Mirror Surface Reconstruction from a Single Image: Supplementary Material

Anonymous CVPR submission

Paper ID 903

1. Derivation of the Degree 2 Polynomial Equation

In the paper, we have shown that the starting depth for the initial value problem that defines the shape of the surface could be obtained by solving a degree 2 polynomial equation. This polynomial equation was derived by studying the order of integration of two partial differential equations. Here, we give more details on this derivation.

Recall that our differential geometry formulation allowed us to derive the pair of first-order partial differential equations that govern the shape of the mirror surface

$$\begin{aligned}\frac{\partial s}{\partial x} &= \frac{-s\, n_x(x,y,s)}{\langle \mathbf{n}(x,y,s), \mathbf{v}(x,y)\rangle} \\ \frac{\partial s}{\partial y} &= \frac{-s\, n_y(x,y,s)}{\langle \mathbf{n}(x,y,s), \mathbf{v}(x,y)\rangle}\end{aligned} \tag{1}$$

where the surface normal $\mathbf{n}$ and the visual ray $\mathbf{v}$ are explicitly expressed as functions of the image coordinates (x, y). More specifically, $\mathbf{v}(x,y) = (x,y,1)^\top$, and $\mathbf{n}$ is the normal to the mirror surface at point $\mathbf{p}(x,y) = s(x,y)\mathbf{v}(x,y)$. The normal is computed as the bisector of the angle between $-\mathbf{v}(x,y)$ and the ray from $\mathbf{p}(x,y)$ to the known corresponding point $\mathbf{m}(x,y)$ on the reference plane. Thus, the normal $\mathbf{n}(x,y,s)$ is entirely determined by x, y, s and $\mathbf{m}(x,y)$.

Assuming that the mirror surface is C^2 continuous, we can write the constraint

$$\frac{\partial^2 s}{\partial x \partial y} = \frac{\partial^2 s}{\partial y \partial x}. \tag{2}$$

Substituting Eq. (1) into Eq. (2) lets us derive a polynomial equation of the form

$$(As^2 + Bs + C) + (as + b)\sqrt{ps^2 + qs + t} = 0, \tag{3}$$

where A, B, C, a, b, p, q and t are coefficients whose value depends on the observed data, but not on the unknown parameter s. *The complete expression for the polynomial is provided in the file* '*originalPolynomial.png*'. To solve for s, we multiply both sides of Eq. (3) by $(As^2 + Bs + C) - (as + b)\sqrt{ps^2 + qs + t}$, which yields the degree 4 polynomial equation

$$(A^2 - a^2 p)s^4 + (2AB - 2abp - a^2 q)s^3 + (B^2 + 2AC - b^2 p - a^2 t - 2abq)s^2 + (2BC - 2abt - b^2 q)s + C^2 - b^2 t = 0. \tag{4}$$

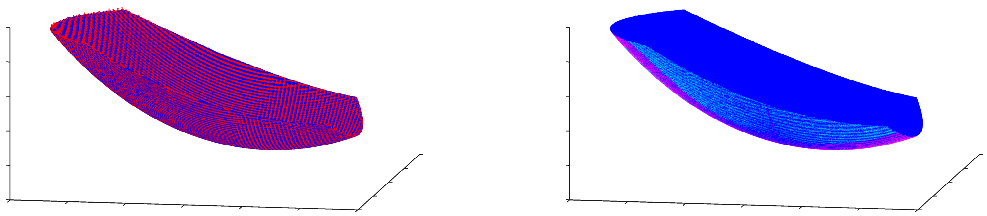
However, the algebraic expression of the coefficients A, B, C, a, b, p, q and t is such that the degree 4 and degree 3 terms in the polynomial (4) vanish. *This can be verified using the Matlab script 'Degree2PolynomialDerivation.m'*. Therefore, Eq. (4) reduces to the degree 2 polynomial equation

$$(B^2 + 2AC - b^2 p - a^2 t - 2abq)s^2 + (2BC - 2abt - b^2 q)s + C^2 - b^2 t = 0. \tag{5}$$

This lets us obtain s analytically as

$$s = \frac{-(2BC - 2abt - b^2) \pm \sqrt{(2BC - 2abt - b^2)^2 - 4(B^2 + 2AC - b^2 p - a^2 t - 2abq)(C^2 - b^2 t)}}{2(B^2 + 2AC - b^2 p - a^2 t - 2abq)}. \tag{6}$$

Although this suggests that there exist two valid solution for s, for generic surfaces, only one of them satisfies Eq.(3). This is due to the fact that our strategy to remove the square root in Eq. (3) introduces an additional, invalid solution.



Sphere Polynomials Sphere Integration

Figure 1. **Shapes obtained by solving polynomial equations and by integration:** *Red dots* denote the surface reconstructed by solving polynomial equations, *Cyan dots* the surface points obtained by solving the PDEs in order A, and *Magenta dots* in *order B*. *Blue dots* denote the ground truth.

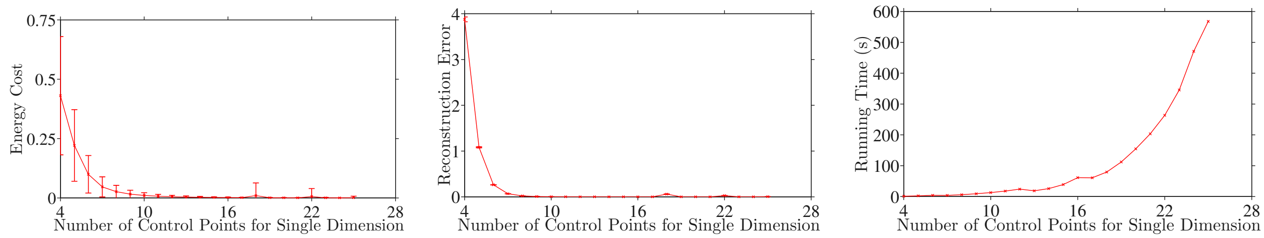
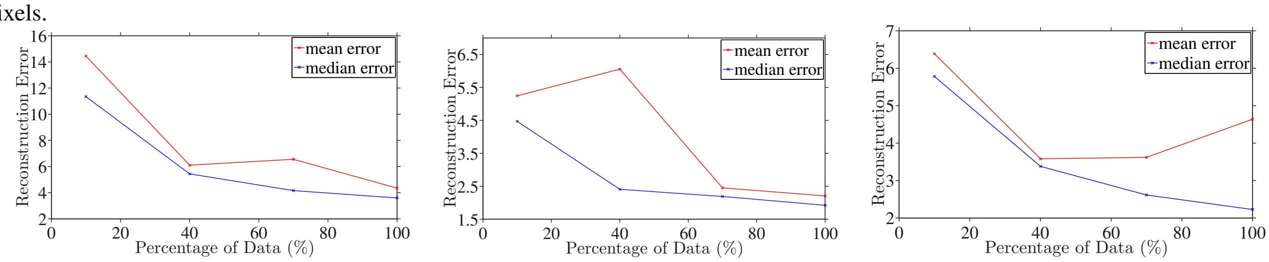


Figure 2. **Influence of control points for sphere reconstruction:** From left to right, we show the influence of the number of control points of the UCBS used to approximate the sphere on the mean pixel objective function value, the reconstruction error and the runtime. To really estimate the approximation error, these results were obtained with noise-free data. The standard deviations were computed over the image pixels.



UCBS Ellipsoid Sphere

Figure 3. **Reconstruction errors as a function of the density of noisy data by solving polynomials**: We show the average reconstruction error and median error for all the used pixels over 50 runs with different random noise.

2. Additional Experiments

Fig. 1 shows the reconstruction results for sphere by solving polynomial equations and by integration. Fig. 2 shows the influence of the number of control points of the UCBS used to approximate the sphere on the mean pixel objective function value, the reconstruction error and the runtime. Fig. 3 shows the average reconstruction error (*mean error*) and the median reconstruction error (*median error*) as a function of the density of noisy correspondences for the UCBS, the ellipsoid and the sphere when the shape is obtained by solving polynomials. The large difference between the mean and the median indicates that, while some points have large errors, at least half of them can still be reconstructed accurately.

3. Videos

We provide videos to better show the reconstructions obtained with our 3 different methods. '*Inte.mp4' depicts the reconstructed mirror surface obtained by integration both in *orderA* and *orderB*; '*Poly.mp4' shows the reconstructed mirror surface obtained by solving polynomials; '*Opt.mp4' shows, in a first phase, the evolution of the surface during optimization, and in a second phase, a view of the reconstruction from different angles. When available, ground-truth is also shown in the video for comparison.