Depth from Focus using Windowed Linear Least Squares Regressions

Corentin Cou, Gaël Guennebaud

Depth from Focus using Windowed Linear Least Squares Regressions

Corentin Cou[1*] and Gaël Guennebaud[2*]

[1]CNRS, France.
[2]Inria, Bordeaux University, France.

*Corresponding author(s). E-mail(s): corentin.cou@inria.fr; gael.guennebaud@inria.fr;

Abstract

We present a novel depth from focus technique. Following prior work, our pipeline starts with a focal stack and an estimation of the amount of defocus as given by, for instance, the Ring Difference Filter. To improve robustness to outliers while avoiding to rely on costly non-linear optimizations, we propose an original scheme that linearly scans the profile over a fixed size window, searching for the best peak within each window using a linearized least-squares Laplace regression. As a post-process, depth estimates with low confidence are reconstructed though an adaptive Moving Least Squares filter. We show how to objectively evaluate the performance of our approach by generating synthetic focal stacks from which the reconstructed depth maps can be compared to ground truth. Our results show that our method achieves higher accuracy than previous non-linear Laplace regression technique, while being orders of magnitude faster.

Keywords: Depth map acquisition, Depth from focus.

1 Introduction

Capturing the 3D structure of real world scenes is a long standing problem in the computer vision and computer graphics community. A wide range of techniques have been proposed so far such as photogrammetry, laser scan or structured lights. When targeting micro-scale reliefs, however, such triangulation based optical systems become impracticable because the depth-of-field decreases quadratically with the field of view. For instance, when targeting a resolution of one pixel per micrometer with a classical digital camera, the typical depth of field will range between 5 and 50 micrometers only.

In this context, depth-from-focus methods become particularly appealing. Those techniques strive to recover 2.5D depth information from a focal stack by estimating, for each pixel, the depth of the focus plane. Most depth from focus methods start by applying a focus measure on the input focal stack, and then estimate the depth by seeking, for each pixel, a robust peak of this focus measure profile along the z direction. Most techniques manage to recover a discrete depth information with reasonably high accuracy, and the relative depth accuracy of the reconstruction is thus expected to increase as the field of view decrease. However, their performance at recovering continuous depth information remains unclear yet.

In this paper we propose a novel peak estimation method that strives to address this shortcoming. This paper makes the following four key contributions:

1. *Sliding window.* To improve robustness to outliers while avoiding costly non-linear optimizations, we propose an original scheme that linearly scans the profile over a fixed size window, searching for the best peak within each window (Section 4.1).
2. *Linear least-Squares Laplace regression.* Within each window, we identify the peak by fitting a Laplace distribution over the data. We greatly improve the computation cost by proposing a linearization scheme (Section 4.2).
3. *Adaptive smoothing.* As a post-process, we present an adaptive and feature preserving smoothing scheme based on a per pixel confidence value and Moving-Least-Squares like regressions (Section 5).
4. *Quantitative evaluation.* We objectively evaluate the performance of our method by generating synthetic focal stacks from which the reconstructed depth maps can be compared to ground truth (Section 6).

2 Related Works

Depth from defocus. Exploiting depth of field information to reconstruct depth information was first proposed by Pentland [1]. Their *Depth from defocus* methods estimate the depth from the amount of defocus estimated from a pair of images taken at different focal lengths, and many variants of this approach have been proposed so far [2]. Using a *coded aperture*, Levin et al. [3] even showed how to get rid of the back and front ambiguity when working on a single image. However, since it is extremely difficult to estimate the amount of defocus independently of the texture frequency of the scene itself, those *Depth from defocus* methods can only recover very coarse depth information that can only be used for simple tasks such as autofocus, foreground/background segmentation and the likes, but unlikely for 3D scanning.

Depth from focal stack. A focal stack is a sequence of images taken for different focal planes. It can be obtained by either moving the camera along a translation unit, or by changing the focal length of the optics. Usually, each point of the scene is in focus in one and only one slice such that an *all-in-focus* image can be reconstructed by peaking the sharpest pixel along each view direction [4]. A discrete depth map can be jointly obtained by assigning to each selected pixel the depth of its respective focal plane [5]. Depth from focal stack can be then used alone to determine depth, or in conjunction with Depth from Defocus and other methods [6].

Focus measure. In such *depth from focus* approaches, a central ingredient is the so called *focus measure*, whose goal is to estimate the level of sharpness of every pixel of the stack. Dozens of heuristics have been proposed, either based on differential operators, contrast estimator, or frequency analysis [7]. According to Pertuz et al [8]'s study, Laplacian-based operators turned out to be the best performing. This is confirmed by more recent works improving upon classical Laplacian operators such as the Ring Difference Filter (RDF) [9]. Other methods tried to learn optimal combinations of different measures [10, 11].

Depth estimation. Simply taking the extrema of the focus measure to recover depth information [5] is fast but subject to high noise and limited to discrete depth values. Higher accuracy and robustness is achieved through fitting methods using either local Gaussian [12] or global Laplacian [11] distributions. Some recent work have also experimented with neural networks for this task [13], but they face the hunger for data for the learning stage. Nonetheless, the raw output of such estimators still have to be post-processed to reduce the noise and complete incorrect parts, for instance using cost-aggregation techniques [9] or cost-volume methods based on a reliability map [11]. Javidnia et al. [14] globally optimize both the depth and in-focus color of each pixel by minimizing the difference between the input reference stack and a procedurally generated stack from the current depth map.

3 Overview of the pipeline

As most depth-from-focal-stack methods, our approach relies on the three main steps identified in the previous section: focus measure, peak extraction, and post-processing. Our pipeline is depicted in Figure 1. More precisely, we start from a stack $\mathcal{I}$ of n images where each pixel is denoted $\mathcal{I}^k_{i,j}$, with i, j the pixel indices within the slice number $k \in [1, n]$, each slice corresponding to a focal plane depth z_k. We assume that the focal planes are uniformly sampled, with a constant

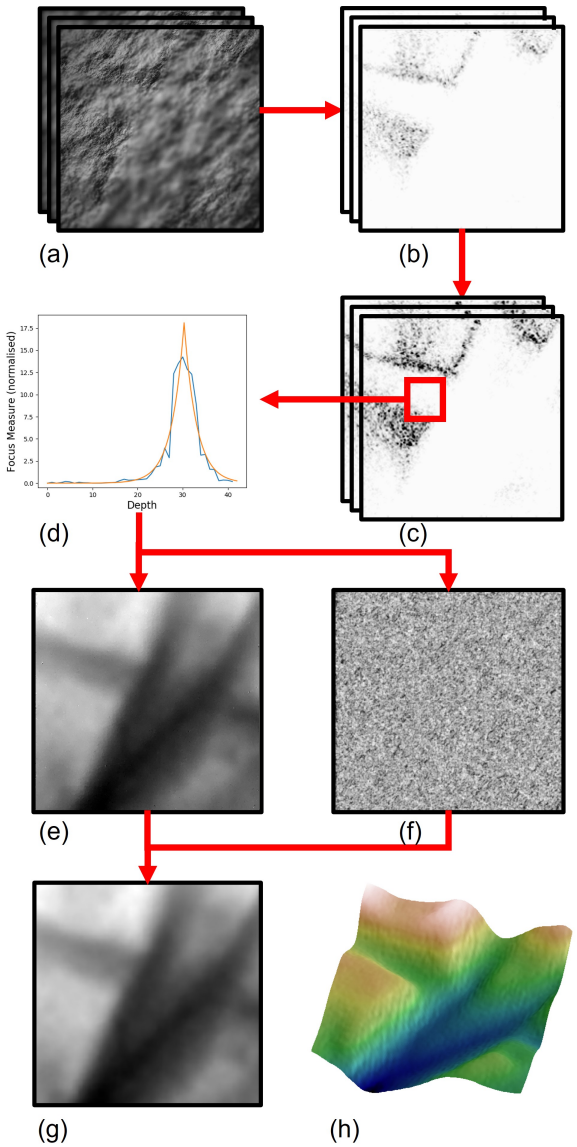

Fig. 1 Our depth from focus pipeline. Starting from the acquired focal stack $\mathcal{I}$ (a), we compute a Focus Measure map $\mathcal{F}$ (b) and a low-pass version $\tilde{\mathcal{F}}$ (c) through Gaussian smoothing. For each pixel, we perform our Peak Estimation procedure (d), yielding a depth map $\mathcal{D}$ (e) and a reliability map $\mathcal{R}$ (f). Finally, we apply an adaptive MLS filter from these two maps to obtain our final depth map (g) giving a 3D profile (h).

sampling interval $\delta_z = z_{k+1} - z_k$. Each slice of the stack are also assumed to be pre-aligned such that for each spatial indices i, j, the 1D profile $\{\mathcal{I}^k_{i,j}\}_k$ corresponds to the same point of the scene.

Then we apply a Focus-Measure (FM) independently on each pixel of the stack yielding a FM stack $\mathcal{F}$ (Fig. 1b). The rest of our pipeline can work with any FM producing sharp profiles resembling to a Laplace distribution (Fig. 1d). This is the case of all Laplacian-like differential operators, including the recent composite FM [11]. From our experiments, we found the 5×5 Ring Difference Filter (RDF) [9] to be the best performing, especially in areas of low contrast, while being simple and fast. To reduce the sensitivity to spatial noise, we apply a 2D Gaussian filter on each slice of the FM stack, yielding a smoothed version denoted $\tilde{\mathcal{F}}$ (Fig. 1c).

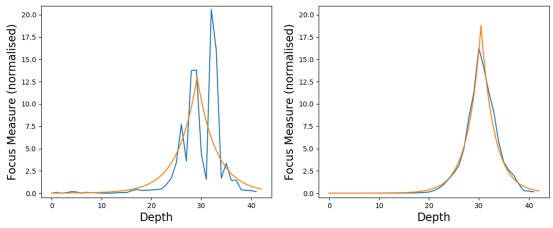

Fig. 2 Demonstration of the importance of the Gaussian filter. Left: curve of FM of one pixel as a function of depth z, on an FM where the Gaussian filter is not applied. Right: curve of FM of the same pixel as a function of depth z, on an FM where the Gaussian filter has a width of 20.

The crucial step then consists in extracting from $\tilde{\mathcal{F}}$ the depth $\mathcal{D}_{i,j}$ of each pixel (i, j). In our approach, this step is carried out independently for each pixel (i, j), so that it boils down to a continuous peak estimation problem from the discrete 1D signal $f_k = \tilde{\mathcal{F}}^k_{i,j}$ (Fig. 1d). This is the main part of our contribution and it is detailed in Section 4. In addition to the depth map $\mathcal{D}$ (Fig. 1e), in this step we jointly compute a reliability map $\mathcal{R}$ (Fig. 1f) that is used to drive an adaptive denoising procedure detailed in Section 5 (Fig. 1g).

4 Peak estimation using Linearized Sliding-window

This section focuses on computing, for each pixel (i, j), its depth value z from its 1D discrete focus measure profile f_k. In the following, it is assumed that z can be recovered as the location of the maximum value of some continuous reconstruction $\bar{f}$ of f_k, that is: $z = \text{argmax}_x\, \bar{f}(x)$. Sakurikar et al. [11] showed that the Laplace distribution

$$L_{\mu,b}(x) = \frac{1}{2b} e^{-\frac{\|x-\mu\|}{b}} \qquad (1)$$

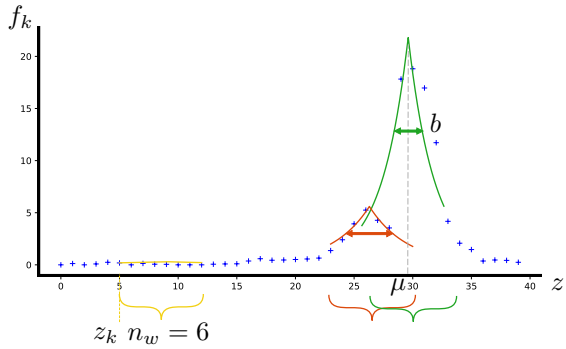

Fig. 3 Illustration of our sliding window strategy. The FM profile is shown as blue crosses and three position of the window are depicted in yellow, red, and green respectively. The window size n_w is constant, and each window starts at a position z_k.

was a good proxy model to reconstruct $\bar{f}$ for a variety of focus measures. They compute the mean μ and average absolute deviation (AAD) b parameters through the global minimization of the following non-linear energy:

$$E_0 = \sum_k \left(L_{\mu,b}(z_k) - f'_k\right)^2 , \qquad (2)$$

where f' is the normalized version of the f_k so that it integrates to 1. Then, we directly get the sought for depth as $z = \mu$, and the AAD value b, or rather its inverse b^{-1}, gives us an estimation of its reliability: the finer and higher the peak of the distribution, the sharper and more reliable it is.

This non-linear regression procedure exhibits two major issues. First, it involves an iterative and computationally expensive minimization. Second, since the regression is carried out on the whole signal in a simple least-squares sense, it is highly sensitive to outliers and noise far away the expected peak. Here we are referring to outliers and noise that might be produced by the focus measure, and not directly to outliers that would be present in the images themselves. For instance, in the presence of strongly visible *bokeh* artefacts, the focus measure signal f_k might exhibit two peaks leading to an estimated depth located in-between the two peaks.

We alleviate both issues through linearization (Sec. 4.2) and localized regressions (Sec. 4.1) respectively.

4.1 Sliding-window estimation

To deal with outliers, we localize the Laplace regression through a basic sliding window strategy described in Algorithm 4.1, and illustrated in Figure 3. The idea is to perform a full peak estimation (e.g., through a Laplace fitting) independently for each sub range $\{f_k, \ldots, f_{k+n_w-1}\}$ of the stack, with n_w the width of the window. Depth estimations falling outside the current window are rejected, and finally we retain the one associated with the best reliability.

Algorithm 1 Sliding window algorithm

```
r ← 0                                              ▷ reliability
for k ∈ [1, n − n_w + 1] do
    μ, b ← Laplace_regression(f[k, k + n_w − 1])
    if b^-1 > r and z_k < μ < z_k + n_w δ_z then
        r ← b^-1
        z ← μ
    end if
end for
return z, r
```

4.2 Linearization as a Triangle function

Running a non-linear regression on each sub-window would be prohibitively expensive. To speed up computation, we thus linearize the energy E_0 by taking the logarithm of both the focus measure f and the Laplace distribution. However, doing so directly on the Laplace distribution yields a triangle function of the form: $\log(\frac{1}{2b}) - \frac{\|x-\mu\|}{b}$, which unfortunately still exhibit a non linear relationship between the slopes and constant terms. To address this issue, we relax the somewhat arbitrary constraint of L being strictly a probabilistic distribution by making its amplitude $\frac{1}{2b}$ in Eq. 1 a free parameter, say a, yielding a triangle function with three degrees of freedom of the form: $\log(a) - \frac{\|x-\mu\|}{b}$. This equation is advantageously rewritten as two piecewise linear functions:

$$T_{\alpha,\beta,\gamma}(x) = \begin{cases} \alpha \cdot x + \beta & \text{if } x < \mu \\ -\alpha \cdot x + \gamma & \text{if } x \geq \mu \end{cases} , \qquad (3)$$

where the position μ of the line intersection and AAD parameter of the initial Laplace distribution are recovered by:

$$\mu = \frac{\gamma - \beta}{2\alpha} \quad , \quad b = \frac{\gamma + \beta}{2} \, .$$

To efficiently deal with the absolute value, or piecewise nature of T, we first enforce the window size n_z to be an even number, and then assume that μ is located somewhere in-between the two slices surrounding the window center $z_c = (z_{k+n_w/2-1} + z_{k+n_w/2})/2$. This way we can partition the current window's samples into two groups, $W_1 = [k, k + n_w/2 - 1]$ and $W_2 = [k + n_w/2, k + n_w - 1]$, each group being associated to one side of the triangle.

The three parameters α, β, and γ are then easily obtained by minimizing the following quadratic least-squares energy:

$$E_1 = \sum_{k \in W_1} \left(\alpha z_k + \beta - \log(f_k)\right)^2 \tag{4}$$

$$+ \sum_{k \in W_2} \left(-\alpha z_k + \gamma - \log(f_k)\right)^2 \, . \tag{5}$$

Let us emphasize that thanks to the sliding window strategy, our *a priori* partitioning with respect to the center of the window is not really an issue because all possible locations of this cut will be tested anyways. Nonetheless, nothing in E_1 prevents the two lines to cross at a position μ within the two slices surrounding z_c. Whereas this constraint could be achieved through a pair of linear inequalities, this would not be desirable as this would simply snap the eccentric estimations on discrete values. Instead, we rather tighten the acceptance test in Algorithm 4.1 to accept only the candidate μ within the sub range centred on the window and of width $3\delta_z$. We chose an acceptance width of $3\delta_z$ to be tolerant to slightly eccentric estimates to prevent the risk of entirely missing a peak.

In the rest of this paper, we will refer to this method as the *Triangle* method. Let us note that this linearization would not be possible without our sliding window strategy, and inversely, our sliding window strategy would be prohibitively expensive without our linearization.

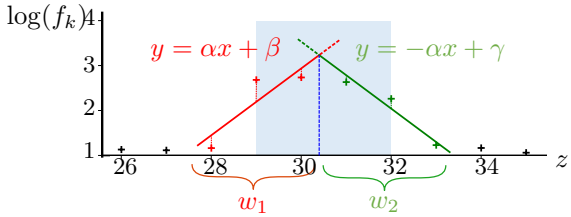

Fig. 4 Illustration of the *Triangle* method. In log space, the current window [28, 31] is partitioned within two parts. Each part is approximated by a piece-wise linear function having opposite slopes.

5 Post-correction using MLS

From the depth map $\mathcal{D}$ and reliability map $\mathcal{R}$, we apply an adaptive low pass filter in order to both denoise the depth map and fix low reliability values from its surrounding. To preserve the depth map features, we designed a filter inspired by the Moving Least Squares filter [15]. For each pixel (i, j), the idea consists in approximating its neighborhood by a bivariate polynomial g, and replace its depth value by the value of g at the pixel location $\mathbf{q}_{i,j}$. We take into account the reliability map $\mathcal{R}$ by introducing it as additional weights within the regression. More formally, the polynomial g is obtained as the minimization of the following quadratic energy:

$$\sum_{x,y \in N_{i,j}} \mathcal{R}_{x,y} \theta(\|\mathbf{q} - \mathbf{q}_{x,y}\|) \left(g(\mathbf{q}_{x,y}) - \mathcal{D}_{x,y}\right)^2 \, , \tag{6}$$

where θ is a compactly supported weighting function of radius h, and $N_{i,j}$ denotes the neighborhood of the pixel (i, j) within a distance h. In our implementation, we used a quadratic polynomials g, and for the weight function $\theta(t) = \left(1 - \frac{t^2}{h^2}\right)^2$ which is a classic in the MLS reconstruction literature [16], but in our discrete filtering context, any Gaussian-like function would perform equally well. Compared to a simple weighted average, the use of a degree two polynomial allows to much better preserve the features of the data. In the presence of uneven sampling or holes as implied here by the reliability map, such high order polynomials also allow to much better preserve the slopes, whereas a simple weighted average would be strongly biased toward the more reliable and more numerous depth values.

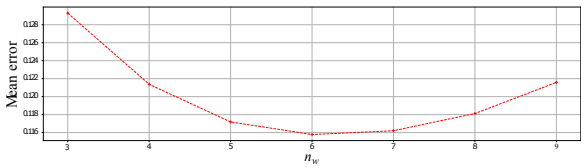

Fig. 5 Average absolute error over 20 synthetic stacks as a function of the window size n_w.

6 Experiments and Results

We implemented a prototype of our pipeline within Matlab. For the spatial smoothing pass of the focus measures, we used a Gaussian filter size of 7 pixels, and a MLS filter size of 5 pixels for the post processing pass. The sliding window size n_w should be adapted to the ratio between the sampling interval δ_z, and the depth-of-focus of the actual setup. For an ideal setup where the twos are roughly equal, we found $n_w = 6$ to be the best performing according to our experiments (see Figure 5). With super-sampling, however, n_w should be enlarged accordingly to ensure the window to be large enough to fully contain the typical Laplace-shaped peaks. It might be possible to automatically estimate the ideal window size of a stack by analyzing a few noise-free pixel profiles, but we left this as future work.

6.1 Evaluation on synthetic stacks

To objectively evaluate the performance of our different contributions, we procedurally generated image stacks from several reference depth and texture maps. To this end, we implemented a simple splatting approach spreading the contribution of each pixel of the input depth and texture maps to each image of the stack as a Gaussian splat approximating the true Airy disk. The radius of the splat is adjusted with respect to the size c of the circle of confusion computed as:

$$c = A \cdot \frac{\|S_2 - S_1\|}{S_2} \cdot \frac{f}{\|S_1 - f\|},$$

with A the aperture, f the focal length, S_1 the distance to the focal plane, and S_2 the distance to the current scene point.

With this procedure, we generated 80 stacks from 4 base depth maps, each being perturbed with 5 levels of low frequency noise mimicking surface details. Some combinations are shown in Figure 7, first column. Those 20 depth-maps are then combined with 4 different grey-level textures shown in Figure 6. Each stack is formed by images of 400×400 pixels.

Table 1 Statistics of absolute error prior to MLS post-correction. The 4th to 6th columns report the worst error for the given percentiles.

Method	Mean	Std-dev	25%	50%	75%	Time
3 pts	0.360	0.354	0.135	0.288	0.497	0.6s
Laplace	0.219	0.256	0.075	0.163	0.292	90.1s
Triangle	0.196	0.257	0.065	0.141	0.259	1.5s

We compare our *Triangle* method against two other peak estimations borrowed from the literature. The first one, called *Laplace*, is the global non-linear regression of a Laplace distribution coming from the the method of Sakurikar et al. [11] and that we recalled in Section 4. Implementation-wise, we used the `lsqnonlin` Matlab's function parametrized to use a Levenberg-Marquardt algorithm with low tolerance values (i.e, `TolX=TolFun=1e-4`) to make the multiple non-linear solves as fast as possible. The second one, called *3 pts*, comes from the three-points method of Billiot et al. [12]. Their method estimates the depth as the location μ of the peak of a generalized Gaussian $a \cdot e^{-\frac{x-\mu}{2\sigma}}$ whose three parameters are computed such that it passes through the sample having the maximal focus measure, and its two neighbors. Likewise, we compute the reliability as the height of the respective normalized Gaussian. We also added results from the RDF method [9] using the implementation provided by the authors, and the full pipeline of Sakurikar et al. [11] which is similar to the "Laplace" pipeline described above but using their *composite* focus measure and a bilateral filter instead of our MLS pass.

Tables 1, and 2 show global statistics for the 80 stacks of the *Mean Absolute Error* (MAE) computed against the reference depth maps for each of the three methods. The reported MAE is normalized such that it should be interpreted as if the distance between two focal planes δ_z is 1.

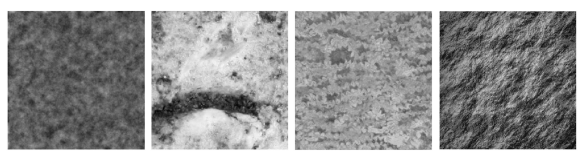

Fig. 6 The four textures used to generate our 80 stacks.

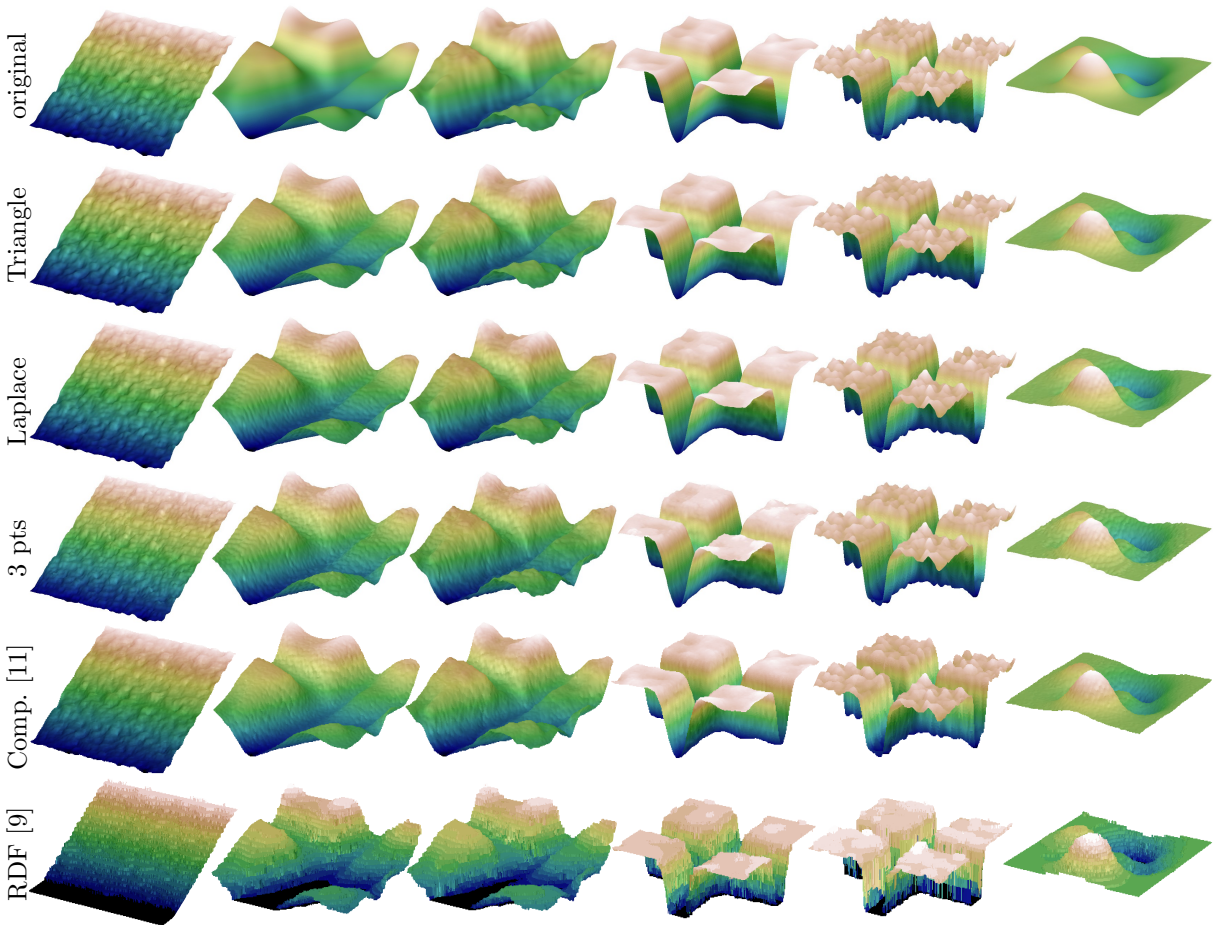

Fig. 7 Qualitative comparison of 3-pts, Laplacian, and Triangle methods on some generated image stacks.

Table 2 Statistics of the absolute error and running time for the full pipelines (i.e., including MLS post-correction).

Method	Mean	Std-dev	25%	50%	75%	Time
3 pts	0.237	0.223	0.089	0.189	0.331	17.7s
Laplace	0.171	0.181	0.060	0.130	0.232	107.2s
Triangle	0.159	0.184	0.054	0.117	0.213	18.6s
RDF[9]	0.827	0.921	0.246	0.534	1.067	14.8s
Comp.[11]	0.236	0.414	0.072	0.157	0.288	94.5s

One can notice that the RDF pipeline produces the worst statistics: this is expected as it produces discrete depth values whereas all others produces continuous ones. Not surprisingly, Sakurikar et al. [11] pipeline exhibits statistics in par with our *Laplace* pipeline.

Now focusing on our three pipeline variants, we can see that the MLS post-correction step provides an average of a 20% to 30% reduction of the MAE. Its visual effect is depicted in Figure 8. Furthermore, all precision-related indicators lead to the same hierarchy, with the *3-pts* method being significantly less accurate than the other two peak-estimators, and our *Triangle* method being the most accurate. Overall, compared to the non-linear *Laplace* method, our *Triangle* method provides a 10% reduction of the MAE while being more than $\times 60$ times faster, and only three times more expensive than the simplest *3-pts* method.

Those 10% gains, however, does not tell everything about the quality improvement in general. In Figure 9 we report the repartition of the computed depth values for each of the three methods, assuming $\delta_z = 1$. Those histograms reveal a clear bias toward a discrete set of predetermined values corresponding to the discrete input sample depths for the *3-pts* method, and their middle for the other twos. Our linearized *Triangle* method exhibits a much lower bias though. In practice, this bias produces depth maps with *plateau*-like artefacts that can be observed in the results of

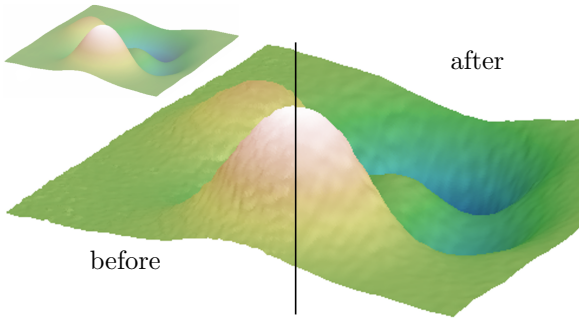

Fig. 8 Reconstruction of a very smooth initial 3D profile (top-left inset), before and after our MLS adaptive post-correction.

Figure 7. Our *Triangle* result thus appears significantly less noisy even-though the MAE differs by 10% only with *Laplace*. From those observations, one might though about correcting this bias by applying to the decimal part of the estimated depths a precomputed transfert function. As show in this figure (right column), this systematic bias is well captured by a cubic polynomial function constrained to pass trough $(0, 0)$ and $(1, 1)$. Applying this post-correction to the raw depth outputs (i.e., prior to MLS smoothing) permits to reduce the MAE of the *3 pts* and *Laplace* methods to 0.322 and 0.211 respectively which are both still clearly larger than the MAE of our *Triangle* method. This post-correction has no effect to our *Triangle* method as the magnitude of this bias effect is already very low, and lower than the amount of noise.

6.2 Depth from focus at micrometer scale

To evaluate our approach in a real-world context, we strived to acquire the relief of prehistoric engraved stones exhibiting groove widths ranging from 50 to 100 µm. The precision of the depth-from-focus pipeline is greatly influenced by the design of the acquisition setup [17]. To achieve a $\times 2$ magnification while minimizing the depth of field, we stacked together a 100 mm lens with an inverted 50 mm lense playing the role of a close-up lens. To minimize the depth of field, both lenses are fully opened (respectively 1:2.8 and 1:1.4). With this setup we found that a step size δ_z of 20 µm to be a good tradeoff. Automatic staking is achieved through a motorized rail, while the

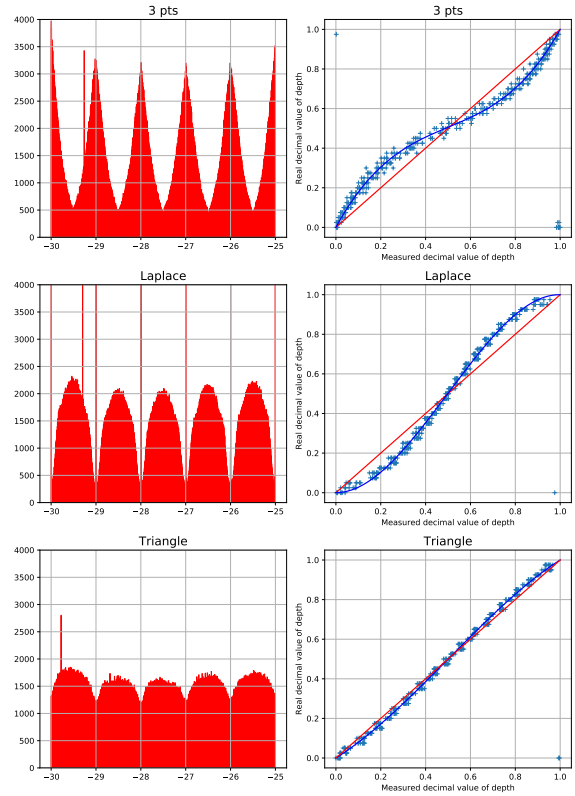

Fig. 9 For each of the three methods, the left histograms show the distributions of the computed depth values for the 80 depth maps $\mathcal{D}$, and the right scatter plots show the ground truth decimal depth values as a function of the decimal of the computed depth values. Those plots also include least-squares cubic approximations of this function (blue curve), and the ideal identity relation in red.

resulting images are aligned with the *Enfuse* software prior to depth estimation with our pipeline. Figure 10 shows one result at the crossing of three strokes. Despite the depth of the strokes (about 40 µm) being of same order as the sampling depth δ_z, we can clearly see the stroke order.

As we can see in Fig. 5, the engravings are visible and visually separable by their depths. In addition, we notice the different grooves in the same engravings, made by the fact that the tool used has been passed several times. Our method therefore makes it possible to produce a sufficiently robust 3D profile on macroscopic images to observe the details of engravings that are impossible to digitize with a laser scan or by photogrammetry. Other practical results can be found in this paper [18], where our method has been used to accurately measure the absolute thickness of

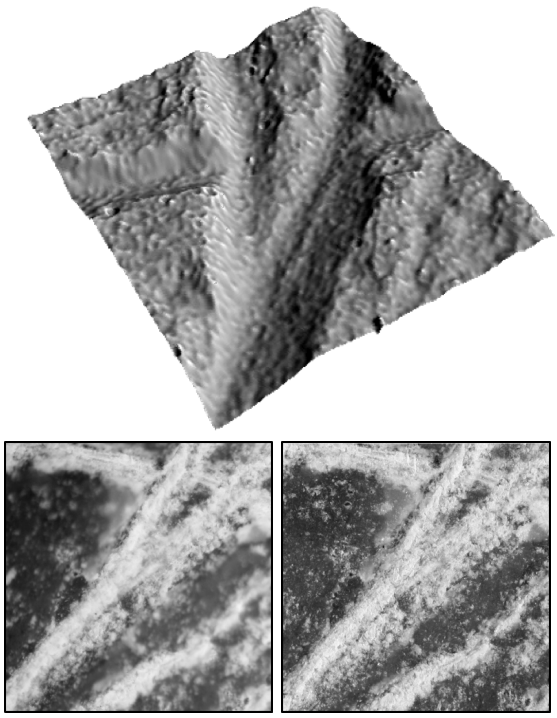

Fig. 10 Results get on a real-word stack from an engraved stone. Top: 3D profile of the acquired engraving area. Bottom left: blurry image obtained on the image stack. Bottom right: reconstituted clear image of the engraving.

thin glaze of ancient ceramics in a non destructive manner.

7 Discussions and Conclusion

In this paper we showed how a classical depth-from-focus reconstruction pipeline can be greatly improved, both in terms of precision, robustness, and computation speed, using a localized and fully linear regression scheme. Our naive MatLab implementation, processing one pixel a time, achieves about 1.5s per image stack, and we believe a compiled and vectorized implementation could cut down the computation time by one or two orders of magnitude. Moreover, as our approach boils down to the regression of simple linear polynomials, it would be interesting to investigate the use of more advanced regression schemes such as total-least-squares and statistically robust methods that might be implemented at a reasonable computation time overhead. One of the main limitation of our pipeline is that the underlying depth map is implicitly assumed to be continuous. A related limitation is that our pipeline processes each pixel independently. As future work, it would thus be interesting to extend our peak estimator with a more global *view* as well as with some mechanisms to handle discontinuities produced by occlusions.

Declarations

Data availability

The source code of our method and the stacks used to conduct our evaluations and to produce Figure 10 are publicly available on this github repository: https://github.com/corentincou/Depth-from-Focus-using-Windowed-Linear-Least-Squares-Regressions. The source code used to generate the procedural stacks and to compare the different methods can be obtained from the authors on demand.

References

[1] Pentland, A.P.: Depth of scene from depth of field. Technical report, SRI INTERNATIONAL MENLO PARK CA (1982)

[2] Kulkarni, J., Chetty, M.S.R.: Depth estimation from defocused images: A survey. Journal of Engineering and Applied Sciences (3), 331–335 (2019)

[3] Levin, A., Fergus, R., Durand, F., Freeman, W.T.: Image and depth from a conventional camera with a coded aperture. ACM transactions on graphics (TOG) **26**(3), 70 (2007)

[4] Pertuz, S., Puig, D., Garcia, M.A., Fusiello, A.: Generation of all-in-focus images by noise-robust selective fusion of limited depth-of-field images. IEEE Transaction on Image Processing **22**(3) (2013)

[5] Grossmann, P.: Depth from focus. Pattern recognition letters **5**(1), 63–69 (1987)

[6] Lee, J.Y., Park, R.-H.: Complex-valued disparity: Unified depth model of depth from stereo, depth from focus, and depth from defocus based on the light field gradient. IEEE transactions on pattern analysis and machine intelligence (2019)

[7] Sun, Y., Duthaler, S., Nelson, B.J.: Autofocusing in computer microscopy: selecting the optimal focus algorithm. Microscopy research and technique **65**(3), 139–149 (2004)

[8] Pertuz, S., Puig, D., Garcia, M.A.: Analysis of focus measure operators for shape-from-focus. Pattern Recognition **46**(5), 1415–1432 (2013). https://doi.org/10.1016/j.patcog.2012.11.011

[9] Surh, J., Jeon, H.-G., Park, Y., Im, S., Ha, H., So Kweon, I.: Noise robust depth from focus using a ring difference filter. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 6328–6337 (2017)

[10] Mahmood, M.T., Majid, A., Choi, T.-S.: Optimal depth estimation by combining focus measures using genetic programming. Information Sciences **181**(7), 1249–1263 (2011). https://doi.org/10.1016/j.ins.2010.11.039

[11] Sakurikar, P., Narayanan, P.J.: Composite focus measure for high quality depth maps. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV) (2017)

[12] Billiot, B., Cointault, F., Journaux, L., Simon, J.-C., Gouton, P.: 3d image acquisition system based on shape from focus technique. Sensors **13**(4), 5040–5053 (2013)

[13] Hazirbas, C., Soyer, S.G., Staab, M.C., Leal-Taixé, L., Cremers, D.: Deep depth from focus. In: Asian Conference on Computer Vision, pp. 525–541 (2018). Springer

[14] Javidnia, H., Corcoran, P.: Application of preconditioned alternating direction method of multipliers in depth from focal stack. Journal of Electronic Imaging **27**(2), 023019 (2018)

[15] Bose, N.K., Ahuja, N.A.: Superresolution and noise filtering using moving least squares. IEEE Transactions on Image Processing **15**(8), 2239–2248 (2006)

[16] Ledergerber, C., Guennebaud, G., Meyer, M., Bacher, M., Pfister, H.: Volume MLS ray casting. IEEE Transaction on Visualization and Computer Graphics (Proceedings of Visualization 2008) (2008)

[17] Blayvas, I., Kimmel, R., Rivlin, E.: Role of optics in the accuracy of depth-from-defocus systems. JOSA A **24**(4), 967–972 (2007)

[18] Cou, C., Ben Amara, A., Granier, X.: Non-invasive on-site method for thickness measurement of transparent ceramic glazes based on depth from focus. Archaeometry (2022)