Joint Special Interest Group meeting between euspen and ASPE
Dimensional Accuracy and Surface Finish in Additive Manufacturing,
KU Leuven, BE, October 2017
www.euspen.eu

In-line 3D print failure detection using computer vision

R.A. Lyngby[1], J. Wilm[1], E.R. Eiríksson[1], J.B. Nielsen[1], J.N. Jensen[1], H. Aanæs[1], D.B. Pedersen[2]

[1]*Department of Applied Mathematics and Computer Science, Technical University of Denmark*
[2]*Department of Mechanical Engineering, Technical University of Denmark*

raly@dtu.dk

Abstract

Here we present our findings on a novel real-time vision system that allows for automatic detection of failure conditions that are considered outside of nominal operation. These failure modes include warping, build plate delamination and extrusion failure. Our system consists of a calibrated camera whose position and orientation is known in the machine coordinate system. We simulate what the object under print should look like for any given moment in time. This is compared to a segmentation of the current print, and statistical detection of significant deviation. We demonstrate that this methodology precisely and unambiguously detects the time point of print failure.

Failure detection, computer vision, fused deposition modeling (FDM)

1. Introduction

Even with today's rapid development of additive manufacturing processes and techniques, in-line quality control remains as one of the key technical challenges within the field [1]. Although the topic is rapidly gaining interest in the research community, the efforts have so far mostly concerned industrial metal based 3D printers [2,3]. Many of the most popular commercially available 3D printers rely on the Fused Deposition Modeling (FDM) method. Their popularity is not least due to their low cost, which has made rapid prototyping available to small businesses, research laboratories and even consumers. Despite the popularity of FDM based systems, and their continuous development, these systems remain error prone and print jobs fail from time to time. The relatively long cycle time of these systems means that they usually operate unsupervised, which is why failures are often only detected after significant amounts of time and material has been lost. An in-line quality control system could potentially surveil the process and either stop the printing process to save time and material, or mitigate the failure to restore the print quality. However, the cost of such a system needs to be reasonably low to allow for feasible integration into low cost printing systems.

Inspired by [4], and based on our personal experience with FDM printers, we have identified three failure modes which seem to be common and symptomatic of these systems. They are:

- Workpiece warpage
- Build plate delamination
- Extrusion failure

The purpose of this paper is to explore if a computer vision system composed of only one camera and one light source can utilize relatively simple, and well known, analysis techniques to detect these failure modes.

In [4], an in-line computer vision system was presented which uses differential imaging between consecutive video frames and BLOB detection to identify the three failure modes. As their method concerns relative changes between added layers, it implicitly assumes that each layer is added correctly at first, without warpage. It further assumes that the build process does not fail when putting down the first layer. Except for the detection of extrusion failure, these two assumptions limit the sensibility of the method as well as its scalability to other failure modes.

A different idea was proposed in [5]. There, a camera array of five cameras was used to capture images of the workpiece from multiple angles. The images were saved and compared pixel-wise against images of the finished workpiece, captured after the printing process had finished. The comparisons were then used, post-manufacturing, to determine if the print had failed during manufacturing. While this approach has obvious limitations and issues, the general idea of comparing a current workpiece state against a model of that state has good potential. Our approach continues this line of thought. We use the workpiece CAD model to synthesize the expected workpiece states based on knowledge of the build plate geometry, obtained through photogrammetry.

The paper starts by detailing our approach of failure mode detection in section 2. Section 3 presents results obtained from a proof-of-concept system based on our approach, which are discussed in Section 4. Finally, Section 5 concludes the paper.

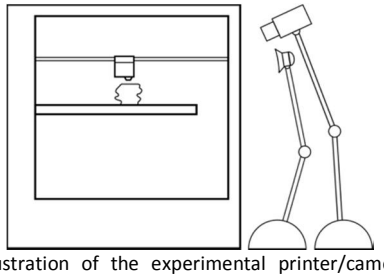

Figure 1. Illustration of the experimental printer/camera setup. A camera and a light source are positioned with good overview of the build plate.

2. Failure mode detection

Assume that we have a printer/camera system setup as illustrated on Figure 1. A color camera captures an image, $\boldsymbol{A}$, of the illuminated workpiece after each new layer is added. Using the workpiece CAD model, and perspective projection, an expected image of the workpiece geometry, $\boldsymbol{B}$, is rendered for each of the captured images. The failure modes defined in Section 1 can be jointly detected by comparing the image of the actual workpiece with that of the rendered workpiece. Note, however, that the current version of our method is not able to distinguish between the three failure-modes. Instead, it simply detects if there is *Matter Out Of Place* (MOOP).

As argued in the following sections, the interactions between light, scene and workpiece makes it difficult to render a photorealistic view of the workpiece. Inspired by [4], we overcome this issue by segmenting the workpiece, in both the captured and rendered image, and compare their segmentation masks, $s(\boldsymbol{A})$ and $s(\boldsymbol{B})$. This simplifies the error estimation to an elementwise logical exclusive disjunction between two sparse matrices with elements from the Boolean domain:

$$\boldsymbol{D} = s(\boldsymbol{A}) \oplus s(\boldsymbol{B}) \tag{1}$$

with $\boldsymbol{A}, \boldsymbol{B} \in \mathbb{B}^{N\times M\times 3}$ and $s(\boldsymbol{A}), s(\boldsymbol{B}), \boldsymbol{D} \in \mathbb{B}^{N\times M}$ where $M \times N$ is the image size. The elements in $\boldsymbol{D}$ are true wherever there is a difference between the actual view and the expected view, thus indicating an error. To intuitively understand $\boldsymbol{D}$, we need to understand how a camera pixel measures the world. Each pixel, if viewed as ideal, samples incident light, originating from a unique direction. Thus, each pixel covers a unique azimuth- and elevation angle, relative to the camera lens, why each pixel can be thought of as covering a unique frustum of the world. Our method simply predicts if a given frustum, covered by a given pixel, should contain build material or not, and then compares the actual containment of material with the predicted.

Based on $\boldsymbol{D}$, we define an error metric

$$\boldsymbol{E} = N_{\boldsymbol{D}}/N_{\text{total}} \tag{2}$$

Where $N_{\boldsymbol{D}}$ is the number of true pixels in $\boldsymbol{D}$ and N_{total} is the number of foreground pixels in the rendered mask of the complete object.

2.1. Workpiece segmentation

When segmenting the workpiece from a captured image, we utilize the color of the printing material. As our approach is targeted at low-cost consumer printers, the material is expected to be monochrome PLA plastics. The camera captures color images using the Bayer filter technique [6]. The color of each pixel is represented by three intensities, one for each of Red, Green and Blue (RGB). In combination, these three colors can simulate most all human perceivable colors. While good for humans, this color representation is not ideal for algorithmic segmentation of colored objects. Instead, we transform the color space from RGB to Hue, Saturation and Value (HSV) [7]. The HSV space intuitively represents color, with Hue representing the pure color as an angle from $0°$ to $360°$, and Saturation and Value representing the saturation and intensity of the color as values from 0 to 1.

Given a pixel at position $\boldsymbol{x} \in \mathbb{Z}^2$ in the image, with RGB values $r_{\boldsymbol{x}}, g_{\boldsymbol{x}}, b_{\boldsymbol{x}} \in [0,255]$, we first normalize the values:

$$r'_{\boldsymbol{x}} = r_{\boldsymbol{x}}/255 \tag{3}$$
$$g'_{\boldsymbol{x}} = g_{\boldsymbol{x}}/255 \tag{4}$$
$$b'_{\boldsymbol{x}} = b_{\boldsymbol{x}}/255 \tag{5}$$

Then, the maximum intensity, the minimum intensity and their difference are computed:

$$C_{\max} = \max(r'_{\boldsymbol{x}}, g'_{\boldsymbol{x}}, b'_{\boldsymbol{x}}) \tag{6}$$
$$C_{\min} = \min(r'_{\boldsymbol{x}}, g'_{\boldsymbol{x}}, b'_{\boldsymbol{x}}) \tag{7}$$
$$\Delta = C_{\max} - C_{\min} \tag{8}$$

Having those, the HSV representation is computed by:

$$h_{\boldsymbol{x}} = \begin{cases} 0° & \Delta = 0 \\ 60° \cdot \operatorname{mod}\left(\frac{g'_{\boldsymbol{x}} - b'_{\boldsymbol{x}}}{\Delta}, 6\right) & C_{\max} = r'_{\boldsymbol{x}} \\ 60° \cdot \left(\frac{b'_{\boldsymbol{x}} - r'_{\boldsymbol{x}}}{\Delta} + 2\right) & C_{\max} = g'_{\boldsymbol{x}} \\ 60° \cdot \left(\frac{r'_{\boldsymbol{x}} - g'_{\boldsymbol{x}}}{\Delta} + 4\right) & C_{\max} = b'_{\boldsymbol{x}} \end{cases} \tag{9}$$

$$s_{\boldsymbol{x}} = \begin{cases} 0 & C_{\max} = 0 \\ \frac{\Delta}{C_{\max}} & C_{\max} > 0 \end{cases} \tag{10}$$

$$v_{\boldsymbol{x}} = C_{\max} \tag{11}$$

After the captured images are converted to the HSV color space, the PLA plastic is easily segmented by only retaining pixels with a Hue within a range corresponding to that of the material, $H_{\text{t}} = [h_{\min}, h_{\max}]$. In addition, Saturation and Value are thresholded with values T_{S} and T_{V} to remove "dark" pixels, which do not contain much information due to poor illumination. Thus, the value of a pixel at position $\boldsymbol{x}$ in the segmentation mask is computed as follows:

$$\text{s}_{\boldsymbol{x}} = \begin{cases} 1 & h_{\boldsymbol{x}} \in H_{\text{t}}, s_{\boldsymbol{x}} > T_{\text{S}}, v_{\boldsymbol{x}} > T_{\text{V}} \\ 0 & \text{otherwise} \end{cases} \tag{12}$$

This segmentation method implicitly assumes that the workpiece is uniformly illuminated by a diffuse light source and that the workpiece does not share color with any other object inside the printing chamber.

2.2. Segmentation masks versus photorealistic rendering

In principle, it is possible to create a photorealistic rendering of the workpiece. Doing so would allow for detecting a wider range of failure modes than those defined in Section 1. But, as we will describe in this section, it requires detailed knowledge of the radiometric properties of the workpiece material as well as accurate knowledge of the scene geometry to do so.

Three things happen when light crosses from one media into another. Some of the light is transmitted, some is absorbed and some is reflected [8]. We denote the proportion of the light being transmitted, absorbed and reflected as T, A and R, respectively. They sum to 1, as all the energy received by the surface must be transmitted, absorbed or reflected, thus:

$$T + A + R = 1 \tag{13}$$

The transmissivity, T, the absorptivity, A, and the reflectivity, R, are collectively known as radiometric material properties, and are defined as:

$$T = I_{\text{T}}/I_0 \tag{14}$$
$$A = I_{\text{A}}/I_0 \tag{15}$$
$$R = I_{\text{R}}/I_0 \tag{16}$$

where I_0 is the radiant intensity received at the surface of the incident media, I_T is the intensity transmitted through the media, I_A is the intensity absorbed by the media, and I_R is the intensity reflected by the media. In all cases, radiant intensity is defined as watts per steradian $[\text{W}/\text{sr}]$.

When rendering the appearance of a given media, T, A and R has to be modelled. The real world scene, in our case the print bed with workpiece and light source, is modelled digitally. This is done by defining lights, objects, materials and rendering "cameras". Different types of lights are available, but in general a light source has intensity, color, direction, and position. The object, or workpiece, is defined as a polygon mesh with a position and orientation. A material is assigned to the object which defines its radiometric properties and therefore its light interactions. The light-material interactions are modelled using angles and distances between the scene elements. A rendering "camera" defines a 2D plane onto which the image of the rendered scene is projected. It has camera parameters, such as focal length and aperture, as well as a position and orientation.

The amount of radiant intensity being absorbed, transmitted and reflected at a given position on the surface of the

workpiece depends on the incoming angle between the surface and the light source and the angle between the surface and the rendering "camera". If the surface has a thickness, i.e. any real world object, light penetrates the surface where it is scattered, with some portion of light exiting at a different location than it entered. To capture this interaction property, as well as T, A, and R, a *Bidirectional Scattering-Surface Reflectance Distribution Function* (BSSRDF) is needed [9,10]. In short, the BSSRDF relates the outgoing radiance at a given surface point to the incoming irradiance received at another point:

$$dL_r(\boldsymbol{x}_r, \boldsymbol{\omega}_r) = S(\boldsymbol{x}_i, \boldsymbol{\omega}_i, \boldsymbol{x}_r, \boldsymbol{\omega}_r) dE_i(\boldsymbol{x}_i, \boldsymbol{\omega}_i) \quad (17)$$

where $\mathrm{E}_i(\boldsymbol{x}_i, \boldsymbol{\omega}_i)$ is the irradiance $[\mathrm{W/m^2}]$ received at a surface point $\boldsymbol{x}_i$ from the direction $\boldsymbol{\omega}_i$, $L_r(\boldsymbol{x}_r, \boldsymbol{\omega}_r)$ is the radiance $[\mathrm{W/sr/m^2}]$ emitted at another surface point $\boldsymbol{x}_r$ in the direction $\boldsymbol{\omega}_r$, and $S(\boldsymbol{x}_i, \boldsymbol{\omega}_i, \boldsymbol{x}_r, \boldsymbol{\omega}_r)$ is the proportion between L_r and E_i, which per definition is the BSSRDF.

A given material have a given BSSRDF which needs to be estimated before that material can be rendered. Generally, the BSSRDF can be an analytic function or a look-up table based on radiance-irradiance measurements of densely sampled perturbations of inward and outward directions. Either way, acquiring the measurements for model fitting is time consuming, requires specialized hardware and is rather difficult to do accurately enough.

In addition to obtaining an accurate radiometric model, a process model, simulating the surface microstructure created by the 3D printer, is required. Finally, the positions of all scene elements, such as lighting, workpiece, camera, and build-plate, would have to be known. The exact required precision depends on the specific parameters of the involved materials.

We assert that the added benefits of photorealistic rendering do not outweigh the difficulties involved, when detecting the defined failure modes. Thus, a simple in-line failure detection system for low-cost printers is better off relying on the much simpler segmentation mask comparison approach outlined previously.

2.3. Perspective projection and the pinhole camera model

To render an image of the workpiece and create its segmentation mask, a model is needed for projecting a volumetric object onto the flat imaging plane formed by the image sensor. First, the workpiece CAD model is tessellated into a polyhedral object. This mesh is composed of a set of K vertices connected by edges, $W = (V, E)$, with each vertex being a homogeneous coordinate in a three dimensional Cartesian coordinate frame, $V = \{\boldsymbol{v}_k \in \mathbb{R}^4\}_{k=1}^{K}$, and each edge being a two-element subsets of V,

$E = \left\{ \{\{a, b\} : a, b \in V, a \neq b\}_{l=1}^{L_k} \right\}_{k=1}^{K}$ where L_k is the number of edges connected to vertex k.

A vertex, $\boldsymbol{v} \in \mathbb{R}^4$, is projected onto the image plane by an affine transformation matrix, $\boldsymbol{P} \in \mathbb{R}^{3\times4}$, thus creating $\boldsymbol{u} \in \mathbb{R}^3$:

$$\boldsymbol{u} = \boldsymbol{P}\boldsymbol{v} \quad (17)$$

with $\boldsymbol{u}$ being a homogeneous point on the two-dimensional plane and $\boldsymbol{P}$ defined as:

$$\boldsymbol{P} = \boldsymbol{A}[\boldsymbol{R}\ \boldsymbol{t}] \quad (18)$$

where $\boldsymbol{t} \in \mathbb{R}^3$ is a three-dimensional translation vector, $\boldsymbol{R} \in \mathbb{R}^{3\times3}$ is a three-dimensional rotation matrix, and $\boldsymbol{A} \in \mathbb{R}^{3\times3}$ is:

$$\boldsymbol{A} = \begin{bmatrix} f & \beta & \Delta x \\ 0 & \alpha f & \Delta y \\ 0 & 0 & 1 \end{bmatrix} \quad (19)$$

where f is the focal length of the camera, $(\Delta x, \Delta y)$ is the image coordinate of the principal point, α is a scaling factor, and β is a shearing factor.

In combination, $\boldsymbol{P}$ defines the perspective projection of a three-dimensional, homogeneous world point onto a two-dimensional, homogeneous image point and is known as the *pinhole camera model* [11]. The matrix $[\boldsymbol{R}\ \boldsymbol{t}]$ defines the transformation from the world coordinate frame and into the cameras coordinate frame. It is known as the *extrinsic camera parameters*. The matrix $\boldsymbol{A}$ defines the transformation from the camera coordinate frame and into the image coordinate frame. It is known as the *intrinsic camera parameters*.

2.4. Camera calibration

The camera parameters need to be estimated beforehand. Ideally, the world coordinate frame coincides with the printer's coordinate frame, such that $[\boldsymbol{R}\ \boldsymbol{t}]$ transforms points from the printer's frame into the camera's frame. We define a new coordinate frame which is spanned by the positions of four corner markers. The markers are superimposed onto the workpiece mesh, such that they are printed together and the workpiece has a known position relative to the markers. This approach gives us a well-defined coordinate frame, which can easily be found relative to both the workpiece and the camera. The markers are illustrated in Figure 2.

The four marks are found from the workpiece segmentation mask defined in equation (12). The method described by [12] is used to extract object contours from the mask. Then, algebraic ellipses are fitted to the contours using [13]. The markers contain three contours: one for each of the internal holes and one for the outer ring. We are only interested in finding the position of the outer ring. Therefore, ellipses which are completely enveloped by another ellipse are removed[1]. The centers of the four remaining ellipses closest to the image corners are used to estimate the extrinsic camera parameters.

Note that the real markers are circles which are perspectively projected onto ellipses in the image plane. Because of this, the ellipse centers will generally not coincide with the actual circle centers. Instead, they will be translated towards the camera, which leads to a small error in the estimation.

The intrinsic camera parameters are found using a set of images captured from different views of a checkerboard and follows the method described by [14].

All computer vision related processing is done using the OpenCV library [15].

2.5 Scene illumination

It is preferable to capture as much of the diffuse component of the light reflected off the workpiece as possible. It is argued in [16] that is done when the angle between the surface and the light source, and the angle between the surface and the observer, are both close to $45°$, while the direction difference between the light and observer is small, thus $\boldsymbol{\omega}_r \approx \boldsymbol{\omega}_i$. Because of this relation, we choose to place the light source as close to the camera as possible, while both are angled approximately $45°$ relative to the build plate. Besides capturing as much diffuse light as possible, this configuration also limits the shadows cast by the workpiece. The setup is shown on Figure 2.

The light source and the camera are cross-polarized. Light with one polarization direction keeps its orientation at specular reflections, but loses the orientation at diffuse reflections. Thus, specular reflections can be removed by having a $90°$ polarization difference between the camera and the light

[1] All permutations of unique pairs of ellipses are tested. The transformation which transforms one of the ellipses to the unit circle is applied to the other ellipse. If the ℓ_2 norm of the most distant point on the transformed ellipse is less than 1 then the point is completely enveloped by the other ellipse.

source. Remaining reflections in the glass build plate were eliminated by adding diffusing black tape.

Figure 2. The experimental camera and illumination setup. Notice the diffused black tape on the build plate which removes reflections.

3. Results

The method described in section 2 was applied to a test print geometry containing the letters "DTU" as shown in figure 2. This geometry is challenging to print with FDM based printers due to unsupported overhangs. Support structure generation was disabled to illustrate the detection of a failing print.

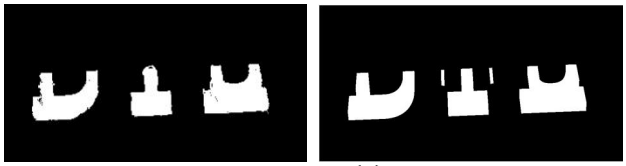

Figure 3. Illustration of image mask, $\boldsymbol{s(A)}$, at layer 71 (left) and the corresponding rendered mask, $\boldsymbol{s(B)}$ (right).

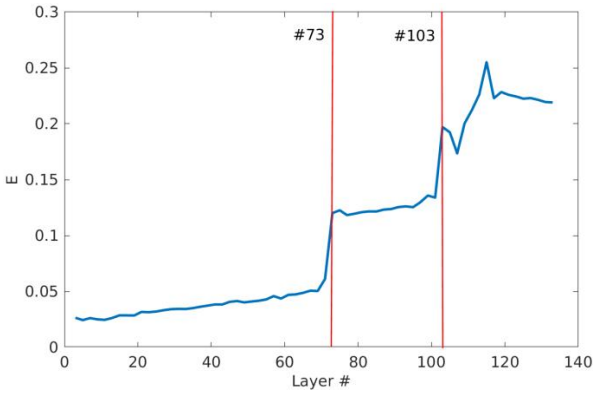

Figure 4. Plot of the error signal as a function of number of printed layers. Print failure occurs in layers 73, and further worsens at layer 103.

An example of the segmented and rendered image masks is shown in Figure 3. Figure 4 shows the evolution of the error measure as defined in Eqn. 2. Clearly, a sudden increase in $\boldsymbol{E}$ occurs before layer 73 and again at layer 103. Figure 5 shows the print shortly before and after print failure is detected. In Figure 6, the situation before and after layer 103 is shown.

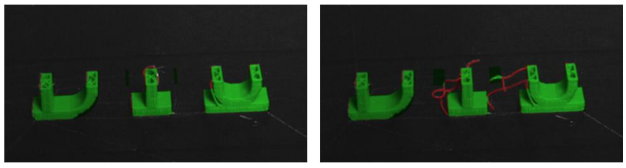

Figure 5. Close-up of the print at layer 71 (left) and layer 73 (right). False colouring was applied such that pixels in $\boldsymbol{s(A)}$ are green, pixels in $\boldsymbol{D}$ are coloured red.

It is seen that layer 73 corresponds to the emergence of severe print failure (mainly due to the unsupported serifs of the letter T). In layer 103, a large amount of hanging material is pushed into an area in which no material is expected, thus increasing the value of $\boldsymbol{E}$.

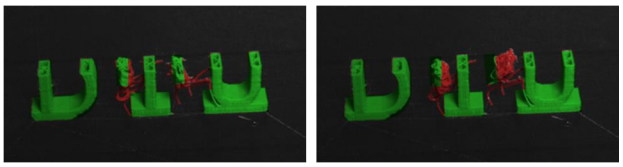

Figure 6. Close-up of the print at layer 101 (left) and layer 103 (right). False colouring was applied such that pixels in $\boldsymbol{s(A)}$ are green, pixels in $\boldsymbol{D}$ are coloured red.

4. Conclusion and Discussion

We have demonstrated a complete method for print failure detection by means of computer vision and computer rendering of the expected print geometry. Though the implemented system is a rough prototype, the results show the success of our method, and demonstrate that print failure can be detected early and very clearly using a simple camera system.

The demonstrated method is simple, low-cost, and applicable to all FDM based printers, and will detect failure in most cases of unexpected material deposition, print lifting, etc. A natural extension of the method involves multiple cameras reconstructing the visual hull of the currently printed geometry.

References

[1] Eiríksson ER et al., 2017, "Computer Vision for Additive Manufacturing", *Ph.D. thesis, DTU Compute* **447**

[2] Everton SK et al., 2016, "Review of in-situ process monitoring and in-situ metrology for metal additive manufacturing", *Materials & Design* **95** 431-445.

[3] Tapia G and Elwany A, 2014, "A review on process monitoring and control in metal-based additive manufacturing", *Journal of Manufacturing Science and Engineering* **136.6** 060801.

[4] Baumann F and Roller D, 2016, "Vision based error detection for 3D printing processes", *Int. Conf. on Frontiers of Sensors Technologies* **59** 06003.

[5] Straub J, 2015, "Initial Work on the Characterization of Additive Manufacturing (3D Printing) Using Software Image Analysis", *Machines* **3(2)** 55-71.

[6] Bayer BE, 1976, "Color imaging array", *US Patent 3,971,065*

[7] Smith AR, 1978, "Color gamut transform pairs", *Computer Graphics* **12(3)** 12–19.

[8] Callister WD and Rethwisch DG, 2011, "Materials science and engineering: An introduction", *John Wiley & Sons*, **8**.

[9] Nicodemus FE, Richmond JC, Hsia JJ, Ginsberg IW and Limperis T, 1977, "Geometrical Considerations and Nomenclature for Reflectance", *Technical Report NBS MN-160, National Bureau of Standards*

[10] Jensen HW, Marschner SR, Levoy M, Hanrahan P, 2001, "A Practical Model for Subsurface Light Transport", *Proceedings of ACM SIGGRAPH* 511–518.

[11] Hartley R and Zisserman A, 2003, "Multiple view geometry in computer vision", *Cambridge university press*.

[12] Suzuki S and Abe K, 1985, "Topological Structural Analysis of Digitized Binary Images by Border Following", *CVGIP* **30(1)** 32-46.

[13] Fitzgibbon AW and Fisher RB, 1995, "A Buyer's Guide to Conic Fitting", *Proc.5th British Machine Vision Conference* 513-522.

[14] Zhang Z, 2000, "A Flexible New Technique for Camera Calibration", *IEEE Transactions on Pattern Analysis and Machine Intelligence* **22(11)** 1330-1334.

[16] Bradski G, 2000, "The OpenCV Library", Dr. Dobb's Journal: Software for the Professional Programmer **25(11).**

[15] Nielsen JB, Jensen HW and Ramamoorthi R, 2015, "On Optimal, Minimal BRDF Sampling for Reflectance Acquisition", *ACM Transactions on Graphics* **34(6)**.