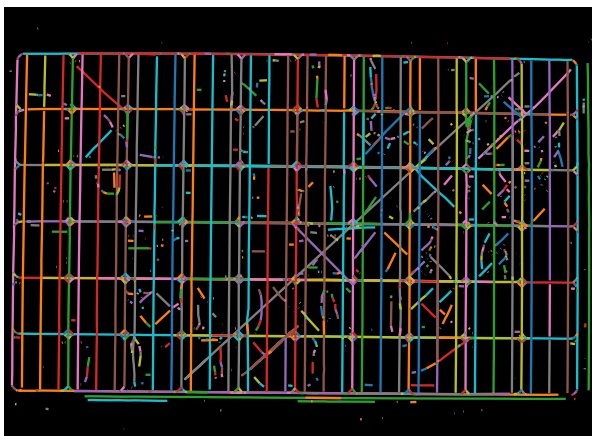
(a) EL image of a monocrystalline PV module
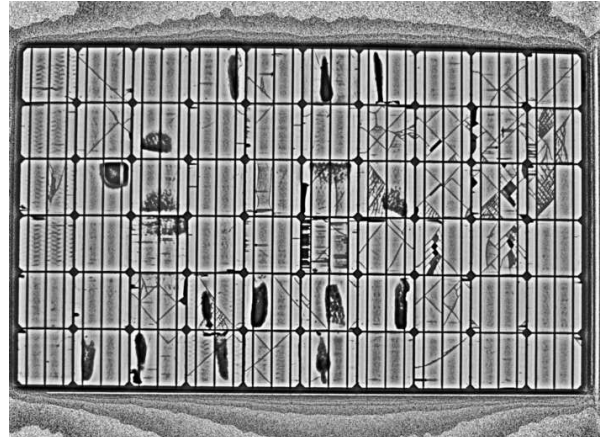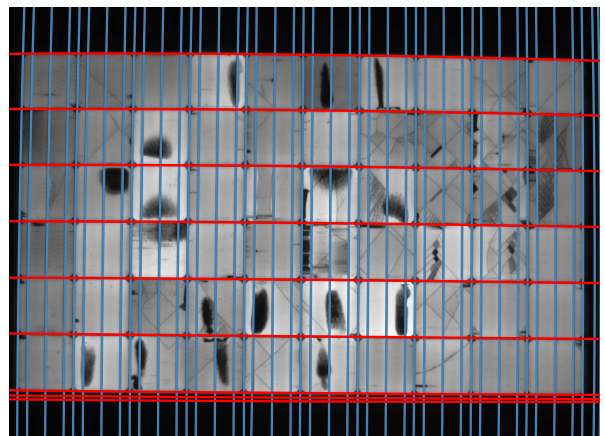

(b) Background equalized image

(c) Rigidness image $R(\boldsymbol{u})$ from the multiscale vesselness filter

(d) Stickness of the contextually enhanced ridgeness using tensor voting

(e) Extracted line segments grouped by their curvature

(f) Horizontal (red) and vertical (blue) parabolic curves filtered using the intersection constraint

Figure 4: Visualization of the preprocessing, curve extraction, and model estimation stages

coordinates $\tilde{\boldsymbol{x}}_d$ as

$$\boldsymbol{\delta}^{-1}(\tilde{\boldsymbol{x}}_d) = \begin{cases} \dfrac{L^{-1}(r_d)}{r_d} \tilde{\boldsymbol{x}}_d & \text{if } r_d \neq 0 \\ 0 & \text{otherwise} \end{cases} \quad (15)$$

where $r_d = \|\tilde{\boldsymbol{x}}_d\|_2$ is the distance of $\tilde{\boldsymbol{x}}_d$. $L^{-1}(r)$ is the inverse of the lens distortion function in Eq. (14), namely

$$L^{-1}(r) = \frac{\tan r\omega}{2\tan\frac{\omega}{2}} \ . \quad (16)$$

The function that undistorts a point $\boldsymbol{x} \in \Omega$ is thus

$$\boldsymbol{u}(\boldsymbol{x}) = \boldsymbol{n}^{-1}\left(\boldsymbol{\delta}^{-1}\left(\boldsymbol{n}(\boldsymbol{x})\right)\right) \ . \quad (17)$$

3.3.3. Estimation of Initial Lens Distortion Model Parameters

Lens distortion is specified by the distortion coefficient ω, the distortion aspect ratio s_x, and the distortion center $\boldsymbol{c}$. Naive solution leads to a non-convex objective function with several local minima. Therefore, we first seek an initial set of parameters close to the optimum, and then proceed using a convex optimization to refine the parameters.

We initialize the distortion aspect ratio s_x to 1, and the distortion center to the intersection of two perpendicular curves with smallest coefficient in the highest order polynomial. Such curves can be assumed to have the smallest curvature and are thus located near the distortion center.

Distortion Coefficient. Estimation of the distortion coefficient ω from a set of distorted image points is not straightforward because the distortion function $L(r)$ is non-linear. One way to overcome this problem is to linearize $L(r)$ with Taylor polynomials, and to estimate ω with linear least-squares.

To this end, we define the distortion factor

$$k := \frac{L(r)}{r}, \quad \text{for } k \in \mathbb{R}_{>0} \quad (18)$$

which maps undistorted image points $\{\boldsymbol{p}_i\}_{i=1,\dots,n}$ lying on the straight lines to distorted image points $\{\boldsymbol{q}_i\}_{i=1,\dots,n}$ lying on the parabolic curves. Both point sets are related by

$$\boldsymbol{p}k = \boldsymbol{q} \ . \quad (19)$$

The distorted points $\boldsymbol{q}_i$ are straightforward to extract by evaluating the second-degree polynomial of the parabolic curves. To determine $\boldsymbol{p}_i$, we define a line with the first and the last point in $\boldsymbol{q}_i$, and select
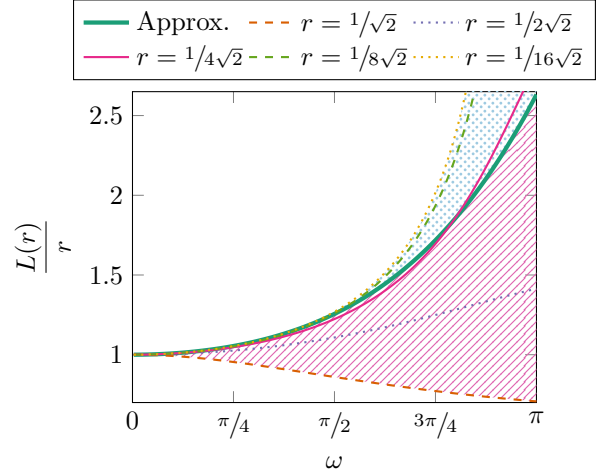

Figure 5: Approximation of the distortion coefficient ω using Eq. (24) compared to the exact solution with respect to varying radii r. For large radii outside the range of normalized coordinates (i.e., the radius of the half-unit circle $r > 1/\sqrt{2}$), the estimate is not accurate. This implies that the ideal sampled points must be both at some distance from the image border and also from the distortion center. As a sidenote, the estimation error becomes unacceptable for wide lenses where $\omega > \pi/4$. However, the EL images in this work are well below this threshold.

points from this line. Collecting these points in the vectors $\boldsymbol{p} \in \mathbb{R}^{2n}$ and $\boldsymbol{q} \in \mathbb{R}^{2n}$ yields an overdetermined system of $2n$ linear equations in one unknown. $\hat{k}$ is then estimated via linear least-squares as

$$\hat{k} = \underset{k}{\operatorname{argmin}} \|\boldsymbol{q} - \boldsymbol{p}k\|_2^2 \ , \quad (20)$$

where the solution is found via the normal equations [32] as

$$\hat{k} := \frac{\boldsymbol{p}^\top \boldsymbol{q}}{\boldsymbol{p}^\top \boldsymbol{p}} \ . \quad (21)$$

To determine ω from the relation $k = L(r)/r$, $L(r)$ is expanded around $\omega_0 = 0$ using Taylor series.

More specifically, we use a second order Taylor expansion to approximate

$$\arctan(x) = x + \mathcal{O}(x^2) \ , \quad (22)$$

and a sixth order Taylor expansion to approximate

$$\tan(y) = y + \frac{y^3}{3} + \frac{2y^5}{15} + \mathcal{O}(y^6) \ . \quad (23)$$

Let $L(r) = \omega^{-1}\arctan(x)$ with $a = \omega^{-1}$, $x = 2r\tan(b)$, and $b = 0.5\omega$, we substitute the Taylor polynomials and a, b, x into Eq. (14) to obtain a

8

biquadratic polynomial $Q(\omega)$ independent of r:

$$L(r) \approx \underbrace{1 + \frac{1}{12}\omega^2 + \frac{1}{120}\omega^4}_{=:Q(\omega)} \ . \qquad (24)$$

By equating the right-hand side of Eq. (24) to k

$$Q(\omega) = k \qquad (25)$$

we can estimate ω from four roots of the resulting polynomial $Q(\omega)$. These roots can be found by substituting $z = \omega^2$ into Eq. (24), solving the quadratic equation with respect to z, and substituting back to obtain ω. This eventually results in the four solutions $\pm\sqrt{z_{1,2}}$. The solution exists only if $k \geq 1$, as complex solutions are not meaningful, and thus corresponds to the largest real positive root.

We evaluated the accuracy of the approximation in Eq. (24), the results are shown in Fig. 5. For large radii, the approximation significantly deviates from the exact solution. In practice, this means that the selected points for the estimation must ideally be well distributed across the image. Otherwise, the lens distortion parameter will be underestimated. Note that this constraint is typically not an issue due to the spatial distribution of the solar cells.

3.3.4. Minimization Criterion for the Refinement of Lens Distortion Parameters

The Levenberg-Marquardt algorithm is used to refine the estimated lens distortion parameters $\boldsymbol{\theta}$ [33, 34]. The objective function is

$$\boldsymbol{\theta}^\star := \operatorname*{argmin}_{\boldsymbol{\theta}} \frac{1}{2} \sum_{i=1}^n \chi^2(\boldsymbol{P}^{(i)}, \boldsymbol{\theta}) \qquad (26)$$

Here, $\boldsymbol{P}^{(i)} \in \mathbb{R}^{2 \times m}$ is a matrix of m 2-D points of the i-th curve. The distortion error χ^2 quantifies the deviation of the points from the corresponding ideal straight line by

$$\chi^2(\boldsymbol{P}^{(i)}, \boldsymbol{\theta}) = t_1 \sin^2\varphi - 2|t_2||\sin\varphi|\cos\varphi \\ + t_3 \cos^2\varphi \qquad (27)$$

where

$$|\sin\varphi| = \sqrt{\frac{1}{2} - \alpha}, \quad \cos\varphi = \sqrt{\frac{1}{2} + \alpha} \ . \qquad (28)$$

with

$$\alpha = \frac{t_1 - t_2}{2\sqrt{(t_1 - t_2)^2 + 4t_2^2}} \qquad (29)$$

and the coefficients t_1, t_2, t_3 are

$$t_1 = \sum_{j=1}^m x_j^2 - \frac{1}{m}\left(\sum_{j=1}^m x_j\right)^2 \qquad (30)$$

$$t_2 = \sum_{j=1}^m x_j y_j - \frac{1}{m}\left(\sum_{j=1}^m x_j\right)\left(\sum_{j=1}^m y_j\right) \qquad (31)$$

$$t_3 = \sum_{j=1}^m y_j^2 - \frac{1}{m}\left(\sum_{j=1}^m y_j\right)^2 \ . \qquad (32)$$

The undistorted image coordinates $\boldsymbol{p}_j := (x_j, y_j)^\top \in \Omega$ are computed as $\boldsymbol{p}_j = \boldsymbol{u}(\boldsymbol{q}_j)$ by applying the inverse lens distortion Eq. (17) to the points $\boldsymbol{q}_j \in \boldsymbol{Q}^{(i)}$ of each curve to obtain $\boldsymbol{p}_j \in \boldsymbol{P}^{(i)}$.

Following Devernay and Faugeras [8], we iteratively optimize the parameters $\boldsymbol{\theta}$. In every step t, we optimize the parameters and then compute the overall error (27) $\epsilon_t := \sum_{i=1}^n \chi^2(\boldsymbol{P}^{(i)}, \boldsymbol{\theta})$ over all curve points. Afterwards, we undistort the curve points and continue the optimization until the relative change in error $\epsilon := (\epsilon_{t-1} - \epsilon_t)/\epsilon_t$ falls below a threshold $\epsilon = 10^{-6}$.

Minimizing the objective function (27) for all parameters simultaneously may cause the optimizer to get stuck in a local minimum. We therefore optimize a subset of parameters in several partitions starting with ω only. Afterwards, we additionally optimize the distortion center $\boldsymbol{c}$. Finally, the parameters $\boldsymbol{\theta}$ are jointly optimized.

3.3.5. Obtaining a Consistent Parabolic Curve Grid Model

The layout of the curves is constrained to a grid to eliminate many outlier curves. Ideally, each horizontally oriented parabola should intersect each vertically oriented parabola exactly once. Also, every parabolic curve should not intersect other parabolic curves of same orientation within the image plane. This set of rules eliminates most of the outliers.

Intersections of Two Parabolic Curves. Curves are modeled as second order polynomials in Eq. (8). Denoting the coefficients of a horizontal curve by a_2, a_1, a_0, and the coefficients of a vertical curve by b_2, b_1, b_0, the position x of a curve intersection is the solution to

$$a_2^2 b_2 x^4 + 2a_1 a_2 b_2 x^3 + x^2\left(2a_0 a_2 b_2 + a_1^2 b_2 + a_2 b_1\right) \\ + x(2a_0 a_1 b_2 + a_1 b_1 - 1) \\ + a_0^2 b_2 + a_0 b_1 + b_0 = 0 \ . \qquad (33)$$

(a) Segmented solar cells (b) Average solar cell (c) Vertical profile (d) Horizontal profile (e) Augmented mask (f) Extra geometry (g) Final mask
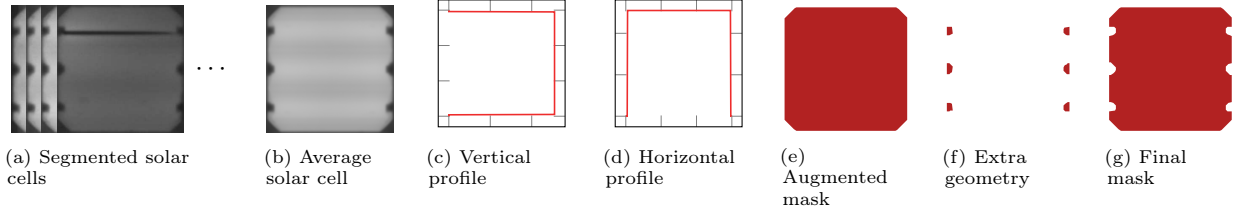
Figure 6: Intermediate steps of the solar mask estimation process.

The real roots of the quartic (33) can be found with the Jenkins-Traub RPOLY algorithm [35]. The corresponding values $f(x)$ are determined by inserting the roots into Eq. (8).

Robust Outlier Elimination. Locally Optimized RAndom Sample Concensus (LO-RANSAC) [36] is used to remove outlier curves. In every LO-RANSAC iteration, the grid constraints are imposed by randomly selecting two horizontal and two vertical curves to build a minimal grid model. Inliers are all curves that (1) exactly once intersect the model grid lines of perpendicular orientation, (2) not intersect the model grid lines of parallel orientation, and (3) whose MSE of the reprojected undistorted points is not larger than one pixel.

Remaining Curve Outliers. Halos around the solar modules and holding mounts (such as in Fig. 4) can generate additional curves outside of the cells. We apply Otsu thresholding on the contrast normalized image [23] and discard those outer curves where the average intensity of the enclosed regions are below the automatically determined threshold.

3.4. Estimation of the Solar Module Topology

A topology constraint on the solar cell is used to remove remaining non-cell curves in the background of the PV module, and the number and layout of solar cells is estimated. Outliers prevent a direct estimation of the number of solar cell rows and columns in the PV module. Additionally, we assume that the number and orientation of segments dividing each solar cell are unknown. To simplify the task, we assume that solar cells are square, i.e., horizontal and vertical cell boundary lines are equally long.

3.4.1. Estimation of the Solar Cell Subdivisions and the Number of Rows and Columns

The solar cells and their layout are inferred from the statistics of the line segment lengths in horizontal and vertical direction. We collect these lengths separately for each dimension and cluster them. DB-SCAN clustering [37] to simultaneously estimate cluster membership and the number of clusters. Despite the presence of outlier curves, the clusters will be representative of the distribution of segment dimensions within a cell. For example, if a solar cell consists of three vertically arranged segments with heights of $20 : 60 : 20$ pixels, the two largest clusters will have the medians 60 and 20. With the assumption that the segment arrangement is typically symmetric, the number of segments is estimated as the number of clusters times two minus one. If clustering yields a single cluster, we assume that the solar cells consist of a single segment. Outlier curves or segments, respectively, are rejected by only considering the largest clusters, with the additional constraint that the sizes of the used clusters are proportional to each other, and that not more than four segments can be expected in a cell. The number of rows and columns of a solar cell is determined by dividing the overall size of the curve grid by the estimated cell side lengths.

3.4.2. Curve Grid Outlier Elimination

The proportions are used to generate a synthetic planar grid that is registered against the curve grid intersections. Specifically, we use the rigid point set registration of Coherent Point Drift (CPD) [38]. The method requires choosing a weight $0 \leq w \leq 1$ of the uniform distribution that accounts for noise and outliers, which we estimate as the proportion of points in the synthetic planar grid and the total number of intersections in the curve grid.

The correspondences between the synthetic grid points and the points in the curve grid may not be unique if outlier rows or columns are present. Thus, if the curve grid contains more points than the synthetic grid, we randomly sample a subset of curve grid intersections of size corresponding to the number of points in the synthetic grid to enhance the robustness.

3.4.3. Undistortion and Rectification

The PV module configuration is used to undistort the whole image using Eq. (17). After eliminating the lens distortion, we use Direct Linear Transform (DLT) [30] to estimate the planar 2-D homography using the four corners of the curve grid with respect to the corners of the synthetic planar grid.

The intersections of the undistorted curve grid may not align exactly with respect to the synthetic planar grid. The remaining misalignment is corrected via affine moving least-squares Schaefer et al. [39], which warps the image using the planar grid intersections as control points and curve grid intersections as deformed positions.

3.5. Estimation the Solar Module Cell Mask

Extracted solar cells of a PV module can be used to generate a solar cell mask representing the exact shape of a solar cell without background and busbars (see Fig. 6). First, we use multiple solar cell regions from a PV module to compute a mean solar cell (see Figs. 6(a) to 6(b)). We then apply adaptive thresholding on 25×25 pixels patches using their mean intensity, followed by morphological opening and flood filling to close any remaining holes. This leads to a binary mask.

Ragged edges at the contour are removed using vertical and horizontal cell profiles (Figs. 6(c) to 6(d)). The profiles are computed as the median of the mask along each image row or column, respectively. We combine the backprojection of these profiles with the convex hull of the binary mask to account for cut-off edges (Fig. 6(e)) using bitwise AND. Oftentimes the side of a solar cell is not straight, for example due to dimples (see Fig. 6(f)). To address this, we combine the initial binary mask and the augmented mask via bitwise XOR.

4. Evaluation

The segmentation performance is evaluated for solar cells and for their subdivision into segments.

4.1. The Dataset

We use a dataset consisting of 44 unique PV modules with various degrees of defects to manually select the parameters for the segmentation pipeline. In this dataset, 26 solar modules are of monocrystalline type, and 18 of monocrystalline type. The 44 solar modules consist in total of 2,624 solar cells of which 715 are definitely defective with defects
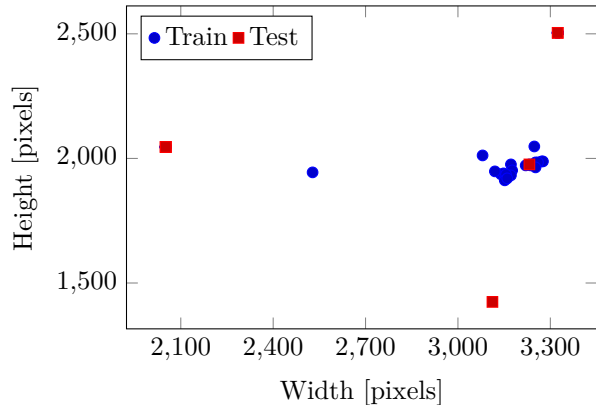

Figure 7: Distribution of EL image dimensions in the training and test datasets.

ranging from micro cracks to completely disconnected cells and deep cracks. 106 solar cells exhibit smaller defects that are not with certainty identifiable as completely defective, and 295 solar cells feature miscellaneous surface abnormalities that are no defects. The remaining 1,508 solar cells are categorized as functional without any perceivable surface abnormalities.

The average resolution of the EL images is $2{,}779.63 \times 2{,}087.35$ pixels with standard deviations of 576.42 and 198.30 pixels, respectively. The median resolution is $3{,}152 \times 2{,}046$ pixels. Fig. 7 shows the distribution of image dimensions.

Ground truth is created by hand-labeling the cell segments. The ground truth also specifies the rows and columns of the solar cells and their subdivisions.

4.2. Quantitative Results

We evaluate the segmentation accuracy of our approach on eight EL images from different PV modules that contain a total of 408 solar cells. Four of these modules are monocrystalline and the remaining four are polycrystalline modules.

Three monocrystalline modules consist of 4×9 cells, the remaining monocrystalline module consists of 6×10 cells. All of these cells are subdivided by the busbars into 3×1 segments.

The polycrystalline modules consist of 6×10 solar cells each. In two of the modules, every cell is subdivided into 3×1 segments. The cells of the other two modules are subdivided into 4×1 segments.

The resolutions ($W \times H$) of the test EL images are in the range of $2{,}649.50 \pm 643.20 \times 2{,}074 \pm 339.12$ with a median resolution of $2{,}581.50 \times 2{,}046$.

Table 1: Overview of the precision, recall, and F_1 scores grouped by the type of the solar module and their combined scores.

Solar wafer	Precision	Recall	F_1 score
Monocrystalline	100 %	90.48 %	95 %
Polycrystalline	100 %	97.5 %	98.73 %
Overall	100 %	94.61 %	97.23 %

4.2.1. Evaluation Metrics

To allow an exact comparison of the segmentation results to the ground truth masks, we warp the estimated solar cell masks using the determined perspective projection and lens distortion parameters. That way, the estimated solar module mask will ideally exactly overlay the hand-labeled ground truth masks of the test dataset.

The performance metrics are computed on pairs of segmented cells and ground truth masks. A ground truth cell mask is matched to the segmented cell with the largest intersection area.

The first set of performance metrics are precision, recall, and the F_1 score. These metrics are computed by considering cell segmentation as a binary pixel-wise classification into cell and background pixels. A matching cell pixel is a true positive, the remaining quantities are defined accordingly.

The second performance metric is the weighted Jaccard index J [40, 41]. This metric extends the common Jaccard index by an importance weighting of the input pixels. As the compared masks are not strictly binary either due to antialiasing or interpolation during mask construction, we define importance of pixels by their intensity. Given two non-binary masks A and B, the weighted Jaccard similarity is

$$J_w = \frac{\sum_{\boldsymbol{u}\in\Omega} \min\{A(\boldsymbol{u}), B(\boldsymbol{u})\}}{\sum_{\boldsymbol{u}\in\Omega} \max\{A(\boldsymbol{u}), B(\boldsymbol{u})\}} \ . \quad (34)$$

4.2.2. Background-Foreground Separation

Precision, recall and F_1 metrics for cell segmentation are reported in Table 1. Precision summarizes the ability to identify the background, and it reaches a perfect score of 100 % on the eight test images. Recall summarizes the ability to identify the foreground, and is on average at 94.61 %. For monocrystalline modules, recall is 90.48 % whereas for polycrystalline solar modules it is 97.50 %. Manual investigation of the failure cases reveals for monocrystalline modules difficulties with large gaps
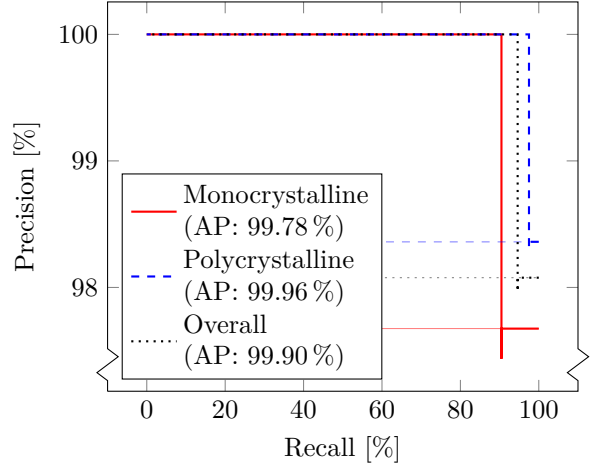

Figure 8: Precision-Recall (PR) curves for the test images. The Area Under the Curve (AUC) of the PR curve is given by the Average Precision (AP). The horizontal semi-opaque lines correspond to the baseline for random guessing. All PR curves are well above the baselines.
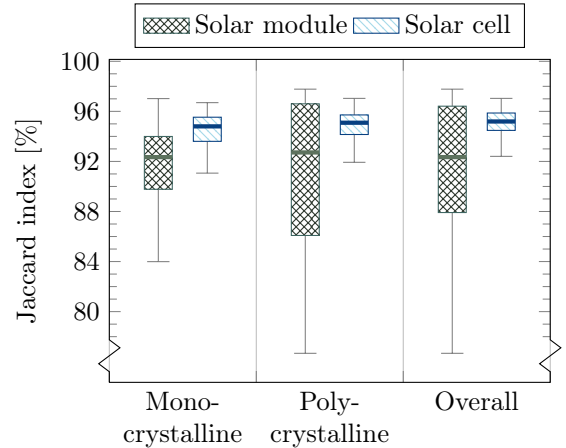

Figure 9: Segmentation accuracy in terms of the weighted Jaccard index. The boxplots ▦ denote how accurately the entire solar module was segmented. The boxplots ▨ specify the segmentation accuracy of individual solar cells. Non-detected cells are omitted here.

between cells combined with cracks, and for polycrystalline modules difficulties with ragged edges during parabolic curve fitting (see Section 4.3 for more details). The F_1 score summarizes the harmonic mean of precision and recall. The overall F_1 score for the eight EL images is 97.23 %. For monocrystalline modules the F_1 score is 95 % and for polycrystalline modules it is 98.73 %.

We also varied the threshold for precision and recall. The respective curve is shown in Fig. 8. In
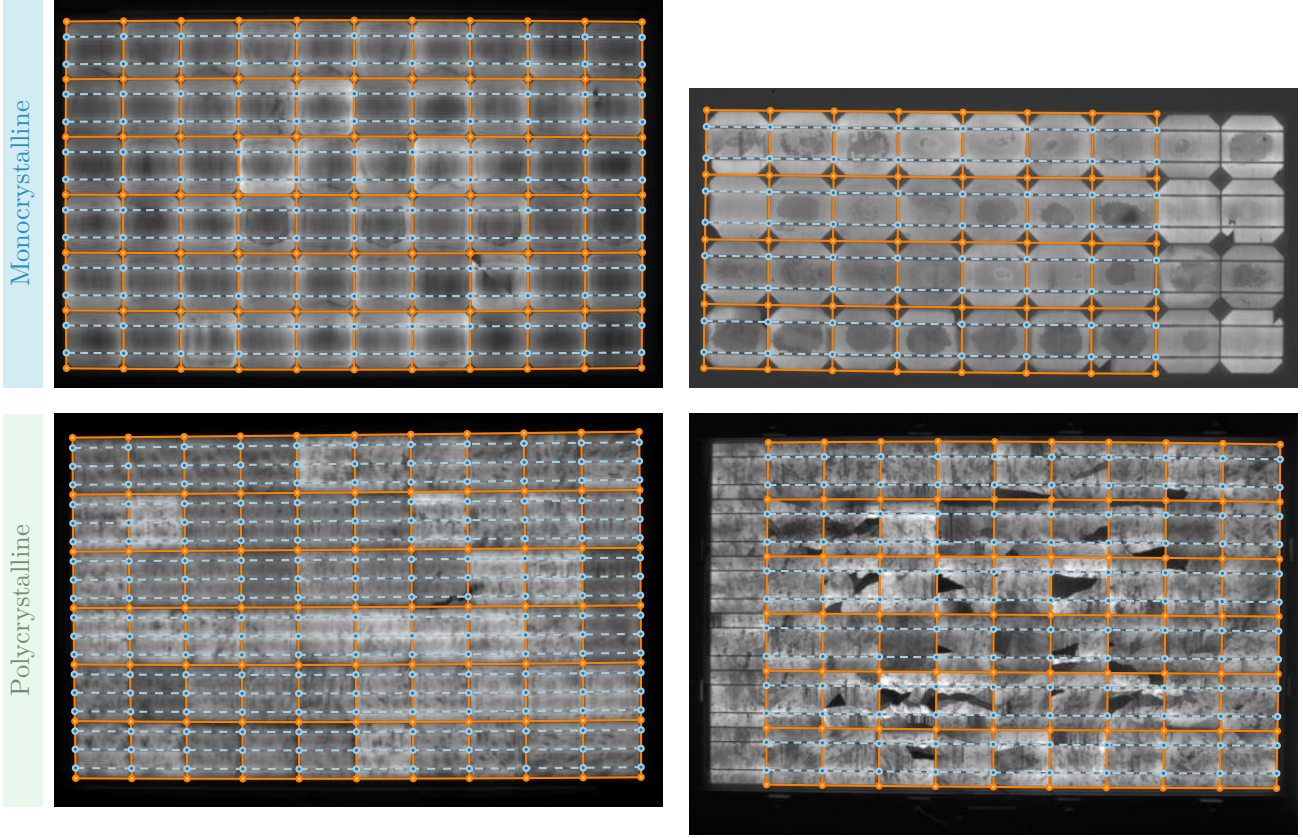
Figure 10: Qualitative segmentation results of four test images depicting the estimated curve grid superimposed over the contrast-normalized input EL image. Images are cropped for visualization.

this experiment, the average precision is at 99.90 %. The baselines, a performance marker for random guessing, are shown as lighter horizontal lines. All PR curves are well above the baselines.

4.2.3. Segment Overlap

The weighted Jaccard index characterizes the segmentation accuracy with respect to the shape of the solar modules and cells. Fig. 9 shows the results as boxplots grouped by the type of solar module and their weighted Jaccard index.

The overall median Jaccard score for segmented solar modules is 92.34 % and 95.09 % for solar cells. For monocrystalline solar modules the score is 92.34 % and for the corresponding solar cells it is 94.80 %. For polycrystalline solar modules the Jaccard score is 92.72 % and for the corresponding solar cells it is 95.19 %. The scores for the modules have a wider spread than the scores for the individual cells, since the scores of the modules include also undetected cells. Overall, the proposed algorithm

segmentation is highly accurate. Particularly once a cell is detected, the cell outline is accurately and robustly segmented.

4.3. Qualitative Results

Fig. 10 shows the qualitative results of the segmentation pipeline on four test images. The two results in the top row are computed on monocrystalline modules, the two results in the bottom on polycrystalline modules. The estimated solar module masks are highly accurate. Even in presence of false negatives (i.e., missed solar cells), the accuracy of the predicted solar module mask is not affected. In the case of monocrystalline solar modules, false negatives arise due to large gaps between solar cell ridges and the relatively short height of the module. False negatives in the polycrystalline solar module are caused by the mount close to the edge of the solar module. This leads to large fitting errors causing parabolic curves to be discarded during curve grid model estimation (see also Section 4.5).
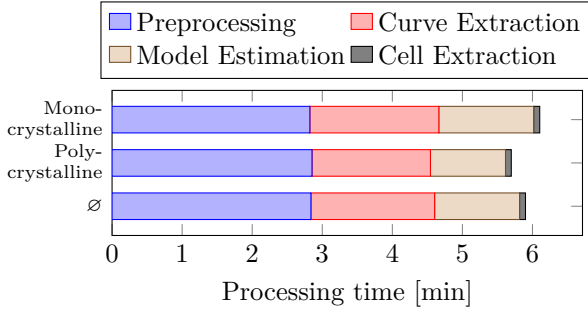
Figure 11: Cumulative processing time of the segmentation pipeline with respect to different solar module types and the distribution of time taken by individual steps, in minutes.

4.4. Runtime Evaluation

Figure 11 shows the runtimes for all 44 images on a mobile workstation with an Intel Xeon E3-1505M CPU clocked at 2.80 GHz with 32 GB of RAM. The first three stages of the segmentation pipeline are implemented in C++ whereas the last stage (except for moving least-squares image deformation) is implemented in Python.

Preprocessing is computationally most expensive, cell extraction is on average cheapest. The standard deviation of the curve extraction and model estimation, however, is larger than that of the preprocessing and the cell extraction (see Fig. 12). This is mostly due to dependency upon the total number of ridge edges and the number of resulting curves combined with the probabilistic nature of LO-RANSAC.

Interestingly, processing EL images of monocrystalline solar modules takes slightly longer on average than processing polycrystalline solar modules. This is due to large gaps between ridges caused by cut-off edges that produce many curve segments which are not merged. Conversely, curve segments in polycrystalline solar modules are closer, which makes it more likely that several curve segments are combined early on.

4.5. Limitations

Mounts that hold PV modules may cause spurious ridge edges. Early stages of the segmentation focus on ridges without analyzing the whole image content, which may occasionally lead to spurious edges and eventually to an incorrect segmentation.

While the algorithm is in principle able to process disconnected (dark) cells, rows or columns with more than 50 % of disconnected cells are wrongly segmented. However, we observed that also human
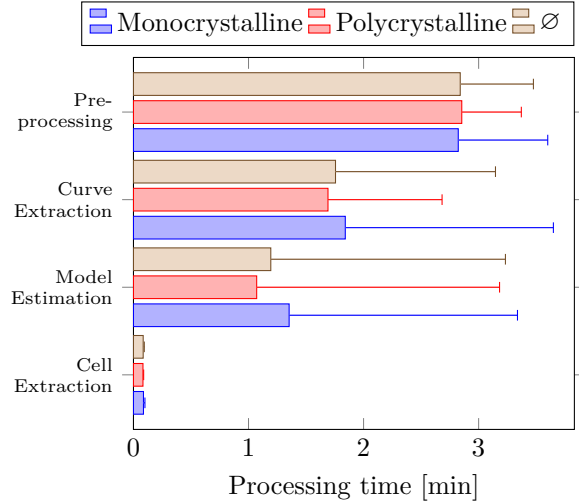

Figure 12: Time taken by individual steps of the segmentation pipeline, in minutes. The error bars denote the upper range of the standard deviation.

experts have problems to determine the contours under these circumstances.

We also observed that smooth edges can result in segmentation failures. This is because the vesselness of smooth edges is weaker and may completely fade away after non-maximum suppression. This problem is also related to situations where the inter-cell borders are exceptionally wide. In such cases, it is necessary to adjust the parameters of the vesselness filter and the tensor voting.

5. Conclusions

In this work, we presented a fully automatic segmentation to extract solar cells from high resolution EL images. The proposed segmentation is robust to underexposure, and works robustly in presence of severe defects on the cell. This can be attributed to the proposed preprocessing and the vesselness filtering, coupled with tensor voting to robustly determine the inter-cell borders. The segmentation is highly accurate, which allows to use its output for further inspection tasks, such as automatic classification of defective solar cells and the prediction of power loss.

We evaluated the segmentation in terms of the Jaccard index on eight different PV modules consisting of 408 hand-labeled solar cells. The approch is able to segment solar cells with an accuracy of 95.09 %. In terms of classification performance, the segmentation pipeline reaches an F_1 score of 97.23 %.

Beyond the proposed applications, the method can also be useful to bootstrap deep learning architectures that could be trained end-to-end to directly segment the solar cells.

Acknowledgments

References

[1] R. O. Duda, P. E. Hart, Use of the Hough Transformation to Detect Lines and Curves in Pictures, Commun. ACM 15 (1) (1972) 11–15, ISSN 0001-0782, doi:10.1145/361237.361242.

[2] M. Rufli, D. Scaramuzza, R. Siegwart, Automatic Detection of Checkerboards on Blurred and Distorted Images, in: 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, ISSN 2153-0858, 3121–3126, doi:10.1109/IROS.2008.4650703, 2008.

[3] S. Placht, P. Fürsattel, E. A. Mengue, H. Hofmann, C. Schaller, M. Balda, E. Angelopoulou, ROCHADE: Robust Checkerboard Advanced Detection for Camera Calibration, in: D. Fleet, T. Pajdla, B. Schiele, T. Tuytelaars (Eds.), Computer Vision – ECCV 2014, vol. 8692 of *Lecture Notes in Computer Science*, Springer, ISBN 978-3-319-10592-5, 766–779, doi:10.1007/978-3-319-10593-2_50, 2014.

[4] P. Fuersattel, S. Dotenco, S. Placht, M. Balda, A. Maier, C. Riess, OCPAD – Occluded Checkerboard Pattern Detector, in: 2016 IEEE Winter Conference on Applications of Computer Vision (WACV), 2016.

[5] M. Hoffmann, A. Ernst, T. Bergen, S. Hettenkofer, J.-U. Garbas, A Robust Chessboard Detector for Geometric Camera Calibration, in: Proceedings of the 12th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2017), ISBN 978-989-758-225-7, 34–43, doi:10.5220/0006104300340043, 2017.

[6] H. Ha, M. Perdoch, H. Alismail, I. S. Kweon, Y. Sheikh, Deltille Grids for Geometric Camera Calibration, in: 2017 IEEE International Conference on Computer Vision (ICCV), 5354–5362, doi:10.1109/ICCV.2017.571, 2017.

[7] D. C. Brown, Close-range Camera Calibration, Photogrammetric Engineering & Remote Sensing 37 (1971) 855–866, ISSN 03331024, doi:10.1.1.14.6358.

[8] F. Devernay, O. Faugeras, Straight Lines Have to Be Straight, Machine Vision and Applications 13 (1) (2001) 14–24, ISSN 0932-8092, doi:10.1007/PL00013269.

[9] A. Fitzgibbon, Simultaneous Linear Estimation of Multiple View Geometry and Lens Distortion, IEEE Computer Society Conference on Computer Vision and Pattern Recognition 1 (2001) I–125–I–132, ISSN 1063-6919, doi:10.1109/CVPR.2001.990465.

[10] D. Claus, A. W. Fitzgibbon, A Plumbline Constraint for the Rational Function Lens Distortion Model, Proceedings of the British Machine Vision Conference Sep (2005) 99–108, doi:10.5244/C.19.10.

[11] D. Claus, A. W. Fitzgibbon, A Rational Function Lens Distortion Model for General Cameras, in: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), vol. 1, IEEE, ISBN 0-7695-2372-2, ISSN 1063-6919, 213–219, doi:10.1109/CVPR.2005.43, 2005.

[12] E. Rosten, R. Loveland, Camera Distortion Self-calibration Using the Plumb-line Constraint and Minimal Hough Entropy, Machine Vision and Applications 22 (1) (2011) 77–85, ISSN 0932-8092, doi:10.1007/s00138-009-0196-9.

[13] R. B. Girshick, J. Donahue, T. Darrell, J. Malik, Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation, CoRR abs/1311.2524, URL http://arxiv.org/abs/1311.2524.

[14] R. B. Girshick, Fast R-CNN, CoRR abs/1504.08083, URL http://arxiv.org/abs/1504.08083.

[15] S. Ren, K. He, R. Girshick, J. Sun, Faster R-CNN: Towards Real-time Object Detection with Region Proposal Networks, in: Advances in Neural Information Processing Systems, 91–99, 2015.

[16] J. Long, E. Shelhamer, T. Darrell, Fully Convolutional Networks for Semantic Segmentation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 3431–3440, 2015.

[17] Y. Li, H. Qi, J. Dai, X. Ji, Y. Wei, Fully Convolutional Instance-aware Semantic Segmentation, CoRR abs/1611.07709, URL http://arxiv.org/abs/1611.07709.

[18] K. He, G. Gkioxari, P. Dollár, R. B. Girshick, Mask R-CNN, CoRR abs/1703.06870, URL http://arxiv.org/abs/1703.06870.

[19] E. Franken, P. Rongen, M. van Almsick, B. ter Haar Romeny, Detection of Electrophysiology Catheters in Noisy Fluoroscopy Images, in: R. Larsen, M. Nielsen, J. Sporring (Eds.), International Conference on Medical Image Computing and Computer-Assisted Intervention, vol. 9 of *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, Berlin, Heidelberg, ISBN 978-3-540-44727-6, ISSN 0302-9743, 25–32, doi:10.1007/11866763, 2006.

[20] A. F. Frangi, W. J. Niessen, K. L. Vincken, M. A. Viergever, Multiscale Vessel Enhancement Filtering, in: W. M. Wells, A. Colchester, S. Delp (Eds.), International Conference on Medial Image Computing and Computer-Assisted Invervention, vol. 1496, Springer Berlin Heidelberg, ISBN 978-3-540-49563-5, ISSN 1361-8415, 130–137, doi:10.1007/BFb0056195, 1998.

[21] G. Medioni, C.-K. Tang, M.-S. Lee, Tensor Voting: Theory and Applications, in: Proceedings of RFIA, 2000.

[22] E. Franken, M. van Almsick, P. Rongen, L. Florack, B. ter Haar Romeny, An Efficient Method for Tensor Voting Using Steerable Filters, in: A. Leonardis, H. Bischof, A. Pinz (Eds.), Computer Vision – ECCV 2006, Springer Berlin Heidelberg, ISBN 978-3-540-33839-0, 228–240, doi:10.1007/11744085_18, 2006.

[23] N. Otsu, A Threshold Selection Method from Gray-level Histograms, IEEE Transactions on Systems, Man, and Cybernetics 9 (1) (1979) 62–66, ISSN 0018-9472, doi:10.1109/TSMC.1979.4310076.

[24] K. Saeed, M. Tabędzki, M. Rybnik, M. Adamski, K3M: A Universal Algorithm for Image Skeletonization and a Review of Thinning Techniques, International Journal of Applied Mathematics and Computer

Science 20 (2) (2010) 317–335, ISSN 1641-876X, doi:10.2478/v10006-010-0024-4.

[25] P. Kovesi, MATLAB and Octave Functions for Computer Vision and Image Processing, URL http://www.peterkovesi.com/matlabfns, 2017.

[26] A. Fabijańska, A Survey of Subpixel Edge Detection Methods for Images of Heat-emitting Metal Specimens, International Journal of Applied Mathematics and Computer Science 22 (3) (2012) 695–710, ISSN 2083-8492, doi:10.2478/v10006-012-0052-3.

[27] J. Nocedal, S. J. Wright, Numerical Optimization, vol. 43 of *Springer Series in Operations Research and Financial Engineering*, Springer New York, 2 edn., ISBN 978-0-387-30303-1, doi:10.1007/978-0-387-40065-5, 2006.

[28] D. H. Douglas, T. K. Peucker, Algorithms for the Reduction of the Number of Points Required to Represent a Digitized Line or its Caricature, Cartographica: The International Journal for Geographic Information and Geovisualization 10 (2) (1973) 112–122, doi:10.3138/fm57-6770-u75u-7727.

[29] M. A. Fischler, R. C. Bolles, Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography, Communications of the ACM 24 (6) (1981) 381–395, ISSN 0001-0782, doi:10.1145/358669.358692.

[30] R. Hartley, A. Zisserman, Multiple View Geometry in Computer Vision, Cambridge University Press, New York, NY, USA, second edn., ISBN 0521540518, 2004.

[31] R. Y. Tsai, A Versatile Camera Calibration Technique for High-accuracy 3D Machine Vision Metrology Using Off-the-shelf TV Cameras and Lenses, IEEE Journal on Robotics and Automation 3 (4) (1987) 323–344, ISSN 0882-4967, doi:10.1109/JRA.1987.1087109.

[32] G. H. Golub, C. F. Van Loan, Matrix Computations, Johns Hopkins Studies in the Mathematical Sciences, Johns Hopkins University Press, 4th edn., ISBN 978-1421407944, 2013.

[33] K. Levenberg, A Method for the Solution of Certain Non-linear Problems in Least Squares, Quarterly of Applied Mathematics 2 (2) (1944) 164–168, ISSN 0033569X, 15524485, URL http://www.jstor.org/stable/43633451.

[34] D. W. Marquardt, An Algorithm for Least-Squares Estimation of Nonlinear Parameters, Journal of the Society for Industrial and Applied Mathematics 11 (2) (1963) 431–441, doi:10.1137/0111030.

[35] M. A. Jenkins, J. F. Traub, A Three-stage Algorithm for Real Polynomials Using Quadratic Iteration, SIAM Journal on Numerical Analysis 7 (4) (1970) 545–566, doi:10.1137/0707045.

[36] O. Chum, J. Matas, J. Kittler, Locally Optimized RANSAC, in: Joint Pattern Recognition Symposium, Springer, 236–243, 2003.

[37] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, et al., A Density-based Algorithm for Discovering Clusters in Large Spatial Databases with Noise., in: KDD, vol. 96, 226–231, 1996.

[38] A. Myronenko, X. Song, Point Set Registration: Coherent Point Drift, IEEE Transactions on Pattern Analysis and Machine Intelligence 32 (12) (2010) 2262–2275, ISSN 0162-8828, doi:10.1109/TPAMI.2010.46.

[39] S. Schaefer, T. McPhail, J. Warren, Image Deformation Using Moving Least Squares, ACM Transactions on Graphics 25 (3) (2006) 533, ISSN 0730-0301, doi:10.1145/1141911.1141920.

[40] F. Chierichetti, R. Kumar, S. Pandey, S. Vassilvitskii, Finding the Jaccard Median, in: Proceedings of the Twenty-first Annual ACM-SIAM Symposium on Discrete Algorithms, SODA '10, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, ISBN 978-0-898716-98-6, 293–311, URL http://dl.acm.org/citation.cfm?id=1873601.1873626, 2010.

[41] S. Ioffe, Improved Consistent Sampling, Weighted Minhash and L1 Sketching, in: Proceedings of the 2010 IEEE International Conference on Data Mining, ICDM '10, IEEE Computer Society, Washington, DC, USA, ISBN 978-0-7695-4256-0, 246–255, doi:10.1109/ICDM.2010.80, 2010.