HAL
open science

Parabolic-cylindrical moving least squares surfaces

Brett Ridel, Gael Guennebaud, Patrick Reuter, Xavier Granier

To cite this version:

Brett Ridel, Gael Guennebaud, Patrick Reuter, Xavier Granier. Parabolic-cylindrical moving least squares surfaces. Computers and Graphics, 2015, pp.7. 10.1016/j.cag.2015.05.006 . hal-01169572v1

HAL Id: hal-01169572

https://inria.hal.science/hal-01169572v1

Submitted on 29 Jun 2015 (v1), last revised 15 Jul 2015 (v2)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Parabolic-Cylindrical Moving Least Squares Surfaces

Brett Ridel[a,b,c], Gaël Guennebaud[a,b], Patrick Reuter[a,b], Xavier Granier[a,b,c]

[a] *Inria Bordeaux Sud-Ouest, France*
[b] *LaBRI (UMR 5800 CNRS, Université Bordeaux), France*
[c] *LP2N (UMR 5298 CNRS, Université Bordeaux, IOGS), France*

Abstract

Moving least squares (MLS) surface approximation is a popular tool for the processing and reconstruction of non-structured and noisy point clouds. This paper introduces a new variant improving the approximation quality when the underlying surface is assumed to be locally developable, which is often the case in point clouds coming from the acquisition of manufactured objects. Our approach follows Levin's classical MLS procedure: the point cloud is locally approximated by a bivariate quadratic polynomial height-field defined in a local tangent frame. The a priori developability knowledge is introduced by constraining the fitted polynomials to have a zero-Gaussian curvature leading to the actual fit of so-called *parabolic cylinders*. When the local developability assumption cannot be made unambiguously, our fitted *parabolic cylinders* seamlessly degenerate to linear approximations. We show that our novel MLS kernel reconstructs more locally-developable surfaces than previous MLS methods while being faithful to the data.

1. Motivation

Moving least squares (MLS) surfaces, also known as point set surfaces [14, 3] have proven to be useful for various point cloud processing tasks including surface reconstruction, resampling, smoothing, denoising and surface analysis. Given an unstructured and possibly noisy point cloud, MLS define a continuous surface by means of a continuously moving surface proxy approximating the input points nearby the considered evaluation point and some weight functions playing the role of a low-pass filter. By adjusting the support size of the weight functions, smoothness can be traded for proximity to the data [17].

Since the pioneering work of Alexa et al. [3] and Levin [14], countless variants have been developed to improve robustness [4, 11], or to take into account additional apriori knowledge such as local feature sizes [8], sharp edges [18, 22, 15], or anisotropic sampling [1].

This work is motivated by a different apriori knowledge of the MLS surface to be reconstructed, namely *developability*. By definition, developable surfaces, a subset of ruled surfaces, have zero-Gaussian curvature, or equivalently, one of its principal curvatures is null everywhere [9]. Less formally, developable surfaces can be unrolled onto a flat plane without tearing or stretching, and can thus be created from planar sheets of material without any torn or stretched deformations. This explains why an important subset of curved surfaces that are surrounding us are composed of developable pieces.

Developable surfaces have been mostly studied in the context of surface modeling [21, 23, 13, 25]. Several works aim to deform an existing mesh to make it developable and thus ease its fabrication [28, 16, 27, 29]. Eigensatz et al. [10] deform an existing mesh through arbitrary curvature manipulations. Tang and Chen [26] strive to fit an existing developable mesh to a sparse set of constraint points through isometric deformations. All these methods work on an existing mesh and cannot really be applied in the context of surface reconstruction from point clouds. In the context of reverse engineering, some work aims at recovering developable surfaces within point clouds. For instance, in the work of Peternell [20], such surfaces are seen as envelopes of a one-parameter family of tangent planes. Perriollat and Bartoli [19] cast the problem as a 2D parametrization. However, such methods are limited to simple shapes (e.g., cylinder, sheet of paper) and, to the best of our knowledge, there is no work that addressed developability in the context of MLS approximations.

In this work, we are interested by the more general class of *locally developable* surfaces that we define as surfaces that are developable almost everywhere. Given a noisy acquired point cloud, our general objective is thus to define a MLS operator that yields to a faithful and more locally-developable surface, while being resilient to noise and non-developable areas. The first two criteria mean that the defined surface should be as close as possible to the input data while exhibiting an as small as possible absolute Gaussian curvature. To this end, we introduce the *Parabolic-Cylindrical MLS* method (PC-MLS) whose key insight is to replace the general polynomial surface approximation of Levin's standard MLS method by a zero-Gaussian curvature surface proxy. In particular, we focus on bivariate degree two polynomials with a zero-Gaussian curvature. Such anisotropic surfaces are also known as *parabolic cylinders*. The main technical contribution of this paper is thus a least-square fitting procedure allowing to fit such parabolic cylinders in a continuous manner: our procedure seamlessly degenerates the parabolic cylinders to a linear model when the developability cannot be resolved unambiguously because the surface is locally flat or clearly non-developable.

We emphasize that our approach weakly enforces the dif-

ferential property of the a-priory zero-Gaussian curvature to the fitted local surface proxy rather than on the resulting MLS surface itself. This strategy preserves all the advantages of MLS approaches (fast and purely local computation without pre-computation), whereas constraining the resulting MLS surface would lead to a non-linear and global differential problem. Such an approach has already been successfully followed in the MLS literature, for instance to reconstruct isometric transformations [24] or curl-free unit-vector fields [6].

Our results show that our novel approach exhibits an excellent trade-off by producing reconstructions almost as faithful as the ones obtained with a full quadratic patch, while being more developable.

2. PC-MLS surfaces

Our MLS variant follows a classical two-step polynomial approximation strategy that we briefly present in the Section 2.1, before presenting in detail our variant based on parabolic cylinders in Section 2.2, and the resulting projection procedure in Section 2.3.

2.1. Bivariate polynomial approximation

Given a set of 3D input points $\{\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_n\}$ and an evaluation point $\mathbf{x}$ in the neighborhood of the input point cloud, the key ingredient of MLS surfaces is to approximate the input points closest to $\mathbf{x}$ by an analytic surface proxy. In former MLS approaches [14, 3], neighbor points are approximated by a bivariate polynomial height-field that is computed in two steps (see Figure 1).

The first step consists in computing a reference tangent plane approximating the neighbors of $\mathbf{x}$. This reference plane is typically obtained as the one passing through the weighted mean:

$$\bar{\mathbf{p}}(\mathbf{x}) = \frac{\sum_i \theta(\|\mathbf{x} - \mathbf{p}_i\|)\, \mathbf{p}_i}{\sum_i \theta(\|\mathbf{x} - \mathbf{p}_i\|)} \,, \tag{1}$$

with normal $\mathbf{n}(\mathbf{x})$ corresponding to the eigenvector of the smallest eigenvalue of the covariance matrix:

$$\mathrm{C} = \sum_i \theta(\|\mathbf{x} - \mathbf{p}_i\|)(\mathbf{p}_i - \bar{\mathbf{p}}(\mathbf{x}))(\mathbf{p}_i - \bar{\mathbf{p}}(\mathbf{x}))^T \,. \tag{2}$$

The weight function θ defines the neighborhood of $\mathbf{x}$. It must be monotonically decreasing, and, as in many previous works, we use the following compactly supported function:

$$\theta(t) = \begin{cases} (1 - (\frac{t}{h})^2)^4 & \text{if } t < h \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

where h defines the support radius.

If the input points are equipped with oriented normals $\mathbf{n}_i$, then the normal of the reference plane can be advantageously computed as a weighted average [2]:

$$\mathbf{n}(\mathbf{x}) = \frac{\sum_i \theta(\|\mathbf{x} - \mathbf{p}_i\|)\, \mathbf{n}_i}{\|\sum_i \theta(\|\mathbf{x} - \mathbf{p}_i\|)\, \mathbf{n}_i\|} \,.$$

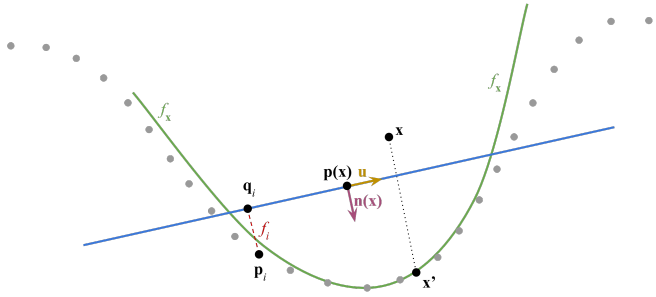


Figure 1: Illustration of the two MLS polygonal approximation steps with in gray the input points, in blue the reference plane, and in green the polynomial function $f_{\mathbf{x}}$ approximating the neighborhood of the evaluation point $\mathbf{x}$. The point $\mathbf{x}'$ is the projection of $\mathbf{x}$ onto $f_{\mathbf{x}}$.

This reference plane defines a reference frame $\mathrm{B}(\mathbf{x}) = [\mathbf{u}\,\mathbf{v}\,\mathbf{n}]$ centered at $\bar{\mathbf{p}}(\mathbf{x})$, where $\mathbf{u}$ and $\mathbf{v}$ are arbitrary tangent vectors. The neighborhood can then be seen as a height-field by decomposing each neighbor $\mathbf{p}_i$ as a pair of 2D tangential coordinates $\mathbf{q}_i$ and a height component f_i:

$$\begin{pmatrix} \mathbf{q}_i \\ f_i \end{pmatrix} = \mathrm{B}(\mathbf{x})^T(\mathbf{p}_i - \bar{\mathbf{p}}(\mathbf{x})) \,. \tag{4}$$

The second step then consists in approximating this height-field by a bivariate polynomial $f_{\mathbf{x}} : \mathbb{R}^2 \to \mathbb{R}$ in a weighted least-square sense:

$$f_{\mathbf{x}} = \arg\min_f \sum_i \theta(\|\mathbf{x} - \mathbf{p}_i\|)\, (f(\mathbf{q}_i) - f_i)^2 \,. \tag{5}$$

A typical choice is to take f among the set of quadratic polynomials, which can be expressed as:

$$f(\mathbf{q}) = c + \mathbf{g}^T\mathbf{q} + \mathbf{q}^T\mathrm{H}\mathbf{q} \,, \tag{6}$$

where c and $\mathbf{g} \in \mathbb{R}^2$ are the constant and linear coefficients, respectively, and H is the 2×2 symmetric Hessian matrix. Indeed, linear polynomials tend to over-smooth the data, while higher-order polynomials are more expensive to compute and prone to over-fitting issues (e.g., oscillations).

To compute a point on the MLS surface, one can for instance project $\mathbf{x}$ onto its local approximation $f_{\mathbf{x}}$ (see Figure 1) and iterate until convergence [2].

2.2. Parabolic cylinder approximation

As motivated above, our key insight to obtain higher quality reconstructions in the context of locally-developable surfaces consists in restricting the fitted polynomials f to the subset of developable ones. Recall that this subset is defined by the polynomials for which the Gaussian curvature vanishes. In the case of quadratic polynomials as in Eq. 6, this condition is satisfied if and only if the product of the eigenvalues of the Hessian matrix H vanishes, that is if and only if its determinant vanishes:

$$|\mathrm{H}| = 0 \,. \tag{7}$$

Quadratic surfaces fulfilling this constraint are called *parabolic cylinders*.

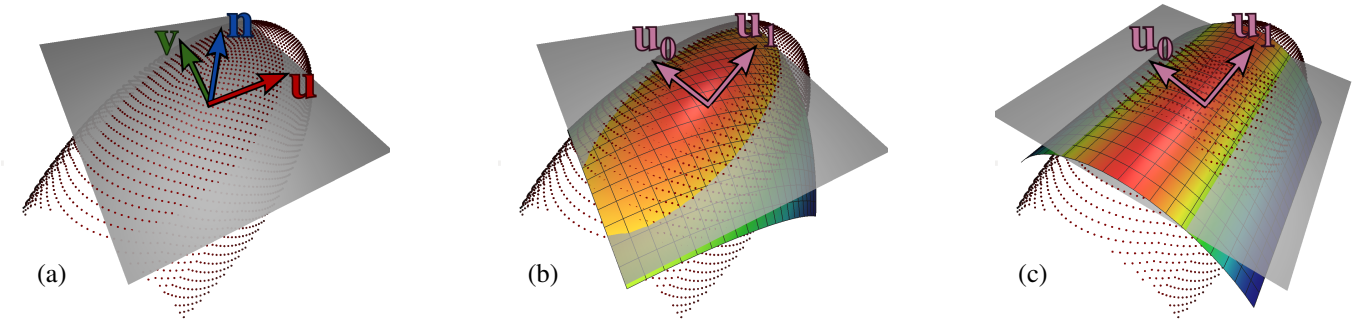


Figure 2: Illustration of our fitting procedure of *parabolic cylinders*. (a) Given a point cloud (in red), we start by fitting a plane (in grey) defining a reference frame $[\mathbf{u}\,\mathbf{v}\,\mathbf{n}]$. (b) Then, interpreting the neighbors at a height-field, fitting a full quadratic polynomial allows us to estimate the principal directions of curvature $[\mathbf{u_0}\,\mathbf{u_1}]$. (c) The last step consists in fitting a second quadratic polynomial within this new reference frame and such that there is no quadratic variation along the $\mathbf{u_0}$ axis. This yields a parabolic cylinder aligned with the directions of principal curvatures.

Directly integrating this constraint into the polynomial fit (Eq. 5) would lead to a costly non-convex minimization problem, which is not compatible with a MLS approach. Indeed, let us recall that in a MLS framework, the number of fits that have to be carried out is typically of the order of millions, and the efficiency of the fitting procedure is thus crucial. More importantly, the existence and continuity of the resulting surface is conditioned by the continuity of the fitting procedure: $f_{\mathbf{x}}$ must smoothly vary with respect to small changes of $\mathbf{x}$ and small changes of the input points. Since we are in the presence of a non-convex problem, the continuity of $f_{\mathbf{x}}$ cannot be satisfied. In this case, the projection procedure sketched above might not converge at all, and no surface will be defined in some area.

To address the performance issue, we propose a two-step fitting procedure of *parabolic cylinders* involving only two linear problems and one 2×2 eigenvalue problem. To address the continuity issue, we extend our direct fitting procedure by allowing *parabolic cylinders* to seamlessly degenerate to linear polynomials when the problem becomes ambiguous. These two ingredients are detailed below.

Direct fitting of parabolic cylinders

Our direct fitting procedure of parabolic cylinders is depicted in Figure 2. As in standard MLS, it starts by the fitting of a reference frame and a complete quadratic polynomial (Eq. 6). Then, we perform an eigenvalue decomposition of the resulting Hessian matrix:

$$\mathrm{H} = \mathrm{U}\begin{pmatrix} \lambda_0 & 0 \\ 0 & \lambda_1 \end{pmatrix}\mathrm{U}^T , \tag{8}$$

where $\mathrm{U} = (\mathbf{u}_0\ \mathbf{u}_1)$ is the eigenvector matrix, and the eigenvalues are ordered such that $|\lambda_0| \geq |\lambda_1|$. Since the principal curvatures are proportional to these eigenvalues, our insight consists in enforcing its smallest eigenvalue λ_1 to be zero while maintaining the quadratic polynomial as close as possible to the input data. Practically, this is accomplished by aligning our current frame with the eigenvectors U, which also correspond to the principal curvature directions. Within this new frame, the Hessian coincides with the diagonal eigenvalue matrix, and enforcing λ_1 is accomplished by reducing the polynomial basis as follows:

$$f(\mathbf{q}) = c + \mathbf{g}^T\mathbf{q} + a(\mathbf{u_0}^T\mathbf{q})^2 , \tag{9}$$

where the new coefficients c, $\mathbf{g}$, and a are computed through a second linear minimization as in Eq. 5.

To summarize, this procedure is equivalent to first align the reference frame with the principal curvatures, and then fit a bivariate polynomial that is linear in one dimension, and quadratic in the other.

Continuous fitting of parabolic cylinders

The above procedure is fast but becomes unstable as soon as the eigenvalues are of similar magnitude, in which case discontinuities occur (Figure 3). This issue is also depicted in the following figure on a saddle configuration ($\lambda_0 \approx -\lambda_1$) in which case a very small variation of the evaluation point $\mathbf{x}$ leads to two very different fits. Umbilical regions (i.e., $\lambda_0 \approx \lambda_1$) exhibit a similar behavior.

To address this issue and ensure a continuous fit, we enforce the parabolic cylinder to degenerate to a plane when the difference of the eigenvalue magnitudes vanishes (Figure 4). This is accomplished through a third linear fit: the constant and linear coefficients c and $\mathbf{g}$ are re-computed by fitting the polynomial basis of Eq. 9 with the quadratic coefficient fixed to αa, where the value of a comes from the previous fit. In practice, this refitting is equivalent to a linear blend proportional to α between the parabolic cylinder obtained in the previous step, and the fit of a linear polynomial.

The coefficient α is computed such that it is equal to 1 if the problem is *unambiguously* developable, and 0 when the problem tends to an umbilical or saddle configuration. In order to get a scale invariant measure, we propose to compute α as:

$$\alpha = \min\left(1, 2\frac{|\lambda_0| - |\lambda_1|}{|\lambda_0| + 1/h}\right) . \tag{10}$$

In this equation, the normalization by $|\lambda_0|$ would have been a natural choice as it would adapt to the maximum curvature. However, normalizing the difference of the eigenvalue magnitudes by $|\lambda_0|$ only would make the estimation of α unstable in flat but noisy areas. Indeed, when λ_0 is small, small variations

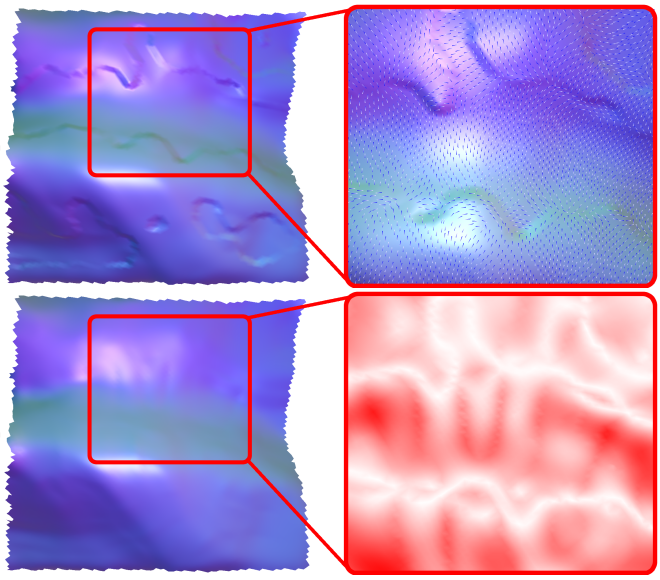


Figure 3: Top row: The output surface computed using exclusively *parabolic cylinders* on a fabric model. Zoom with direction of curvature on the output surface (right). Note the correspondences between discontinuities and the variations of the curvature directions. Bottom row: The output surface using our continuous fitting procedure of *parabolic cylinders*. Zoom on the α weighting (right). The more white, the more constrained is the local fit.

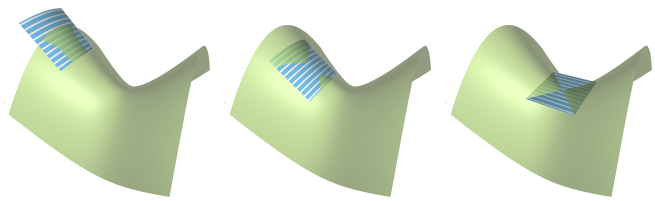


Figure 4: Illustration of the local approximation with PC-MLS at 3 different locations. When approaching a region where the developability cannot be resolved unambiguously (here: a saddle configuration), the local approximation seamlessly degenerates to a linear model.

of the eigenvalues lead to high variations of α. We thus add to the normalization term $|\lambda_0|$ an amount of curvature $1/h$ that we consider to be meaningful with respect to the current scale and neighborhood size h. Finally, the factor 2 permits to saturate to one as soon as the problem is clearly unambiguous.

We emphasize that this coefficient does not exactly measure the local level of developability of the point cloud, but rather the level of ambiguity of a parabolic cylindrical fit. In practice, we found such a measure more stable than taking, for instance, the absolute Gaussian curvature. To understand why, let us consider a simple example of a nearly flat and isotropic area. This area will obtain a very low α value even though it is almost-developable. In this case, enforcing a linear fit is important to guarantee the convergence of the projection procedure, even though this will have almost no effect on the result since the quadratic fit is already flat anyway. This α coefficient is depicted in Figure 6 on a garment model where, as expected, it can be seen that all inflection and flat areas exhibit a low value of α.

2.3. *Projection operator*

To summarize the different steps of our approach, we present below the algorithm that projects an arbitrary point $\mathbf{x}$ onto our PC-MLS surface. This iterative projection procedure follows the *almost orthogonal* projection strategy [2].

Algorithm 1: PC-MLS

Input: An evaluation point $\mathbf{x}$ and a set of points $\{\mathbf{p}_i\}$
Output: The projection $\mathbf{x}'$ of $\mathbf{x}$ on the PC-MLS surface

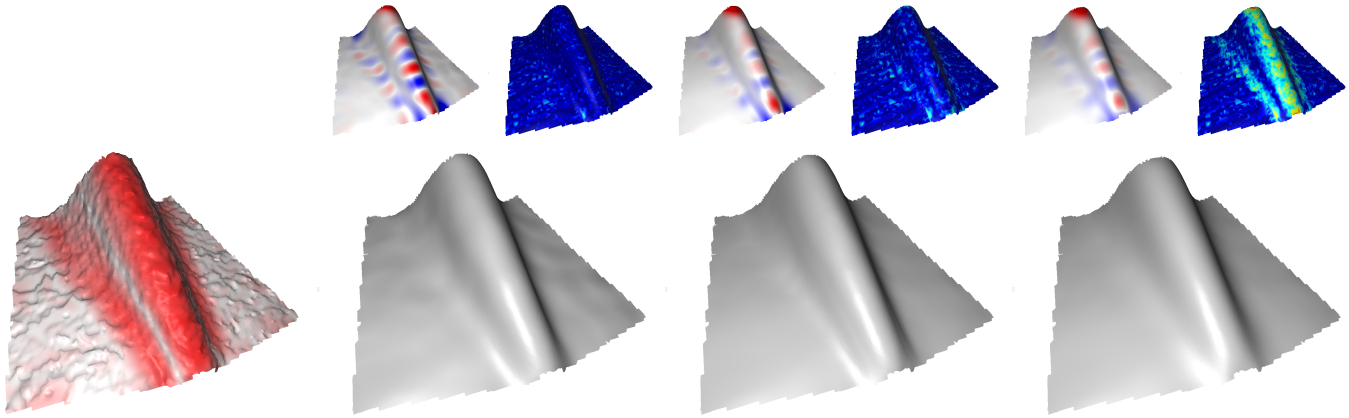
$\mathbf{x}^0 \leftarrow \mathbf{x}$
$k \leftarrow 0$
do
  Collect weighted neighbors of $\mathbf{x}^k$ — Eq. 3
  Compute reference frame $\mathrm{B}(\mathbf{x}^k), \bar{\mathbf{p}}(\mathbf{x}^k)$ — Eq. 1,2
  Express neighbors in reference frame — Eq. 4
  Compute Hessian matrix H *by fitting a full quadratic polynomial* — Eq. 6,5
  Compute eigenvalue decomposition of H — Eq. 8
  Compute quadratic coefficient a by fitting an axis-aligned parabolic cylinder — Eq. 9,5
  Update quadratic coefficient: $a \leftarrow \alpha a$ — Eq. 10
  Compute constant and linear coefficients c, $\mathbf{g}$ *by fitting an axis-aligned parabolic cylinder with the quadratic coefficient fixed* — Eq. 9,5
  Projection of $\mathbf{x}$ *onto* $f_{\mathbf{x}^k}$: $\mathbf{x}^{k+1} \leftarrow \mathrm{B}^T(\mathbf{x} - \bar{\mathbf{p}}) + f_{\mathbf{x}^k}([\mathbf{u}\ \mathbf{v}]^T(\mathbf{x} - \bar{\mathbf{p}})) \cdot \mathbf{n}$
  $k \leftarrow k + 1$
while $\|\mathbf{x}^k - \mathbf{x}^{k-1}\| > \epsilon$
return $\mathbf{x}^k$

3. Results

Performance-wise, our approach remains similar to a standard MLS approach based on the fitting of quadratic polynomials. Indeed, the cost of the additional operations required to continuously fit our parabolic cylinders is small compared to the cost of collecting the neighbors and performing the full quadratic fit. The table below gives some timings in milliseconds for the projection procedure of the input points to the respective MLS surface using either a full quadratic polynomial basis, our method, or a linear basis. The latter method appears to be slower than the use of a full quadratic basis because it requires more iterations to converge.

	# points	Quadratic	PC-MLS	Linear
half-cylinder (Fig 7)	20k	714	1050	762
garment (Fig 6)	50k	1010	1235	1150

In order to evaluate the effect of parabolic cylinders for MLS surface reconstruction, we compare our approach to the two standard MLS surfaces (as described in Section 2.1) based on the fitting of full quadratic polynomials and linear polynomials, respectively. The behavior of each of these methods is depicted in Figures 5, 6, 8, and 9 on various point clouds obtained



sheet of paper 2 | full quadratic poly. | **our method** | linear poly.

Figure 5: Comparison of three MLS variants on a scanned sheet of paper (11k points). Input model and its associated α weight (left). Top row shows Gaussian curvature and distance between input and output surfaces (the max. distance in red is equal to 1.59%). Bottom row shows the computed surfaces. Note that our method has a lower overall absolute Gaussian curvature while remaining close to the data.

Model	Developability								Mean distance (%)		
	Input Model		Quadratic poly.		PC-MLS Method		Linear poly.		Quadratic poly.	PC-MLS Method	Linear poly.
	Mean	Std Dev	Mean	Std Dev	Mean	Std Dev	Mean	Std Dev			
Half-cylinder	1.30e-05	1.71e-05	8.09e-06	9.93e-06	**4.20e-06**	**5.68e-06**	**3.85e-06**	**5.08e-06**	**0.16**	**0.17**	**0.18**
Sheet of paper 1	1.11e-04	5.03e-04	8.72e-05	3.56e-04	5.08e-05	1.74e-04	**3.93e-05**	**1.18e-04**	**0.01**	0.023	0.026
Sheet of paper 2	4.23e-04	7.53e-04	3.37e-04	5.47e-04	**2.18e-04**	**4.56e-04**	**2.26e-04**	**4.05e-04**	**0.12**	**0.16**	0.28
Garment	1.04e-03	1.97e-03	9.90e-04	2.00e-03	**7.47e-04**	**1.70e-03**	**6.45e-04**	**1.33e-03**	**0.042**	**0.051**	0.103
Fabric	1.32e-04	1.82e-04	9.46e-05	1.22e-04	**4.24e-05**	**5.46e-05**	**3.70e-05**	**5.38e-05**	**0.33**	0.41	0.47
Half-pipe	6.19e-03	2.20e-02	5.15e-03	1.80e-02	**2.94e-03**	**9.61e-03**	**2.30e-03**	**7.33e-03**	**0.042**	0.081	0.13

Table 1: Comparison of developability and mean distance. Best measures with 15% of tolerance in **bold**.

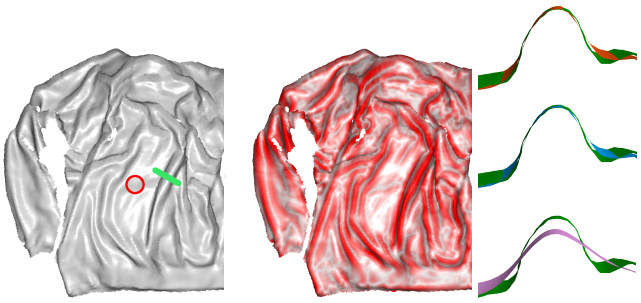
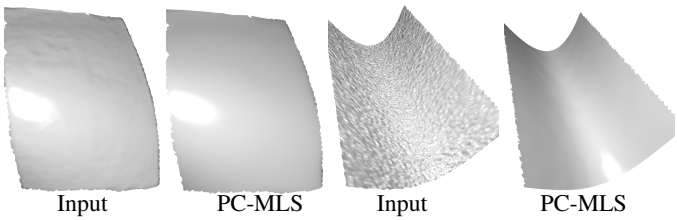


Figure 6: Garment model with the associated scale in red and a cutting plane in green (left). Visualization of the α weight (middle). On the right, reconstruction profiles on a wrinkle with quadratic polynomials (orange), our PC-MLS method (blue), and with linear polynomials (pink). The input data is shown in green.



Input | PC-MLS | Input | PC-MLS

Figure 7: Other models presented in Table 1: sheet of paper 1 (left), and half-cylinder (right) with both the input and the PC-MLS reconstruction.

with Kinect Fusion [12]. In all these results, it can be seen that the full quadratic fit tends to reproduce the input noise. This observation is confirmed in Figures 5 and 9 by a visualization of the Gaussian curvature. Reconstructions made by a full quadratic fit clearly exhibit a larger amount of absolute Gaussian curvature than the ones performed with our approach or linear fits. The distances to the input data are visualized in Figure 6 through a cut, and in Figure 5 as a color code. It can be seen that our approach approximates the data almost as tightly as a full quadratic fit, whereas linear fits over-smooth the surface features. Overall, our approach provides a good compromise by producing smooth reconstructions which tightly fit the input data.

In addition to visual inspections of the reconstructed surfaces, we measured the Hausdorff distance between the input and the output surfaces [7]. We also measured the amount of developability of the reconstructed surfaces as the average and standard deviation of the Gaussian curvature that we estimated from the quadratic fits as in the Jet-fitting method [5]. These measures are summarized in Table 1 for six different point clouds of the previous figures and the ones of Figure 7. These objective measures confirm the previous observations: it can be seen that overall, the developability of the surfaces obtained with our method is close to the one obtained with a linear fit, while providing much tighter reconstructions. On the other hand, full quadratic fits exhibit a significantly worse developability.

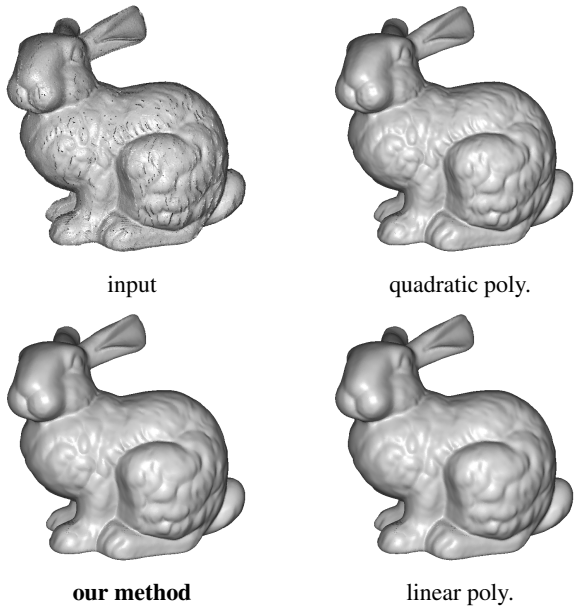


Figure 11: The raw point cloud of the Stanford bunny model reconstructed with different MLS variants.

The sensitivity to noise of our method is depicted in Figure 10 on a synthetically perturbed point cloud. As can be seen, despite a very large amount of noise, our method remains consistent and close to the original data.

Finally, Figure 11 shows how our approach compares on a model for which none of the parts are nearly developable. In this case, our developability prior is counterproductive and, as expected, using a full quadratic polynomials basis is preferred. Nonetheless, this result shows that even for unsuitable models, our approach does not introduce unwanted artifacts and still performs better than a linear basis by producing a sharper reconstruction.

4. Conclusion

In this paper, we introduced a fast least-square fitting method of parabolic cylinders that is continuous in space and scale. Thanks to these properties, our fitting method can be seamlessly integrated within a MLS framework. We have shown that parabolic cylinders achieve higher quality approximations when the underlying surface can be assumed to be partly developable. Our approach preserves all the nice properties of MLS surfaces: no precomputation, purely local computation, low-memory consumption, works at arbitrary scale, etc. Our approach is thus ready for high versatility of use.

Central to our approach is a detection of areas that are not locally developable for which we continuously fall back to a linear approximation. Currently, this detection is carried out at the actual reconstruction scale by measuring some kind of level of ambiguity of the parabolic cylinder fit. As future work, it might be beneficial to study whether some kind of multi-scale analysis [17] could allow to resolve some of the ambiguous cases as this might allow to produce even more developable reconstructions. Our approach could also be easily adapted to fall back to a full quadratic approximation instead of a linear one. In the ambiguous areas, this would have the effect to favor fidelity to the data over smoothness and developability. Finally, as the original MLS method for surfaces, our current approach is limited by the need of a local reference frame parametrizing the input point cloud, and it would also be interesting to extend our approach to other high-order parametrization-free MLS methods such as [11].

Acknowledgements

This project has received funding from the V-MusT.net Network of Excellence (European Community's Seventh Framework Programme under Grant Agreement 270404).

References

[1] A. Adamson and M. Alexa. Anisotropic point set surfaces. In *Proceedings of the 4th International Conference on Computer Graphics, Virtual Reality, Visualisation and Interaction in Africa*, AFRIGRAPH '06, pages 7–13. ACM.

[2] M. Alexa and A. Adamson. On normals and projection operators for surfaces defined by point sets. In *Proc. of SPBG'04*, pages 149–156.

[3] M. Alexa, J. Behr, D. Cohen-Or, S. Fleishman, D. Levin, and C. T. Silva. Point set surfaces. In *Proceedings of the Conference on Visualization '01*, VIS '01, pages 21–28. IEEE Computer Society, 2001.

[4] N. Amenta and Y. J. Kil. Defining point-set surfaces. *ACM Trans. Graph.*, 23(3):264–270, 2004.

[5] F. Cazals and M. Pouget. Estimating differential quantities using polynomial fitting of osculating jets. In *Proc. of Eurographics SGP'03*, pages 177–187.

[6] J. Chen, G. Guennebaud, P. Barla, and X. Granier. Non-oriented MLS gradient fields. *Computer Graphics Forum*, 32(8):98–109, 2013.

[7] P. Cignoni, C. Rocchini, and R. Scopigno. Metro: Measuring error on simplified surfaces. Technical report, Paris, France, France, 1996.

[8] T. K. Dey and J. Sun. An adaptive MLS surface for reconstruction with guarantees. In *Proc. Eurographics SGP'05*.

[9] M. P. Do Carmo and M. P. Do Carmo. *Differential geometry of curves and surfaces*, volume 2. Prentice-hall Englewood Cliffs, 1976.

[10] M. Eigensatz, R. W. Sumner, and M. Pauly. Curvature-Domain Shape Processing. *Computer Graphics Forum*, 2008.

[11] G. Guennebaud and M. Gross. Algebraic point set surfaces. In *ACM SIGGRAPH'07*.

[12] S. Izadi, D. Kim, O. Hilliges, D. Molyneaux, R. Newcombe, P. Kohli, J. Shotton, S. Hodges, D. Freeman, A. Davison, and A. Fitzgibbon. Kinectfusion: Real-time 3D reconstruction and interaction using a moving depth camera. In *Proc. of ACM UIST'11*, pages 559–568.

[13] M. Kilian, S. Flöry, Z. Chen, N. J. Mitra, A. Sheffer, and H. Pottmann. Curved folding. *ACM Trans. Graph.*, 27(3):75:1–75:9, Aug. 2008.

[14] D. Levin. Mesh-independent surface interpolation. *Geometric Modeling for Scientific Visualization*, 3, 2003.

[15] Y. Lipman, D. Cohen-Or, and D. Levin. Data-dependent MLS for faithful surface approximation. In *Proc. of Eurographics SGP'07*, pages 59–67.

[16] Y. Liu, H. Pottmann, J. Wallner, Y.-L. Yang, and W. Wang. Geometric modeling with conical meshes and developable surfaces. *ACM Trans. Graph.*, 25(3):681–689, July 2006.

[17] N. Mellado, G. Guennebaud, P. Barla, P. Reuter, and C. Schlick. Growing least squares for the analysis of manifolds in scale-space. *Comp. Graph. Forum*, 31(5):1691–1701, Aug. 2012.

[18] C. Öztireli, G. Guennebaud, and M. Gross. Feature preserving point set surfaces based on non-linear kernel regression. *Proc. Eurographics CGF'09*, 28(2):493–501.

[19] M. Perriollat and A. Bartoli. A quasi-minimal model for paper-like surfaces. In *IEEE CVPR'07*, pages 1–7.

[20] M. Peternell. Developable surface fitting to point clouds. *Comput. Aided Geom. Des.*, 21(8):785–803, Oct. 2004.

[21] H. Pottmann and J. Wallner. Approximation algorithms for developable surfaces. *Computer Aided Geometric Design*, 16(6):539 – 556, 1999.

[22] P. Reuter, P. Joyot, J. Trunzler, T. Boubekeur, and C. Schlick. Surface reconstruction with enriched reproducing kernel particle approximation. In *Proc. of SPBG'05*, pages 79–87.

[23] K. Rose, A. Sheffer, J. Wither, M.-P. Cani, and B. Thibert. Developable surfaces from arbitrary sketched boundaries. In *Eurographics SGP'07*, pages 163–172.

[24] S. Schaefer, T. McPhail, and J. Warren. Image deformation using moving least squares. *ACM Trans. Graph.*, 25(3):533–540, 2006.

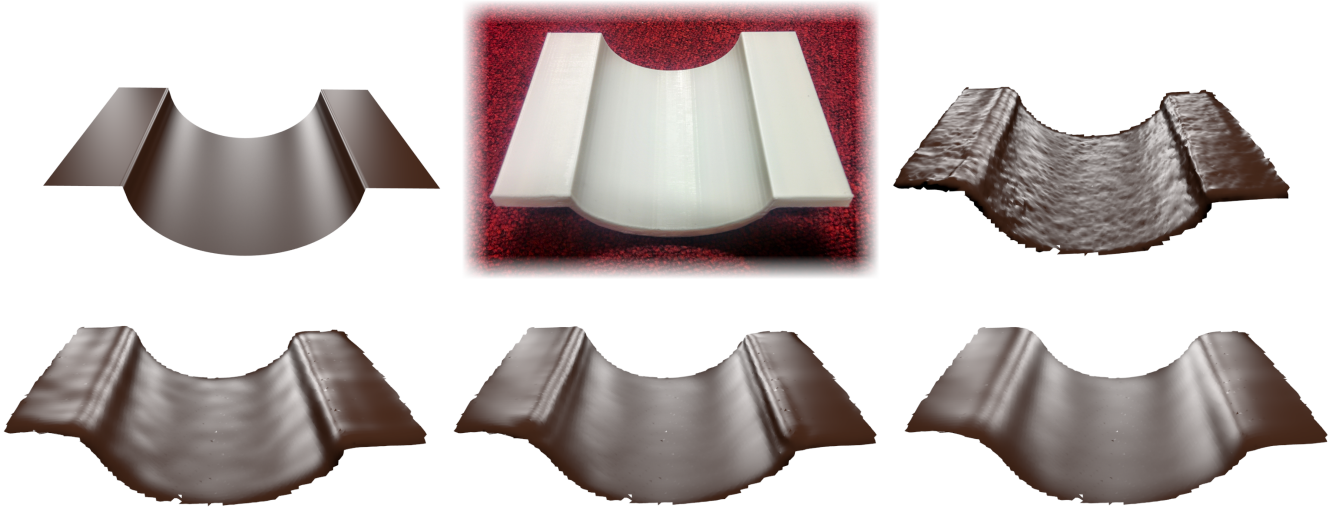
[25] J. Solomon, E. Vouga, M. Wardetzky, and E. Grinspun. Flexible developable surfaces. *Comp. Graph. Forum*, 31(5):1567–1576, Aug. 2012.

[26] K. Tang and M. Chen. Quasi-developable mesh surface interpolation via mesh deformation. *IEEE TVCG'09*, 15(3):518–528.

[27] C. C. L. Wang. Towards flattenable mesh surfaces. *Comput. Aided Des.*, 40(1):109–122, Jan. 2008.

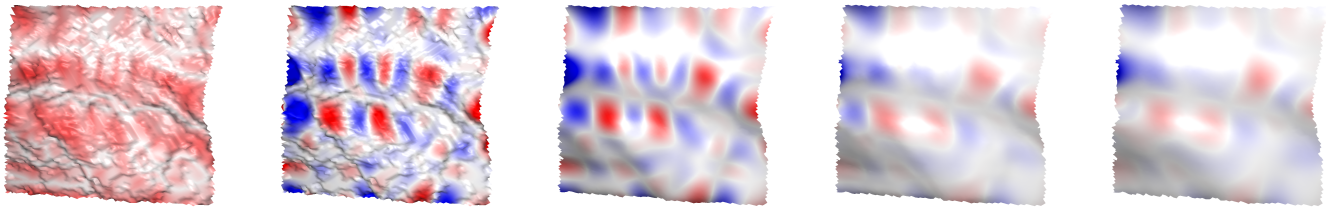
[28] C. C. L. Wang and K. Tang. Achieving developability of a polygonal surface by minimum deformation: A study of global and local optimization approaches. *The Visual Computer*, 20(8-9):521–539, Nov. 2004.

[29] L. Zeng, Y.-J. Liu, M. Chen, and M. M.-F. Yuen. Least squares quasi-developable mesh approximation. *Computer Aided Geometric Design*, 29(7):565–578, 2012.



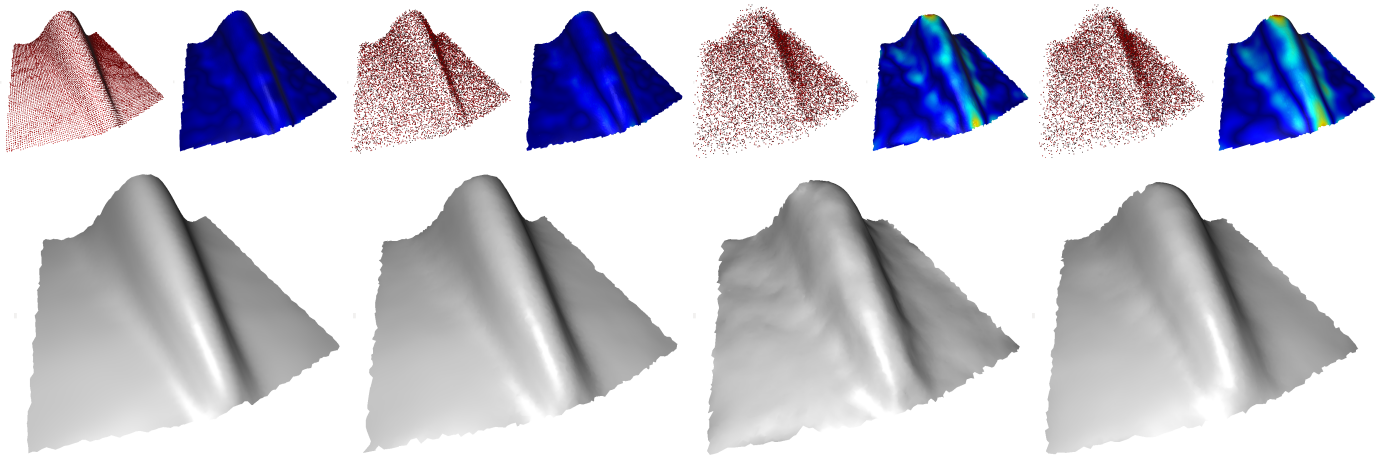
full quadratic poly. **our method** linear poly.

Figure 8: Top row illustrates a CAD half-pipe model, the printed 3D version, and its scanned version. Bottom row shows the reconstruction with three MLS variants. Note the distance of the linear reconstruction at the sharp edges, and the bumps on the surface with the quadratic reconstruction. The reconstruction with our method stays flat and close to the surface.



α input full quadratic poly. **our method** linear poly.

Figure 9: Fabric model. Left image: visualization of the weight α (the more red, the more α is high). Right images: visualization of the Gaussian curvature of the input surface and reconstruction with various methods.



0.1% of noise (scale of 9%) 0.9% of noise (scale of 9%) 3% of noise (scale of 9%) 3% of noise (scale of 13%)

Figure 10: Illustration of the sensitivity of our method to noise. From left to right, results of the PC-MLS on model perturbed with different amounts of random noise (bottom row). Top row shows the input noisy model and distance between smoothed and approximated model (the max. distance in red is equal to 2.31%). Our PC-MLS reconstruction remains consistent and close to the data.