Fitting Smooth Manifolds to Point Clouds in a Level Set Formulation

Hossein Soleimani(✉), George Poothicottu Jacob, and Oleg V. Michailovich

University of Waterloo, Waterloo, Canada
{h3soleim,gpoothic,olegm}@uwaterloo.ca

Abstract. Since its inception, curves and surfaces have been the principal means of representation of observed geometry in computer vision. In many practical applications, one's knowledge of the shapes of real-life objects is obtained through discrete measurements, which are subsequently converted into their continuous counterparts through the process of either curve or surface fitting, depending on the object dimensionality. Unfortunately, the measurement noise due to environmental effects, operator errors and/or hardware limitations makes the fitting problem a challenging one, requiring its solutions to possess a substantial degree of robustness. Moreover, in the case of surface fitting, the use of relatively complex fitting mechanisms might be disadvantageous due to their typically higher computational requirements, which could, in turn, create an implementation bottleneck due to the high dimensionality of the data. Accordingly, in this work, we propose a unified approach to fitting of smooth geometric manifolds, such as curves and surfaces, to point clouds. The proposed method is based on a level-set formulation, which leads to a simple and computationally efficient algorithm, the practical value of which is demonstrated through a series of examples.

Keywords: Surface fitting · Point clouds · Level-set functions · Total-variation

1 Introduction

The problem of fitting of smooth geometric manifolds, such as spatial curves and surfaces, to finite sets of Euclidean points is unarguably one of the classical problems of image processing and computer vision. Such sets of points, commonly referred to as *point clouds*, are typically acquired by means of range/depth sensors to provide a raw description of the shape and geometry of scanned objects. Due to their irregular and noisy nature, however, point clouds are rarely used by higher level algorithms and visualization software. Instead, the latter "prefer" to process continuous representations of observed geometries, which are derived from point clouds through the process of curve/surface fitting. Virtual and augmented reality, reverse engineering, and medical imaging are only a few examples of application areas, where such fitting problems routinely arise [1].

F. Karray et al. (Eds.): ICIAR 2019, LNCS 11662, pp. 139–149, 2019.
https://doi.org/10.1007/978-3-030-27202-9_12

Moreover, the recent advance in digital scanning technologies has resulted in a widespread proliferation of a variety of new depth/range scanning devices, which often present unexplored challenges from the viewpoint of curve/surface fitting. To cope with these challenges, a broad spectrum of fitting methodologies has been so far proposed [2], which generally differ in their ability to withstand the effects of noise and scarce sampling as well as to effectively deal with non-trivial topologies [3].

The current arsenal of methods for curve/surface fitting is vast, suggesting a variety of possible taxonomies. A particular way to classify such methods could be based on their property of being either *explicit* or *implicit* [4]. In the explicit formulation, it is standard to represent surfaces by means of triangulated meshes obtained based on Voronoi diagrams [5] or Delaunay triangulation [6]. In their core, such methods are based on systematically connecting the points of a cloud until a predefined completeness criterion is met. Under weak noise conditions and in the absence of considerable gaps between data points, explicit representations are known to provide satisfactory and useful results. Unfortunately, their quality degrades quickly when the aforementioned assumptions start to fail.

In implicit representation, curves and surfaces are typically defined by means of either the indicator functions of closed subsets of $\mathbb{R}^d$ or the level sets of Lipschitz-continuous functions [3,7]. Thus, for instance, the fitting procedure in [8] utilized the zero level-set of a distance function, in which case it was also possible to determine the topological types of reconstructed surfaces. While promising in many respects, however, the method lacked in geometric accuracy. In [9], the same group of authors introduced an accurate surface model for unorganized point data. Unfortunately, this work provided little evidence on the performance of the proposed method in the case of incomplete and irregularly sampled data [10]. A method called Poisson Surface Reconstruction was introduced in [11] based on the observation that the normal field of the boundary of a solid can be interpreted as the gradient of the indicator function of a surface, which can, in turn, be used to model this surface in an implicit manner. Further modifications of this method were described in [12], including its adjustments to the case of sparse point clouds and improvements of computational efficiency. An interesting method exploiting the underlying geometric structure of point clouds in combination with a convex formulation of image segmentation was described in [3]. The method has been shown to perform reliably under a variety of conditions, including scenarios with incomplete/scarce data and complex topologies. Yet, to achieve such performance, one requires to possess prior information in the form of point normals, which might not always be available. Finally, a surface fitting formulation based on the concept of function approximation by means of Radial Basis Functions (RBF) was reported in [13]. In addition to its other merits, the method was also shown to be computationally efficient, which makes it particularly valuable when dealing with large data sets [14].

The level-set framework, in which curves and surfaces are defined in terms of the level-set of Lipschitz-continuous functions, offers a number of critical advantages, among which are the simplicity of parameterization and the ability to deal with relatively complex topologies in an easy and straightforward manner. The practical implementation of such methods in the context of curve/surface fitting, however, may be rather challenging, thereby risking to put these methods in disadvantage with respect to other techniques. Accordingly, the main purpose of this work is to introduce a level-set based formulation of the problem of curve/surface fitting which offers two important advantages. First, the formulation is independent of the problem dimensionality, and it can be applied in the cases of curve and surface fitting with virtually no adjustments. Second, the formulation leads to a particularly simple and computationally efficient solution, which allows its application in both time-critical and data-extensive scenarios.

The rest of this paper is organized as follows. Sections 2 and 3 provide a formal description of the fitting problem and its solution, respectively. Experimental results with 2D and 3D point clouds are presented in Sect. 4, followed by a discussion and conclusions in Sect. 5.

2 Problem Formulation

Let $\phi : \mathbb{R}^d \rightarrow \mathbb{R}$ be a Lipschitz-continuous function (with $d = 2$ and $d = 3$ corresponding to the cases of curve and surface fitting, respectively), whose zero level set

$$\Gamma = \left\{ \mathbf{r} \in \mathbb{R}^d \mid \phi(\mathbf{r}) = 0 \right\}$$

will be used to model the geometric manifold of interest (i.e., either a curve or a surface). In particular, in what follows, the *level-set* function $\phi(\mathbf{r})$ will be defined to be the *signed distance function* (SDF) of the zero level-set Γ of ϕ. Such SDF is given by the unique solution of Eikonal equation

$$\begin{cases} |\nabla \phi(\mathbf{r})| = 1, & \forall \mathbf{r} \in \mathbb{R}^d \\ \phi(\mathbf{r}) = 0, & \forall \mathbf{r} \in \Gamma \end{cases},$$

with $\nabla \phi$ standing for the gradient of ϕ and $|\nabla \phi(\mathbf{r})|$ being its magnitude at $\mathbf{r}$.

By the nature of its definition, the value $\phi(\mathbf{r})$ of an SDF ϕ returns the distance between $\mathbf{r}$ and the zero-level set Γ. Thus, given a set of N points $\{\mathbf{r}_i\}_{i=1}^N$, the average cumulative distance between the point cloud and Γ can be computed as

$$D(\phi) = \frac{1}{N} \sum_{i=1}^{N} |\phi(\mathbf{r}_i)|, \tag{1}$$

where the absolute value is used to cancel out the effect of the sign of ϕ (which we assume to have negative values inside Γ). Clearly, the closer the points $\mathbf{r}_i$ to Γ, the lower the value of $D(\phi)$. Consequently, the fitting problem at hand can be formulated as a minimization problem in ϕ, whose optimal solution returns the minimum possible value of $D(\phi)$.

An obvious flaw of the above approach consist in its proneness to the "curse of dimensionality". Indeed, in an attempt to minimize $D(\phi)$, the optimization is likely to result in a solution with an excessively variable, fluctuating Γ which might contradict the physical nature of the object represented by the point cloud. Such situations are particularly frequent in the case of noisy and/or incomplete (scarce) measurements, which necessitates the use of regularization. To this end, it has been proven effective to penalize the area of Γ, which (owing to the co-area formula), amounts to requiring the SDF ϕ to be a function of bounded variation (BV) [19]. Such functions are known to have relatively small values of their *total-variation* (TV) semi-norms defined as

$$\|\phi\|_{\mathrm{TV}} = \int_{\mathbf{r}\in\mathbb{R}^d} |\nabla\phi(\mathbf{r})| \, d\mathbf{r}. \tag{2}$$

Subsequently, to guarantee the zero level-set Γ (and, hence, the resulting manifold) has a plausible configuration, the minimization of $D(\phi)$ should be restricted to the functions with relatively small values of TV semi-norms, which leads to an optimization problem of the form

$$\min_{\phi} \left\{ D(\phi) + \lambda \|\phi\|_{\mathrm{TV}} \right\}, \tag{3}$$

where $\lambda > 0$ is a user-defined regularization constant. Denoting the cost functional in (3) by $E(\phi)$, namely $E(\phi) := D(\phi) + \lambda\|\phi\|_{\mathrm{TV}}$, its (local) minimizers ϕ^* are characterized by the Euler-Lagrange condition

$$\frac{\delta E(\phi^*)}{\delta\phi} = 0, \tag{4}$$

with $\delta E/\delta\phi$ being the first variational derivative of $E(\phi)$. A particularly simple and computationally efficient solution to the above equation is detailed in the following section.

3 Proposed Solution

A solution to the Euler-Lagrange equation (4) can be found as a stationary point of the *gradient flow* given by [15]

$$\frac{\partial\phi}{\partial t} = -\frac{\delta E(\phi)}{\delta\phi}, \tag{5}$$

where $t > 0$ is an artificial time that can be thought of as a continuous equivalent of an iteration index. Thus, to implement the gradient flow, the first variational derivative $\delta E/\delta\phi$ needs to be computed first. To this end, one can first redefine $D(\phi)$ as

$$D(\phi) = \frac{1}{N}\sum_{i=1}^{N} |\phi(\mathbf{r}_i)| = \frac{1}{N}\sum_{i=1}^{N} \int |\phi(\mathbf{r})| \, \delta(\mathbf{r}-\mathbf{r}_i) \, d\mathbf{r} = \int |\phi(\mathbf{r})| \, g_N(\mathbf{r}) \, d\mathbf{r}$$

with

$$g_N(\mathbf{r}) = \frac{1}{N} \sum_{i=1}^{N} \delta(\mathbf{r} - \mathbf{r}_i)$$

and $\delta(\mathbf{r})$ be the standard Dirac delta function. In this case, the first variational derivative of $D(\phi)$ can be shown to have the form of

$$\frac{\delta D(\phi)}{\delta \phi} = \text{sign}(\phi) \cdot g_N,$$

where we have adopted the sign function, sign(x), as a practical approximation of the first-order derivative of $|x|$. On the other hand, the first variational derivative of $\|\phi\|_{\text{TV}}$ is well known to be equal to $-\text{div}(\nabla\phi/|\nabla\phi|)$, with div standing for the operator of divergence. Therefore, combining the two results leads to

$$\frac{\delta E(\phi)}{\delta \phi} = \text{sign}(\phi) \cdot g_N - \lambda \, \text{div}(\nabla\phi/|\nabla\phi|), \tag{6}$$

and, subsequently, to the gradient flow of the form

$$\frac{\partial \phi}{\partial t} = \lambda \, \text{div}(\nabla\phi/|\nabla\phi|) - \text{sign}(\phi) \cdot g_N. \tag{7}$$

The next step is to discretize (7), to which end we take advantage of a *semi-implicit* approach to produce

$$\frac{\phi_{t+\Delta t} - \phi_t}{\Delta t} = \lambda \, \text{div}(\nabla\phi_{t+\Delta t}/|\nabla\phi_{t+\Delta t}|) - \text{sign}(\phi_{\text{t}}) \cdot g_N,$$

where ϕ_t and $\phi_{t+\Delta t}$ denote the values of ϕ at times t and $t + \Delta t$, respectively. Thus, rearranging the terms in the above numerical approximation, one obtains

$$\phi_{t+\Delta t} - (\lambda \Delta t) \, \text{div}(\nabla\phi_{t+\Delta t}/|\nabla\phi_{t+\Delta t}|) = \phi_t - \Delta t \, \text{sign}(\phi_{\text{t}}) \cdot g_N. \tag{8}$$

To conclude the derivations, we recall that any real subdifferentiable functional $F : \mathcal{X} \to \mathbb{R}$ with a monotone subdifferential (where $\mathcal{X}$ is the Banach space to which ϕ is supposed to belong, such as the space of BV functions in our case) can be associated with its *proximal map* $\text{prox}_F : \mathcal{X} \to \mathcal{X}$ defined as [16]

$$\text{prox}_F(\psi) = \arg \inf_{\phi \in \mathcal{X}} \left\{ \frac{1}{2} \|\phi - \psi\|_2^2 + F(\phi) \right\}. \tag{9}$$

In the special case when $F(\phi) = \tau\|\phi\|_{\text{TV}}$, with some $\tau > 0$, one has

$$\text{prox}_{\tau\|\cdot\|_{\text{TV}}}(\psi) = \arg \inf_{\phi \in \mathcal{X}} \left\{ \frac{1}{2} \|\phi - \psi\|_2^2 + \tau\|\phi\|_{\text{TV}} \right\}, \tag{10}$$

which is always well-defined due to the strict convexity of its associated cost functional. Although this operator does not admit a closed-form definition, there are a number of numerical procedures that can be used to compute $\text{prox}_{\tau\|\cdot\|_{\text{TV}}}$ in a computationally efficient manner [17,18]. It is worthwhile noting that, in the context of image processing, finding efficient numerical methods for computation of $\mathrm{prox}_{\tau\|\cdot\|_{\mathrm{TV}}}$ dates back to the seminal work in [19] that pioneered the field of *TV-denoising*.

Now, having the proximal map (10) defined, it can be shown that the update equation in (8) can be alternatively defined as

$$\phi_{t+\Delta t} = \mathrm{prox}_{(\lambda \Delta t)\|\cdot\|_{\mathrm{TV}}} \left\{ \phi_t - \Delta t \, \mathrm{sign}(\phi_{\mathrm{t}}) \cdot g_N \right\}. \tag{11}$$

Consequently, starting from some initial curve/surface Γ associated with $\phi_0 = \phi_{t=0}$, an optimal solution to the problem of curve/surface fitting *in the sense of minimum cumulative distance* can be computed as a stationary point of the sequence of fixed-point iterations (i.e., recursive re-substitutions) produced by

$$\phi \mapsto T_{\lambda, \Delta t}\{\phi\}, \tag{12}$$

where $T_{\lambda,\Delta t} : \mathcal{X} \to \mathcal{X}$ stands for the composition of the proximal map $\mathrm{prox}_{(\lambda\Delta t)\|\cdot\|_{\mathrm{TV}}}$ with the simple non-linear map defined by $I - \Delta t \mathrm{sign}(\cdot) \cdot g_N$. In practice, we replace g_N with its smooth approximation $g_{N,\sigma}$ obtained via spatial convolution with a normalized, isotropic Gaussian kernel G_σ of standard deviation σ, viz.

$$g_{N,\sigma}(\mathbf{r}) = \left(g_N * G_\sigma\right)(\mathbf{r}) = \frac{1}{N}\sum_{i=1}^{N} G_\sigma(\mathbf{r} - \mathbf{r}_i), \quad \forall \mathbf{r} \in \mathbb{R}^d.$$

In practice, such sums can be computed efficiently, e.g., by first uniformly quantizing the values of $\mathbf{r}_i$ to the vertices of a (sufficiently dense) rectangular lattice, and then performing the summation in the domain of a discrete Fourier transform (DFT), followed by an inverse Fourier transformation.

It is interesting to notice that the map $x \mapsto x - \Delta t \, \mathrm{sign}(x) a$, with $0 \leq a \leq 1$, is neither linear nor monotone, and, what is more important, it does not preserve the sign of x. Thus, applying this map to an SDF ϕ may, in general, forfeit the fundamental property of such ϕ to have values of opposite signs on both sides of its zero-level set. To avoid this undesirable effect, it has proven useful to replace the above map by *soft thresholding* $S_{\Delta t g_{N,\sigma}}(x) := \max(|x| - \Delta t \, g_{N,\sigma}, 0) \cdot \mathrm{sign}(x)$ which, similarly to the former, reduces the positive values of ϕ, while increasing the negative values of ϕ by $\Delta t \, g_{N,\sigma}$. (In some sense, both maps try to "shorten the distance" to points $\mathbf{r}_i$, which are represented by the "peaks" of $g_{N,\sigma}$.) However, $S_{\Delta t g_{N,\sigma}}$ provides the additional advantage of continuity, while keeping the sign of its argument intact.

In practical computations, we used $\Delta t \in (0, 1]$, while adapting σ recursively according to $\sigma \mapsto \beta\sigma$, starting with some initial $\sigma > 0$ and a predefined $0.9 < \beta < 1$. In this case, the proposed numerical scheme acquires the flavor of a multi-grid method. Particularly, in the beginning of iterations (when σ is still relatively large) the intermediate solutions are predominantly effected by a rough collective configuration of the data points as a cloud, while becoming more and more responsive to the individual forces of each point $\mathbf{r}_i$ towards convergence.

To summarize, the final algorithm can be described by the pseudo-code shown below.

```
Data: $\{\mathbf{r}_i\}_{i=1}^N, \phi_0, \lambda, \Delta t, \sigma, \beta$, #iterations
$i = 1$;
$\phi = \phi_0$;
while $i \leq$ #iterations do
    $g_{N,\sigma}(\mathbf{r}) \leftarrow (1/N) \sum_{i=1}^N G_\sigma(\mathbf{r} - \mathbf{r}_i)$;
    $\phi \leftarrow T_{\lambda,\Delta t}\{\phi\}$;
    $\phi \leftarrow \mathrm{ReDist}(\phi)$;
    $\sigma \leftarrow \beta\,\sigma$;
    $i \leftarrow i + 1$
end
Result: $\phi^* = \phi$;
```

Note that, for any value of Δt, $S_{\Delta t g_{N,\sigma}}$ is guaranteed to be a contraction, which, in conjunction with the contractiveness of the proximal map, suggests that the sequence of solutions produced by $\phi \mapsto T_{\lambda,\Delta t}\{\phi\}$ always converges to a stationary point. However, for any given ϕ, its corresponding $T_{\lambda,\Delta t}\{\phi\}$ does not yield an SDF, in general. For this reason, the above pseudo-code has been augmented with the procedure of *redistancing* ReDist, which effectively replaces a given $\phi \in \mathcal{X}$ with an SDF that has a zero-level set Γ identical to that of ϕ[1]. Even thought such mapping may not always be contractive, from the practical point of view, the proposed algorithm has never failed to converge in a relatively small number of iterators (see below), under broad experimental conditions.

4 Experimental Results

The proposed method has been tested on a dataset of both 2D and 3D artificial point clouds. In the course of evaluations, the algorithm's performance was tested under a variety of different experimental conditions, as defined by various levels of measurement noise, the presence of skewness in the distribution of $\{\mathbf{r}_i\}_{i=1}^N$ and various fractions of data points assumed to be missing.

In our experiments, the original geometric manifolds (i.e., curves and surfaces) were used as the mean values around which the cloud points $\mathbf{r}_i$ had then been randomly distributed. To this end, we used a centered uniform distribution in conjunction with an offset to emulate the effect of asymmetric sampling. In all the experiments, ϕ_0 was computed by fast matching initiated at the boundary of the convex hull of its associated data cloud. The values of $\lambda, \Delta t, \sigma$, and β were set to be equal to $100, 0.95, 0.03$, and 0.99, respectively. All coding and simulations were performed in MATLAB (Mathworks Inc, version R2018b).

In the case of symmetric sampling in 2D (i.e., when the number of planar data points is approximately equal on both sides of the original curve), some representative results obtained using the proposed method are shown in Fig. 1,

[1] As mentioned earlier, in numerical computations, redistancing is usually performed by means of fast marching [20].

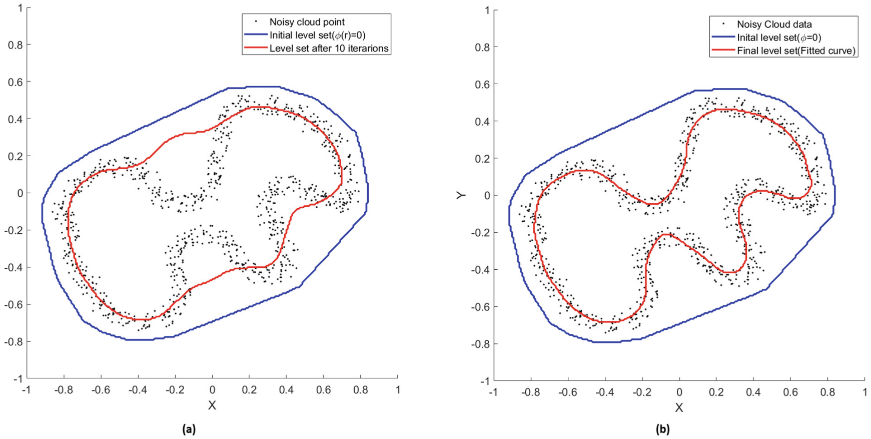

Fig. 1. Results applying the proposed algorithm to 2D "noisy" point clouds. (a) optimal solution after 10 iterations, (b) optimal solution after 30 iterations. The blue and red colors correspond to the initial and final solutions, respectively. (Color figure online)

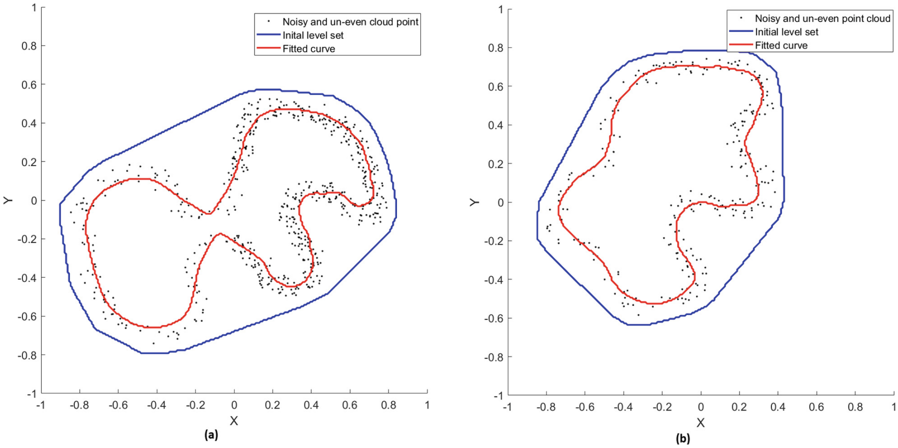

Fig. 2. Results analogous to Fig. 1 obtained in the case of asymmetric data sampling.

with Fig. 2 showing analogous results for the case of asymmetric sampling. One can see that, in all the cases, the proposed algorithm has managed to converge to the expected shapes just in a few tens of iterations.

To quantitatively evaluate the proposed algorithm, a distance between the original and fitted manifolds has been computed. In each case, the distance was defined to be the average distance between the points on the fitted manifold and its continuous counterpart. To facilitate the computation of the distances, the original manifolds were chosen to be

– a circle (radius 50), an oval (max radius 50, axis ratio 1:0.5), and a square (side length 100), in the case of $d = 2$,
– a sphere (radius 50) and an ellipsoid (max radius 50, axis ratio 1:1:0.5), in the case of $d = 3$.

The conditions of asymmetric sampling were emulated by sub-sampling the data points on one size of a given manifold by a factor of 2. Moreover, the data points were further subjected to random removals of their localized subsets to imitate the effects of “holes”. Finally, all data points were contaminated by additive Gaussian noise with different values of its standard deviation σ_n.

Table 1. Empirical mean cumulative errors (average ± 1 standard deviation)

	Circle	Oval	Square	Sphere	Ellipsoid
Number of points N	50	50	50	1200	1200
$\sigma_n = 0.03\,R/L$	1.93 ± 0.81	1.85 ± 1.04	3.41 ± 2.81	3.68 ± 0.86	2.61 ± 0.92
$\sigma_N = 0.10\,R/L$	3.48 ± 1.65	3.19 ± 1.75	5.69 ± 4.24	4.55 ± 0.96	3.21 ± 1.05

Table 1 summarizes the results of present quantitative analysis, showing the values of mean cumulative distance for different types of manifolds as well as the values of N and σ_n. Each entry in Table 1 has been computed as an average of 10 independent trials. As expected, the noisy scenarios with higher extents of skewness and incompletion in measurements results in higher values of the errors. A relative increase of the error can also be observed in the case of square, due to the non-smoothness of its shape. Overall, however, the proposed method demonstrated a stable and reasonably accurate performance. A sample results of surface and curve fitting is shown in Fig. 3.

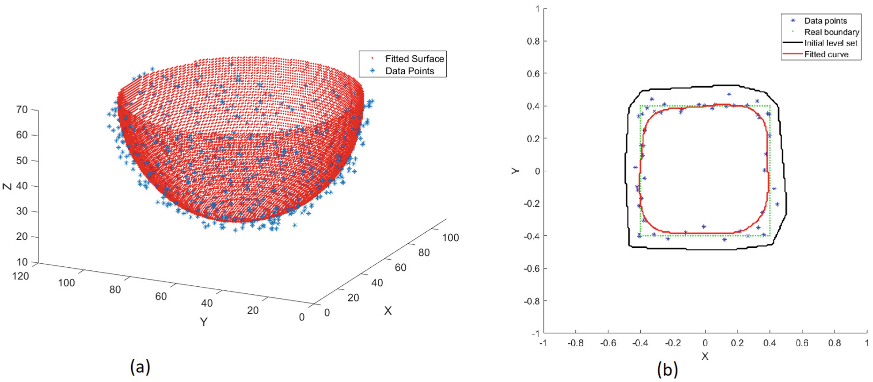

Fig. 3. Results of fitting the sphere surface and the square curves.

5 Conclusion

In this paper, a simple numerical method for fitting closed curves and surfaces of various topologies to points clouds has been presented. The proposed numerical solution was based on implicit discretization, which allows using substantially larger time steps Δt (as opposed to explicit schemes), thus improving the rate of convergence. Moreover, the proposed formulation is parameter-free, offering the same unified algorithmic structure in both 2D and 3D settings.

References

1. Mencl, R., Muller, H.: Interpolation and approximation of surfaces from three-dimensional scattered data points. In: Scientific Visualization, pp. 223–232. IEEE Computer Society (1997)
2. Berger, M., et al.: A survey of surface reconstruction from point clouds. Comput. Graph. Forum **36**(1), 301–329 (2017)
3. Liang, J., Park, F., Zhao, H.: Robust and efficient implicit surface reconstruction for point clouds based on convexified image segmentation. J. Sci. Comput. **54**(3), 577–602 (2013)
4. Poothicottu Jacob, G.: HoloLens framework for augmented reality applications in breast cancer surgery. Master's thesis, University of Waterloo (2018)
5. Franz, A.: Voronoi diagramsa survey of a fundamental geometric data structure. ACM Comput. Surv. (CSUR) **23**(3), 345–405 (1991)
6. De Loera, J.A., Rambau, J., Santos, F.: Triangulations: Structures for Algorithms and Applications. Algorithms and Computation in Mathematics, vol. 25, 1st edn. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-12971-1
7. Carr, J.C., Fright, W.R., Beatson, R.K.: Surface interpolation with radial basis functions for medical imaging. IEEE Trans. Med. Imaging **16**(1), 96–107 (1997)
8. Hoppe, H., DeRose, T., Duchamp, T., McDonald, J., Stuetzle, W.: Surface reconstruction from unorganized points. ACM SIGGRAPH Comput. Graph. **26**(2), 71–78 (1992)
9. Hoppe, H., et al.: Piecewise smooth surface reconstruction. In: Proceedings of the 21st Annual Conference on Computer Graphics and Interactive Techniques, pp. 295–302 (1994)
10. Tang, R., Halim, S., Zulkepli, M.: Surface reconstruction algorithms: review and comparisons. In: The 8th International Symposium on Digital Earth (ISDE) (2013)
11. Kazhdan, M.: Reconstruction of solid models from oriented point sets. In: Proceedings of the Third Eurographics Symposium on Geometry Processing (2005)
12. Kazhdan, M., Hoppe, H.: Screened poisson surface reconstruction. ACM Trans. Graph. (ToG) **32**(3), 29 (2013)
13. Carr, J.C., et al.: Reconstruction and representation of 3D objects with radial basis functions. In: Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques, pp. 67–76 (2001)
14. Khatamian, A., Arabnia, H.R.: Survey on 3D surface reconstruction. J. Inf. Process. Syst. **12**(3), 338 (2016)
15. Caselles, V., Catté, F., Coll, T., Dibos, F.: A geometric model for active contours in image processing. Numerische mathematik **66**(1), 1–31 (1993)
16. Parikh, N., Boyd, S.: Proximal algorithms. Found. Trends® Optim. **1**(3), 127–239 (2014)

17. Chambolle, A.: An algorithm for total variation minimization and applications. J. Math. Imag. Vis. **20**(1–2), 89–97 (2004)
18. https://pythonhosted.org/prox-tv/
19. Rudin, L.I., Osher, S., Fatemi, E.: Nonlinear total variation based noise removal algorithms. Physica D Nonlinear Phenom. **60**(1–4), 259–268 (1992)
20. Sethian, J.: Level Set Methods. Cambridge University Press, Cambridge (1996)