

TOOLPATH GENERATION: FREEFORM SURFACE BLENDING AND EXTRAPOLATION

Neil J. Naples and Allen Y. Yi

Department of Integrated System Engineering
The Ohio State University, Columbus, OH, USA

INTRODUCTION

The overall goal of this research is to develop enabling technologies that vastly improve existing toolpath generation techniques for freeform optics manufacturing. These new technologies allow complex optical surfaces of the form $z(x,y)$ or $z(r,\theta)$ to be efficiently machined. To this end, we introduce the concept of surface transitions: a means for transforming from explicit, freeform, optical sag equations that are defined inside arbitrary, closed boundaries into a useful toolpath. Both surface transitions discussed, blending and extrapolation, have their own explicit, continuous, polar sag equations associated with them. Defined outside the boundary of an optical surface, both techniques have the property of adjustable surface continuity so that a toolpaths' smoothness can be manipulated as desired. These two techniques differ mainly in their treatment of boundaries. Whereas extrapolation smoothly extends a surface outside its boundary to infinity, blending smoothly transitions from one non-intersecting boundary to another.

MATHEMATICAL BACKGROUND

The optical surfaces to be extrapolated or blended must be explicit (of the form $z(x,y)$ or $z(r,\theta)$), continuous, well-behaved, differentiable, and must not exhibit vertical slopes – constraints satisfied by most freeform optical prescriptions used today. For calculations to maintain the highest possible precision, the blending and extrapolating surfaces themselves were developed to work directly with this native explicit surface form. Therefore, it is not necessary to parameterize or transform any optical surface before its sag values are computed.

Boundaries

Most toolpaths consist of multiple local regions, each described by a different smooth, closed boundary. A potentially different explicit sag equation describes the global toolpath inside each local region. Therefore, it is crucial to understand how boundaries are defined and evaluated as their application directly affects the form and continuity of a toolpath.

We define discrete coordinates as the most fundamental inputs for boundary construction. These coordinates are usually supplied by an optical engineer, extracted from a CAD model, or defined as the intersection curve between two surfaces – determined by computing where the difference of two surfaces is equal to zero. After converting to polar coordinates, it is critically important that these discrete coordinates, $(\mathcal{B}_r, \mathcal{B}_\theta)$, trace out a single-valued closed curve – one non-zero radius for every angle.

Evaluating a closed boundary between the discrete coordinates that define it is of paramount importance. Conversion from discrete boundary coordinates to a smooth boundary function is accomplished with a C^2 continuous cubic interpolating spline [1], $S_r(\theta)$, where $\theta \in [0, \dots, 2\pi]$. Figure 1 shows discrete and continuous versions of an example boundary. As seen in this Figure, C^2 continuous splines exhibit non-zero curvature between discrete boundary coordinates because of the cubic polynomials used in their construction. Additionally, it is clear that boundaries defined in this way can exhibit a great deal of non-convexity.

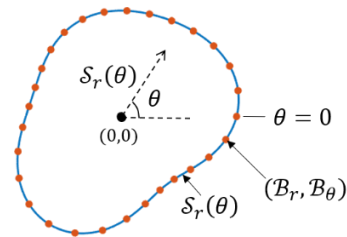


FIGURE 1. Discrete (red) and continuous (blue) boundaries. Input coordinates may be scattered.

Inside/Outside Test

One can now determine if a set of discrete toolpath coordinates lie inside or outside a polar boundary described by the above spline formulation. This is true even if the boundary is not centered on the global toolpath origin, like the boundary of a typical microlens. Illustrated in Figure 2, this calculation is critical for surfaces whose sag equation should not be evaluated outside its defined boundary.

Consider a set of toolpath coordinates, (x_i, y_i) , and the boundary from Figure 1 not centered on the toolpath origin. After translating the toolpath origin to the boundary origin and converting its coordinates to polar, the Inside/Outside test in its entirety is

$$\begin{cases} \text{inside, } r_i \leq S_r(\theta_i) \\ \text{outside, otherwise} \end{cases} \quad \forall i \in [1, \dots, N_{tp}] \quad (1)$$

where N_{tp} is the number of coordinates in the toolpath. That is, if the boundary spline has an equal or greater radius than the i^{th} translated toolpath point, the point lies inside the boundary. Otherwise, the point lies outside the boundary.

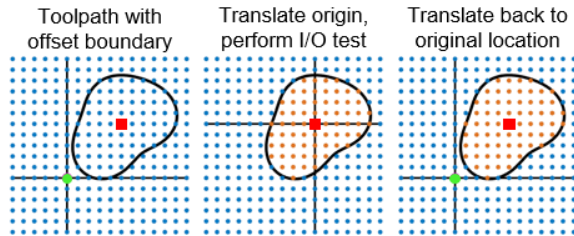


FIGURE 2. Inside/Outside Test. Boundary origin is a red square. Toolpath origin is a green circle. Toolpath coordinates may be scattered.

With the Inside/Outside test applied, calculate sag values by evaluating the explicit optical equation that is defined inside the same polar boundary, as shown in Figure 3.

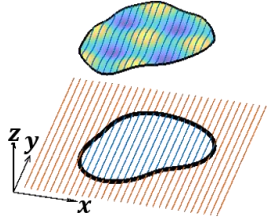


FIGURE 3. Surface described inside closed boundary. Toolpath coordinates may be scattered.

Speaking more generally, the calculations so far represent one half of the necessary framework needed to smoothly describe the entire toolpath for an explicit surface, or array of explicit surfaces, described inside arbitrary boundaries. The other half are the techniques needed to smoothly transition to the rest of the toolpath.

Extrapolating Polynomials

The extension of a 2D function outside its domain is accomplished with polynomial extrapolation. In general, an n^{th} degree extrapolating polynomial, $\mathcal{E}_n(r)$, that extrapolates the curve $z(r)$, beyond the point r_0 , is given as

$$\mathcal{E}_n(r) = \sum_{i=0}^n \frac{z^{(i)}(r_0)}{i!} (r - r_0)^i \quad (2)$$

$\forall r \in (r_0, \infty)$

for $n \geq 1$ and $z^{(i)}(r)$ being the i^{th} derivative of $z(r)$. By varying the degree of the extrapolating polynomial in Equation 2, continuity at the point of extrapolation is controlled. Assuming n derivatives of $z(r)$ exist, an n^{th} -degree extrapolating polynomial will produce a C^n continuous curve. A curve with linear, quadratic, and cubic extrapolation is shown in Figure 4.

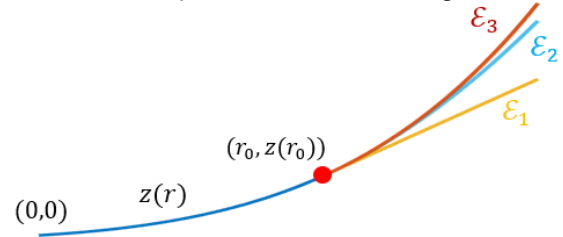


FIGURE 4. 2D curve with three extrapolating polynomials enforcing $C^1 - C^3$ continuity at the point r_0 .

Blending Polynomials

The process of constraining a polynomial to interpolate data points with known derivatives is known as Hermite interpolation [2]. An n^{th} degree 2D Hermite blending polynomial, $\mathcal{H}_n(r)$, that blends the two curves, $z_1(r)$ and $z_2(r)$, together between the coordinates r_1 and r_2 , with $r_2 > r_1$, is given as

$$\mathcal{H}_n(r) = \sum_{i=0}^n C_i r^i \quad (3)$$

$\forall r \in (r_1, r_2)$

for odd $n \geq 1$. Additionally, n obeys

$$n = 2d + 1 \quad (4)$$

where d is the number of derivatives used in the construction of $\mathcal{H}_n(r)$. The $n + 1$ polynomial coefficients, C_i , are computed by solving the $(n + 1) \times (n + 1)$ linear system of equations that constrain $z_1(r)$, $z_2(r)$, and their first $(n - 1)/2$ derivatives to be equal at the points r_1 and r_2 , respectively. For $d = 1$ (cubic, C^1 continuous interpolation) this system looks like

$$\begin{aligned} C_3 r_1^3 + C_2 r_1^2 + C_1 r_1 + C_0 &= z_1(r_1) \\ C_3 r_2^3 + C_2 r_2^2 + C_1 r_2 + C_0 &= z_2(r_2) \\ 3C_3 r_1^2 + 2C_2 r_1 + C_1 &= z_1^{(1)}(r_1) \\ 3C_3 r_2^2 + 2C_2 r_2 + C_1 &= z_2^{(1)}(r_2) \end{aligned} \quad (5)$$

where varying the degree of the blending polynomial, and thus the equations in the above linear system, controls continuity at r_1 and r_2 . Not shown here, it is highly desirable to improve the numerical properties of the above linear system by centering and scaling r_1 and r_2 to have zero mean and unit standard deviation. Assuming d -derivatives of the curve $z_1(r)$ and $z_2(r)$ exist, an n^{th} -degree blending polynomial will produce a $C^{(n-1)/2}$ continuous curve. Two curves joined with cubic, quintic, 7th order, and 9th order blending are shown in Figure 5.

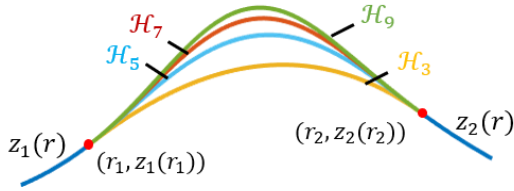


FIGURE 5. 2D curves with four blending polynomials enforcing $C^1 - C^4$ continuity at the points r_1 and r_2 .

FREEFORM SURFACE EXTRAPOLATION

Often, the overall boundary of a freeform optical surface is not equal to that of a toolpath. Some optics, higher order polynomials in particular, rapidly diverge beyond their defined boundaries. The point of freeform surface extrapolation, then, is to force a toolpath for a single surface to be smooth and well-behaved everywhere.

We will use the example freeform surface shown in Figure 6 to illustrate the extrapolation process. Since this surface, denoted Z_L , quickly diverges for large xy -inputs, it is defined inside a rectangle with corner radii, which is described by the single valued polar boundary spline $S_r(\theta)$. The circular boundary of a spiral toolpath, which will bound the final piecewise surface, is also shown in Figure 6. It is this boundary that defines the outer perimeter of the extrapolated surface.

Since $S_r(\theta)$ describes boundary radius as a function of polar angle, the rz -plane – where θ is constant – defines the location and orientation of a unique 2D extrapolating polynomial for points that lie outside a surfaces' polar boundary. Combining boundary splines and the extrapolating polynomial from Equation 2, it is now possible to generate the n^{th} -degree, explicit, continuous, extrapolating surface, Z_ε . Assuming n -derivatives of the freeform optical surface exist and can be computed, Z_ε produces a C^n continuous surface and is given as

$$Z_\varepsilon(r, \theta) = \sum_{i=0}^n \frac{Z_{L,r}^{(i)}}{i!} (r - S_r(\theta))^i \quad (6)$$

$$\forall [r \in (S_r(\theta), \infty) \wedge \theta \in [0, 2\pi]]$$

where $Z_{L,r}^{(i)}$ is the i^{th} radial derivative of Z_L relative to the center of the boundary. With this formulation, we can see that Z_ε is defined everywhere outside the polar boundary that bounds the freeform optical surface.

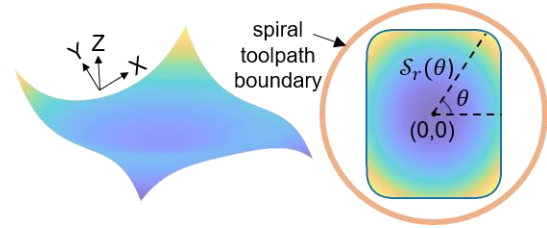


FIGURE 6. Diverging polynomial surface (left) and its optical & toolpath boundaries (right).

The Inside/Outside test is then used to compute toolpath coordinates, evaluating Z_L inside the boundary and Z_ε outside. The top row of Figure 7 shows the example surface and a series of coarsely spaced, radially oriented cubic extrapolating polynomials. When combined in Equation 6, these polynomials generate a C^3 continuous extrapolating surface. The bottom row of Figure 7 shows both spiral and raster line toolpaths applied to the overall surface, illustrating again how the overall surface is continuous and independent of the coordinates input into it.

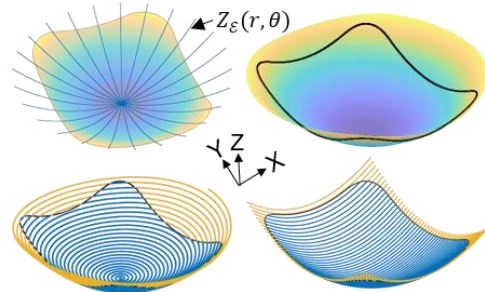


FIGURE 7. Example continuous surface and corresponding toolpaths. Radial extrapolating polynomials (top left), C^3 continuous surface before applying toolpath (top right), spiral toolpath (bottom left), raster line toolpath (bottom right).

Experimental

To experimentally verify that this surface extrapolation technique creates a smooth surface – both outside and at the boundary of an optical surface – two spiral toolpaths exhibiting linear and quartic extrapolation were generated for the

6th degree polynomial seen in Figure 8. These toolpaths are shown in Figure 9, together with the second derivative along the cutting direction of each spiral. It shown that increasing the degree of the extrapolating polynomial, and thus the level of surface continuity, produces a toolpath with smoother accelerations and decreased following error as a machine tool is running.

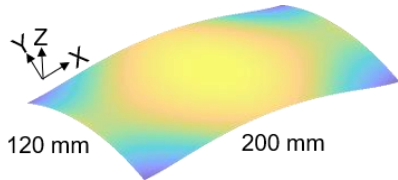


FIGURE 8. 6th degree polynomial defined inside rectangular boundary.

Ten revolutions of each toolpath were evaluated on a Precitech Nanoform X ultraprecision lathe with XZC-axes. Each toolpath was evaluated in the air so all acceleration and following error data was independent of tool force. For each spiral, the X and C-axes moved at a constant velocity due to the constant 1° angular spacing and 5µm radial stepover used, respectively. Therefore, all dynamic motion was constrained and analyzed

along the Z-axis.

The linearly extrapolated surface utilized one radial derivative in its construction. Therefore, the acceleration is expected to be fully discontinuous at the rectangular boundary. Verified in Figure 9, discontinuities in calculated and actual acceleration curves are clearly seen. These discontinuities are directly mirrored by a spike in the following error (FE) curve, resulting in a maximum magnitude of 0.962µm. Both acceleration discontinuities and FE spikes are fundamentally due to the C^1 continuous surface construction.

The surface with quartic extrapolation, on the other hand, utilized four radial derivatives in its construction. Also verified in Figure 9, the calculated and actual accelerations are totally smooth – almost sinusoidal. The equally smooth FE has a maximum magnitude of only 0.386µm. The smoothness of this toolpath is fundamentally due to the C^4 continuous surface construction. However, because this quartic surface is a higher order polynomial when compared to a linear surface, it has a larger range of sag values.

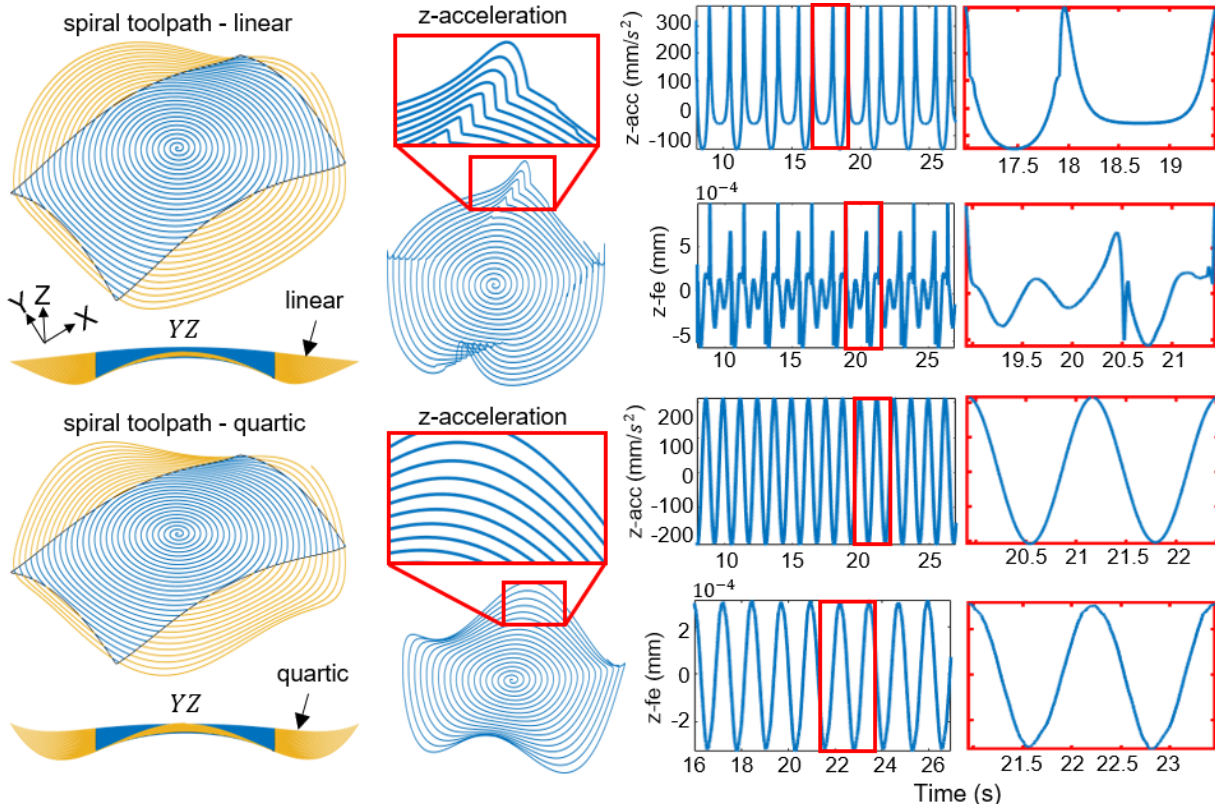


FIGURE 9. Spiral toolpaths exhibiting linear and quartic extrapolation for 6th degree polynomial defined inside rectangular boundary (left column). Stepovers increased for clarity. Computed acceleration (middle column). Acceleration and following error data taken while machine tool was running (right column).

FREEFORM SURFACE BLENDING

The desire to smoothly transition, or blend, between two explicit surfaces is a common situation faced when generating toolpaths. Even though both surfaces may be continuous and well-behaved inside their respective domains, the domains themselves are often different. Using this blending technique, it is possible to smoothly describe an array of surfaces that are blended into a larger base surface – a microlens array for example.

Like extrapolation, the divergent freeform ‘lenslet’, Z_L , will be used to describe the blending process along with a flat base surface, Z_B . In addition to the polar spline, $S_r(\theta)$, that bounds Z_L , it is necessary to define a second, larger polar spline, $S_R(\theta)$, that bounds the blend on the base surface. Illustrated in Figure 10, these two boundaries create a blend ‘zone’ such that $S_r(\theta)$ traces out a curve where the blend is at least tangent to the Z_L and $S_R(\theta)$ creates a curve where the blend is at least tangent to the Z_B .

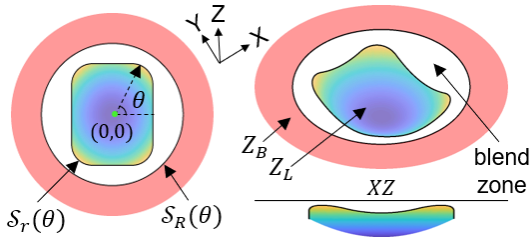


FIGURE 10. Freeform lenslet, flat base surface, and blend zone.

Since $S_r(\theta)$ and $S_R(\theta)$ both describe boundary radius as a function of polar angle, the rz -plane – where θ is constant – defines the location and orientation of a unique 2D Hermite blending polynomial for points that lie inside the blend zone defined by both polar splines. Combining boundary splines and the Hermite blending polynomial from Equation 3, it is now possible to generate the n^{th} -degree, explicit, continuous blending surface, Z_H . Assuming n -derivatives of the freeform optical surface exist and can be computed, Z_H produces a $C^{(n-1)/2}$ continuous surface and is given as

$$Z_H(r, \theta) = \sum_{i=0}^n C_i(\theta) r^i \quad (7)$$

$$\forall [r \in (S_r(\theta), S_R(\theta)) \wedge \theta \in [0, 2\pi]]$$

where $C_i(\theta)$ are the solutions to the linear system of equations that constrain each Hermite

polynomial to be at least tangent to both lenslet and base surface. An example of which, for $n = 1$, is shown in Equations 5. Since the Hermite polynomials are oriented radially, it is necessary to compute radial derivatives of the lenslet and base surface relative to the center of both boundaries.

The Inside/Outside test is then used to compute toolpath coordinates, taking care to evaluate the base surface equation for points that lie outside $S_R(\theta)$. The top row of Figure 11 shows the example lenslet and base surfaces along with a series of coarsely spaced, radially oriented 5th degree blending polynomials. When combined in Equation 7, these polynomials generate a C^2 continuous blending surface. The bottom row of Figure 11 shows both spiral and raster line toolpaths applied to the overall surface.

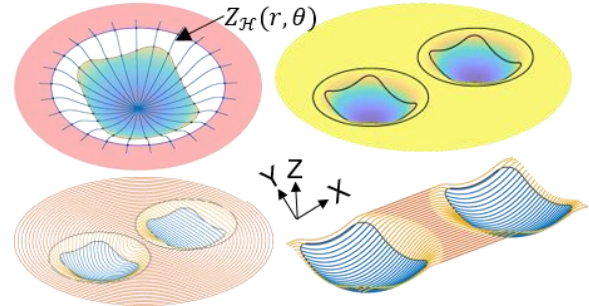


FIGURE 11. Example continuous surface and corresponding toolpaths. Radial blending polynomials (top left), C^2 continuous surface before applying toolpath (top right), spiral toolpath (bottom left), raster line toolpath (bottom right).

Experimental

To illustrate the robustness of the proposed surface generation techniques, the freeform microlens array shown in Figure 12 was machined in brass on a Precitech Nanofom X ultraprecision lathe with a Precitech FTS 5000 Fast Tool Servo. Designed with no exploitable symmetries or simplifications and virtually impossible to model with traditional CAD software, this surface is comprised of an array of explicit freeform – not spherical or aspherical – lenslets superimposed onto an explicit freeform base surface. All lenslets are individually tilted about the X and Y-axes such that they are normal to the base surface at their respective locations. All blend zones are unique and have the same form as the intersection between each respective lenslet and the base surface. Lastly, all lenslet boundary discontinuities were blended with C^3 continuous freeform blends.

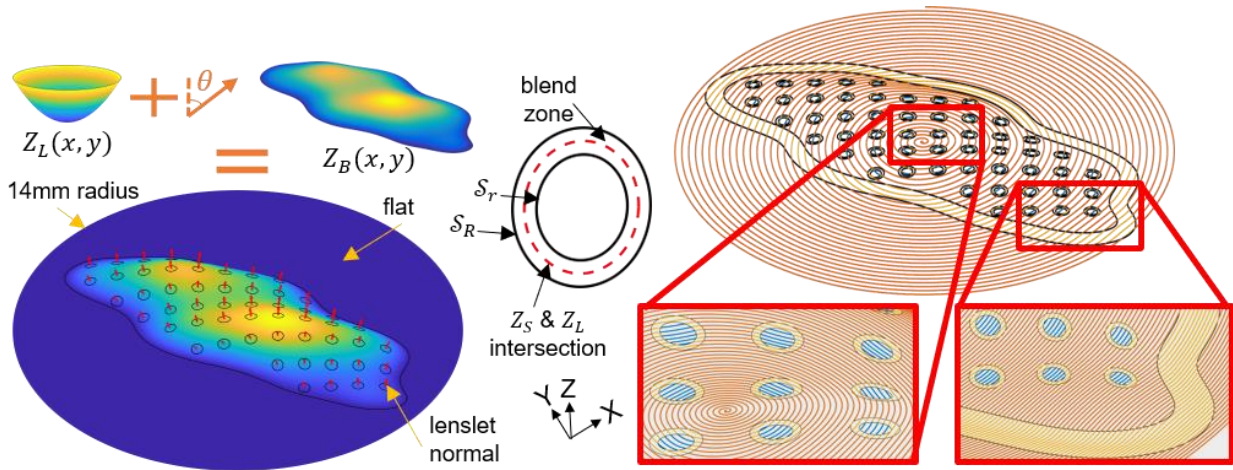


FIGURE 12. Freeform microlens array. Lenslet normals shown in red (left). Sample freeform blend zone (center). Overall C^3 continuous spiral toolpath with increased radial stepovers for clarity (right). Lenslet and base surface sags are $46\mu\text{m}$ & $633\mu\text{m}$, respectively. All lenslet apertures are freeform, unique, and have radii ranging from $256\mu\text{m}$ to $310\mu\text{m}$.

Figure 13 shows the final brass surface. The overall blended surface exhibits maximum radial and azimuthal slopes of 38.8° and 22.8° , respectively. Therefore, a diamond tool with $200\mu\text{m}$ radius, 120° sweep, and 25° clearance angle was used. The lenslet surface roughness, measured with a Taylor Hobson CCI non-contact Optical Profiler, is shown in Figure 14.

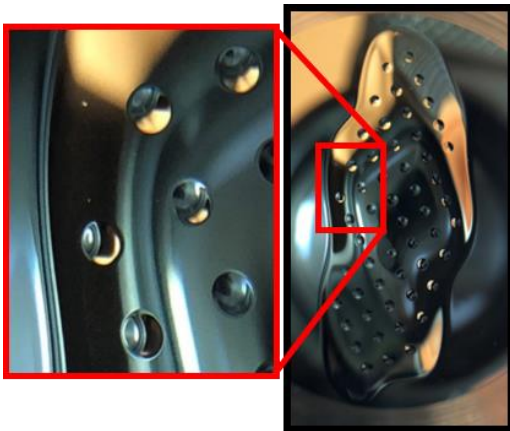


FIGURE 13. Final brass freeform microlens array with C^3 continuous freeform blends.

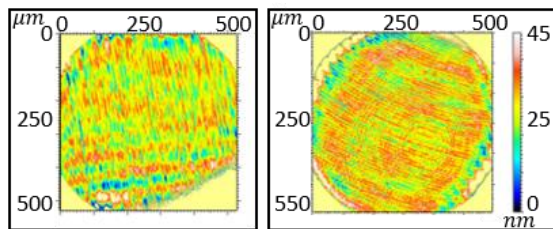


FIGURE 14. Surface roughness. Lenslet near edge: $5.71\text{nm } S_a$ (left). Lenslet near center: $5.43\text{nm } S_a$ (right).

CONCLUSION

This paper briefly introduces the most general surface transition techniques to date for explicitly described freeform optical surfaces that are defined inside arbitrary boundaries. By combining a robust boundary definition, freeform surface extrapolation, and freeform surface blending, key relationships used in the toolpath generation process were described and applied. Designed with a minimum number of assumptions, the proposed surface transitions are described by polynomial curves that emanate radially outside the boundary of an optical surface. It was shown that increasing the degree of the polynomial curves increases overall surface continuity at the expense of increased sag values.

ACKNOWLEDGEMENTS

We would like to thank everyone Ametek Precitech for their support on this research. In particular, we would like to thank Jeff Roblee for his collaboration and guidance.

REFERENCES

- [1] C. de Boor, A Practical Guide to Splines, New York: 1978.
- [2] R. Hamming, Numerical Methods for Scientists and Engineers, New York: McGraw-Hill, Inc., 1973.