A Robust End-to-end Method for Parametric Curve Tracing via Soft Cosine-similarity-based Objective Function

Boran Han, Jeremy Vila
Shell International Exploration and Production Inc.
2 Drydock Ave., Boston MA, 02210
Boran.Han@shell.com, Jeremy.Vila@shell.com

Abstract

Parametric curve tracing enables wide applications,such as lane following in autonomous driving, volumetric reconstruction in seismic, single-molecule/protein tracking in microscopy. Most existing parametric curve tracing methods require several steps, including curve identification and parameterization. Such multi-step methods can lead to lengthy and complicated parameter optimization. Additionally, the performance of curve identification methods can be degraded by noisy or low-light images. To address these challenges, we present a novel single-step approach to trace curves parametrically via optimizing a self-defined non-linear objective function that describes several key properties of the curve. Under the assumption that signals along the curve resemble each other, our objective function will guide this pathfinding process from a seed point along the direction according to maximum cosine similarity. No pre- and post-processing step is required to measure the tangent or normal vectors. We visualize our objective function and conduct several numerical experiments. These empirical experiments demonstrate that our method outperforms other competing methods across image domains. It yields better accuracy even in low signal-to-noise ratio (SNR) conditions.

1. Introduction

Parametric curve tracing is prevalent in image processing and computer vision applications. In contrast to edge or ridge detection, a curve parametrization provides the mathematical description of a path ($\boldsymbol{c}(t)$) in an image, represented as:

$$\boldsymbol{c}(t) = (X(t), Y(t)) \tag{1}$$

where $\boldsymbol{c}(t) \in \mathbb{R}^2$ and $t \in \mathbb{R}$. Information provided after parametrization, such as the tangent vector and curvature, can be especially useful in lane or coastline following, microscopic and seismic feature analysis. However, two significant challenges remain for pre-existing parametric curve tracing methods (Fig. 1): (1) Curve parameterization, which necessitates marking and grouping pixels of the target, and (2) Curve identification in the presence of wide variety of illuminations, acutance and SNR.

One conventional way to identify curves of interest is to detect the local extremum in an image [44] or its corresponding differentiated form [7, 41, 42, 53], followed by a pixel-level segmentation method [9, 22, 26, 28, 57]. However, each method has its trade-off. For instance, some can be highly sensitive to the noise [44, 53] while others are dependent on the choice of hyperparameters [7]. A potentially promising approach to identify the curve in noisy images is to first denoise the image using denoising methods, such as median filter [27], Non-Local Means (NLM) [6] and Block-matching and 3D filtering (BM3D) [14]. However, these methods can blur faint high-frequency signals, making detection even more challenging (Fig. 1d). Another class of curve identification algorithms employs deep learning, particularly deep convolutional networks (DCN) [10, 29, 48, 50]. However, a trained network is usually only predictive in one imaging domain. Additionally, these approaches require extensive and highly-curated datasets to perform robustly in the face of SNR variation.

Curve parametrization, finding parametric equations of a curve defined in Eq. (1), is another key process in parametric tracing. A handful of methods have been developed for curve parametrization, including regression in the presence of outliers [13, 19, 20] and Spline interpolation techniques [24]. Their performance can be sensitive to signal and noise properties. Moreover, in the existing workflow, those properties are directly affected by the outputs from curve identification methods, making the parameter-tuning process lengthy and complicated, posing an obstacle for robust and convenient parametric tracing. Apart from the lengthy parameter-tuning, a

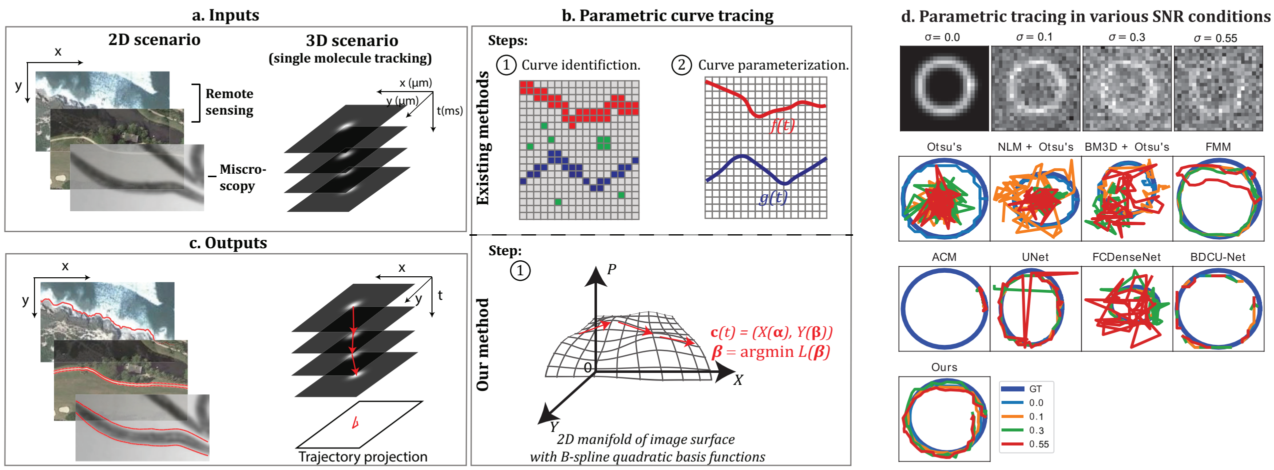

Figure 1. Illustration for curve tracing using existing methods and our method. (a) A few examples of inputs. (b) Illustration of typical existing methods (upper panel) and our method (lower panel) for parametric tracing. In the curve identification step of the existing methods, two curves are colored in red and blue. Small clusters, identified as the noise, are all colored in green. (c) Ideal detected curves with parameterized representation (marked in red) as outputs (red line) overlaid with inputs. (d) An example showing our method outperforms in low SNR. Images are normalized to 1.

failure to detect traces in the first step can also lead to incorrect parametric tracing regardless of any advanced parameterization method being used. Therefore, despite rich studies in the field, robust parametric curve tracing remains difficult.

The problem formulation. To overcome those two limitations, we develop a single-step approach to parametrically trace a curve from a starting seed point. The curve, denoted as $\boldsymbol{c}(t)$, is required to be continuous at any point within the its domain T:

$$\lim_{t \to t_0} \boldsymbol{c}(t) = \boldsymbol{c}(t_0), \forall t_0 \in T \tag{2}$$

The signals along the $\boldsymbol{c}(t)$ must present similarity with regard to pixel intensity. However, the curve is not necessarily differentiable (or smooth) within T:

$$\lim_{t \to t_0^-} \boldsymbol{c}(t)' \not\equiv \lim_{t \to t_0^+} \boldsymbol{c}(t_0)' \tag{3}$$

Given that signal similarity along the curve of target is not a necessary nor sufficient condition to describe edges or ridges accurately, our method can apply to some but *not all* edge or ridge detection problems. More practical examples that our method can solve are shown in Fig. 1, and will also be elaborated in Sec. 5. To summarize, **our contributions** are as follows:

- We propose a single-step method specializing in parameterizable coherent signal tracing as a substitute for the existing multi-step procedure.
- We discuss the continuity, differentiability, and convexity of our objective function systemically via visualization.
- We demonstrate that our method outperforms competing approaches under various SNR or lighting conditions across several application domains.

2. Related Works

Local gradient or extremum detection methods. The standard curve tracing algorithms usually start from detecting pixels with the high local gradient or with local extremum and then grouping the selected pixels into clusters. Otsu's method [44] is one of the most widely used parameter-free local extremum detection methods. Sobel [53] and Canny filtering [7] are two popular methods to calculate the local gradient. Several edge detection methods in low SNR have also been developed recently [41, 56]. After the edge detection, pixel segmentation is then performed. Such methods include contour-based labeling [9], pixel clustering [47] and block-based connected-component labeling [22, 28].

Nonlinear optimization methods used for curve detection. One popular algorithm used for curve tracing is the geodesic method on Riemannian manifolds [34, 45], following salient curvilinear structures in the local domain. This PDE-based algorithm enables curve tracing by connecting two seed points by their geodesic path. To calculate geodesics distance between two seed points, the fast marching method (FMM), a special case of level set methods, was developed by Osher and Sethian [43, 51]. Initially proposed to solve the Eikonal Equation, a PDE problem in wave propagation, FMM assumes that information only propagates outwardly from the seed. The wave propagation given by the FMM represents a distance function that corresponds to the geodesic distance measured with the metric defined by users. It has been applied to satellite images [38, 45] and medical image [36, 49].

An alternative approach is the active contour model (ACM) [32] and its variants [8, 33, 58]. However, those derivative methods are used in closed contour detection, such as vector flow [58] or geometric methods [8, 33]. In contrast, our approach has no limitation on types of contour. Among the ACM method class, the original ACM proposed by Kass *et al.* [32] can be applied in both open and closed contour (or curve) detection. Therefore, in this paper, we only consider ACM [32] in the context of general contour detection (open and closed) for comparison. Similar to FMM, ACM [32] requires two seed points for curve detection. Inspired by these work, we here propose a new end-to-end parametric tracing approach with one seed point required.

DCN. An emerging alternative approach is to employ a neural network that automatically learns the mapping from images to their annotated counterparts. Curve identifications can be achieved by training segmentation models [10, 29, 48, 50]. Although there exists learning-based and physics-based methods outputting parametric features [21, 37], they predict control points and then perform the Spline interpolation between control points. Therefore, they lack the capability to follow the curves accurately between control points, especially when curves that are not smooth (Eq. (3)), which indicates that their designed application domain can be different from ours. Moreover, DCN-based methods require a large training dataset in a particular imaging domain that sufficiently captures nuisance variation.

3. Methodology

To give an overview, our method first uses piece-wise polynomial model to describe the parametric curve that has an arbitrary behavior. Secondly, the polynomial parameters of each segment can be acquired by minimizing our objective function L which consists of up to three terms describing signal similarity L_{corr}, regularization L_{reg} and internal continuity L_{cont}:

$$L = L_{\text{corr}} + \lambda_1 L_{\text{reg}} + \lambda_2 L_{\text{cont}} \tag{4}$$

where $\lambda_1 \in [0, 1]$ and $\lambda_2 \in [0, 1]$ are the weights of L_{reg} and L_{cont}. In this section, we first explain our spline-like model for the parametric curve tracing and the parameters to be optimized (Sec. 3.1). We then introduce the cosine similarity term L_{corr} in the objective function (Sec. 3.2) and its differentiable form (Sec. 3.3). Next, we present L_{reg} and L_{cont} penalizing overfitting and preserving the internal curve continuity if needed (Sec. 3.4).

3.1. Parametric tracing model

We adopt the piece-wise polynomial model as the parametric curve: On each of the half-open interval, each curve segment agrees with a polynomial of degree N. Accordingly, $\boldsymbol{c}(t)$ in Eq. (1) can then be written as:

$$\begin{aligned} \boldsymbol{c}(t) &= (X_k(\boldsymbol{\alpha}_k, t), Y_k(\boldsymbol{\beta}_k, t)) \\ &= (\sum_{n=0}^{N} \alpha_{n,k} t^n, \sum_{n=0}^{N} \beta_{n,k} t^n), k\Delta t \le t < (k+1)\Delta t \end{aligned} \tag{5}$$

with $k \in \mathbb{N}$ presenting the k-th knot and knot vector $\boldsymbol{t} = (0, \Delta t, \ldots, k\Delta t, \ldots \text{and} K\Delta t)$ containing K equidistantly distributed along $(0, K\Delta t)$. X_k and Y_k are the polynomial functions at interval $[k\Delta t, (k+1)\Delta t)$. $(\boldsymbol{\alpha}_k, \boldsymbol{\beta}_k) \in \mathbb{R}^N$ are the polynomial coefficients of curve X_k and Y_k respectively. When the curve of interest follows the behaviors of a polynomial, $K = 1$. Otherwise, $K > 1$. Therefore, the parameters to be optimized in this model are $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ at each interval.

We also require our model to be continuous across knots:

$$\lim_{t \to k\Delta t^-} \boldsymbol{c}(t) = \lim_{t \to k\Delta t^+} \boldsymbol{c}(t), k = 0, 1, \ldots, K \tag{6}$$

Given that $X(\boldsymbol{\alpha})$ and $Y(\boldsymbol{\beta})$ are two orthogonal axes and can be exchanged through linear transformation, $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are therefore independent and symmetric. Hence, for the ease of notation and reduce the dimensionality in visualization, we discuss how to obtain $\boldsymbol{\beta}$ in the following sections.

3.2. Cosine similarity term for parametric curve tracing

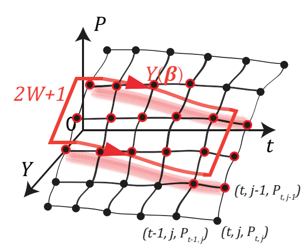

Figure 2. Illustration for parametric curve tracing on a 2D manifold using cosine similarity-based objective term. The three axes are pixel intensity P, t and Y axis in the image. A window moves along one dimension (t axis) to calculate the cosine similarity between pixel-intensity vectors from two adjacent columns. Black dots with red circles are the selected pixels within the window slot at positions $(t, \lfloor Y(\boldsymbol{\beta}) \rfloor + w)$ where $w \in [-W, \ldots, W]$. We note that axis X, which is perpendicular to Y and t, is not drawn in this figure for convenience.

In order to find $\boldsymbol{\beta}$, we defined an objective function within each interval as:

$$L_{\text{corr}} \triangleq \frac{\sum_{t=0}^{\Delta t} (f_{\text{corr}}(\boldsymbol{P}_t(\lfloor Y(\boldsymbol{\beta}) \rfloor), \boldsymbol{P}_{t+1}(\lfloor Y(\boldsymbol{\beta}) \rfloor))}{\Delta t} \tag{7}$$

where $L_{\text{corr}} \in \mathbb{R}$, $\boldsymbol{P} \in \mathbb{R}^{2W+1}$, and $P_{t,j}$ corresponds to the pixel intensity at the coordinate (t, j) within the image. $\boldsymbol{P}_t$ can be written as:

$$\boldsymbol{P}_t = \begin{bmatrix} P_{t,\lfloor Y(\boldsymbol{\beta})\rfloor - W} \\ P_{t,\lfloor Y(\boldsymbol{\beta})\rfloor - W+1} \\ \vdots \\ P_{t,\lfloor Y(\boldsymbol{\beta})\rfloor + W} \end{bmatrix} \tag{8}$$

where $j = \lfloor Y(\boldsymbol{\beta})\rfloor$, $\lfloor Y(\boldsymbol{\beta})\rfloor$ donate the flooring function of $Y(\boldsymbol{\beta})$, and W is a constant which represents the window slot sliding along the X axis, as shown in Fig. 2. f_{corr}: $\mathbb{R}^{2W+1} \to \mathbb{R}$ in Eq. (7) measures the cosine similarity between $\boldsymbol{P}_t$ and $\boldsymbol{P}_{t+1}$:

$$f_{\text{corr}}(\boldsymbol{P}_t, \boldsymbol{P}_{t+1}) = 1 - \frac{\boldsymbol{P}_t \cdot \boldsymbol{P}_{t+1}}{\|\boldsymbol{P}_t\| \|\boldsymbol{P}_{t+1}\|} \tag{9}$$

Unlike other methods which detect the local gradient, we treat pixel intensities within $2W+1$ as a $2W+1$-dimensional vector. We then measure the cosine similarity function between adjacent t.

The similarity term L_{corr} (Eq. (7)), composed from cosine simlarity function (Eq. (9)), depicts that an approach will trace outwardly along the direction according to the maximum cosine similarity, with an assumption of a polynomial function in each interval. However, the inner product between two vectors will make this objective function a nonlinear coupled equation that cannot be solved easily in a general case. In addition, despite that cosine similarity function can be continuous when measuring multivariate distribution, our equation consists of $\boldsymbol{P}_t$ and $\boldsymbol{P}_{t+1}$, which are in discrete space as its indexing requires integers: $t, \lfloor Y(\boldsymbol{\beta})\rfloor \in \mathbb{N}$. Such a factor indicates that Eq. (9) is not differentiable (See mathematical proof in Supplementary Section 1).

3.3. Differentiable cosine similarity term for parametric curve tracing

We hereby create a soft cosine similarity term to update the gradients when optimizing our objective function via gradient descent-based algorithms. We achieve this goal by interpolating the pixel vector using B-spline basis functions. The B-spline basis is a basis function that is widely used in computer graphics, image processing [30, 54] and deep learning [4, 5, 16, 18]. Inspired by those works, we interpolate the pixel vector in $\mathbb{R}^{2W+1}$ using B-spline kernel. Therefore, Eq. (7) can be rewritten as:

$$L_{\text{corr}} = \frac{\sum_{t=0}^{\Delta t}(f_{\text{corr}}(\boldsymbol{P}_t(Y(\boldsymbol{\beta})), \boldsymbol{P}_{t+1}(Y(\boldsymbol{\beta})))}{\Delta t} \tag{10}$$

$$\boldsymbol{P}_t(Y(\boldsymbol{\beta})) = \boldsymbol{P}_{t,\lfloor Y(\boldsymbol{\beta})\rfloor} * \boldsymbol{N}_{\Omega} \tag{11}$$

$$= \sum_{j=0}^{2W+1} P_{t,j} N_{j,\Omega}(Y(\boldsymbol{\beta})) \tag{12}$$

where the modified $\boldsymbol{P}_t(Y(\boldsymbol{\beta}))$ is equal to the discrete $\boldsymbol{P}_{t,\lfloor Y(\boldsymbol{\beta})\rfloor}$ convoluted by Ω-degree B-spline basis kernel ($\boldsymbol{N}_{\Omega}$), $*$ denotes the convolution operator, j is the knot value in the knots vector $\boldsymbol{J} = [0, 1, \ldots, j, \ldots, 2W+1]$. In this soft cosine similarity term, new $\boldsymbol{P}_t(Y(\boldsymbol{\beta}))$ is reconstructed by the linear combination of B-Spline basis vectors in the domain of $Y(\boldsymbol{\beta}) \in \mathbb{R}$. Such process maps the discretized domain $\boldsymbol{J} \in \mathbb{N}^{2W+1}$ into continuous domain $Y(\boldsymbol{\beta}) \in \mathbb{R}^{2W+1}$. Based on the recursive property of B-Spline [15, 46], the derivative of Ω-degree B-spline is a function of $\Omega - 1$-degree B-spline [15]. Therefore, we calculate the gradients of $\boldsymbol{P}_t(Y(\boldsymbol{\beta}))$ with respective to $\boldsymbol{\beta}$, giving

$$\frac{\partial \boldsymbol{P}_t(Y(\boldsymbol{\beta}))}{\partial \boldsymbol{\beta}} = \sum_{j=0}^{2W+1} (P_{t,j+1} - P_{t,j}) N_{j,\Omega-1}(Y(\boldsymbol{\beta})) \tag{13}$$

$\sum_{j=0}^{2W+1} P_{t,j} N_{j,\Omega-1}(Y(\boldsymbol{\beta}))$, equivalent as interpolating via $\Omega - 1$-degree B-spline, is a continuous function when $\Omega \geq 3$. According to the chain rule and proof in Supplementary (Suppl.) Sec.1, when $\frac{\partial \boldsymbol{P}_t(Y(\boldsymbol{\beta}))}{\partial \boldsymbol{\beta}}$ is differentiable, $\frac{\partial L_{\text{corr}}}{\partial \boldsymbol{\beta}}$ will be differentiable.

3.4. Penalty terms

Regularization on higher order coefficients. To prevent over-fitting caused by the higher-order weights, we added regularization term L_{reg}, penalizing the higher order coefficients:

$$L_{\text{reg}} = \sum_{n=2}^{N} \beta_n^2 \tag{14}$$

Curvature continuity. We note that curvature continuity term is not essential or necessary for all applications since our defined curve do not need to be smooth (Eq. (3)). One application without the smoothness requirement is illustrated in Sec. 5. However, in applications where smoothness is required, we defined a penalty term L_{cont} to largely maintain the continuity of traced curve when $K > 1$. The tangent vector $\boldsymbol{T}(t)$ and the curvature $\kappa(t)$ need to be continuous, Therefore, we obtain our continuity term L_{cont} as:

$$\begin{aligned} L_{\text{cont}} = &\left| \sum_{n=2}^{N} \beta_{n,k-1}(k\Delta t)^{n-2} - \sum_{n=2}^{N} \beta_{n,k}(k\Delta t)^{n-2} \right| \\ &+ \left| \sum_{n=1}^{N} \beta_{n,k-1}(k\Delta t)^{n-1}) - \sum_{n=1}^{N} \beta_{n,k}(k\Delta t)^{n-1}) \right| = 0 \end{aligned} \tag{15}$$

More details to obtain Eq. (15) are presented in Suppl. Sec. 2. Therefore, our objective function becomes:

$$L = L_{\text{corr}} + \lambda_1 L_{\text{reg}}, \text{ s.t. } L_{\text{cont}} = 0 \tag{16}$$

To simplify the optimization, we add L_{cont} to the L ($L = L_{\text{corr}} + \lambda_1 L_{\text{reg}} + \lambda_2 L_{\text{cont}}$) as an unconstrained minimization problem. Therefore, given the fact that $\boldsymbol{\beta}$ is a variable in objective function L, best parameters $\boldsymbol{\beta}$ can be obtained by minimizing the objective function:

$$\boldsymbol{\beta} = \arg\min L \tag{17}$$

Our objective with continuity term describes a parametric curve that optimizes the balance between signal similarity along its tangent direction and the continuity of the first- and second-order derivative across knots. The spline-based representation on objective function guarantees its differentiability for the inconvenience of non-linear optimization. Our unique objective function differentiates our method from traditional spline fitting methods, even though both methods employ the piece-wise polynomials as a parametric curve model: In our study, we loosen the restriction on the continuity of first- and second-order derivative but introduce the cosine similarity term, which enables the action of tracing. The summarized algorithm to obtain $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ per curve is shown in Alg. 1. When multiple curves of interest exist together, each of the parametric curve tracing follows the same method (Alg. 1).

Algorithm 1 Parametric curve tracing within Δt

```
Input: Given image U, window slot W, penalty function weight λ, maximum iteration E, degree of B-spline basis function Ω, degree of polynomial N, seed point β^(0)[0] and α^(0)[0].
Initialize: α^(0)[1 : N] = 0 and β^(0)[1 : N] = 0
while e < E do
    Compute L_corr^(e) from:
    while t ∈ U do
        Y(β^(e)) = Σ_{n=0}^N β_n^(e) t^n, X(α^(e)) = Σ_{n=0}^N α_n^(e) t^n
        L_corr^(e) ← L_corr^(e) + f_corr(P_t(⌊Y(β^(e))⌋, ⌊X(α^(e))⌋) ∗ N_Ω, P_{t+1}(⌊Y(β^(e))⌋, ⌊X(α^(e))⌋) ∗ N_Ω)
        t ← t + 1
    end while
    Compute the objective value: L^(e) = L_corr^(e) + λ_1 L_reg^(e) + λ_2 L_cont^(e), L^(e) ∈ ℝ
    Compute gradients L^(e)/∂β, L^(e)/∂α = ∇L^(e) ∈ ℝ^{2N} [1]
    Update β^(e+1) and α^(e+1) via optimizer.
    e ← e + 1
end while
Obtain the final α, β = argmin_{α,β} L
```

A special case The aforementioned t represents the temporal information (T) in a video ($X-Y-T$) or third axis (Z) in a 3D image stack ($X-Y-Z$), as shown in Fig. 1. It is worth mentioning that our method is also capable of tracing curves in static 2D images (X-Y) where temporal information is missing. However, It is limited to the condition that $\boldsymbol{c}(t)$ is expanding outwardly in one axis (X or Y) so that within half-open interval $[k\Delta t, (k+1)\Delta t)$, Eq. (5) can be simplified as:

$$\boldsymbol{c}(t) = \big(t, Y(\boldsymbol{\beta_k}, t)\big) = \sum_{n=0}^{N} \beta_{n,k} x^n \tag{18}$$

which is equivalent as setting $\alpha_0 = 1$ whereas $\alpha_n = 0 (n > 0)$. The method to obtain $\boldsymbol{\beta}$ remains the same as above.

4. Objective function discussion

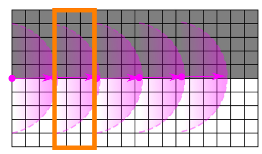

Figure 3. Illustrator of simulation and curve tracing.

As a working objective function that can be optimized using gradient-based methods, the objective L needs to be continuous and differentiable w.ith respect to the parameters to be optimized, $\boldsymbol{\beta}$ and $\boldsymbol{\alpha}$. Without loss of generality, we validate these properties through visualizing the $L(\boldsymbol{\beta})$ via a simple simulation.

Simulation. As shown in Fig. 3, we simulated a horizontal edge across a grid of 20×20 pixels. The horizontal edge splits the grid into two. To plot the objective function in response to $\boldsymbol{\beta}$ during optimization, we set $\Delta t = 2$ pixels, $W = 5$ pixels and $N = 2$. (β_1, β_2) are chosen evenly distributed from -1.5 to 1.5.

Continuity and differentiability of the objective function. We scrutinize three types of objective functions we proposed here: L_{corr}, $L_{\text{corr}} + L_{\text{reg}}$ and $L_{\text{corr}} + 0.1 L_{\text{cont}}$ boxed in Fig. 4. We found that when no B-spline kernel was being added ($\Omega = 0$), all three objective function surfaces behave as step functions. When $\Omega = 1$, *i.e.* linear B-spline basis, objective function surface is continuous but not differentiable across knots [15]: Eq. (13) becomes $\sum_{j=0}^{2W+1} (P_{t,j+1} - P_{t,j}) N_{j,0}(Y(\boldsymbol{\beta}))$. $N_{j,0}(Y(\boldsymbol{\beta})) = 1$ between jth and $j+1$th knot and $N_{j,0}(Y(\boldsymbol{\beta})) = 0$ elsewhere. Therefore:

$$\frac{\partial \boldsymbol{P}_t(Y(\boldsymbol{\beta}))}{\partial \boldsymbol{\beta}} = \sum_{j=0}^{2W+1} (P_{t,j+1} - P_{t,j}) \text{rect}(Y(\boldsymbol{\beta})) \tag{19}$$

where $\text{rect}(Y(\boldsymbol{\beta}))$ is the rectangular function, satisfying:

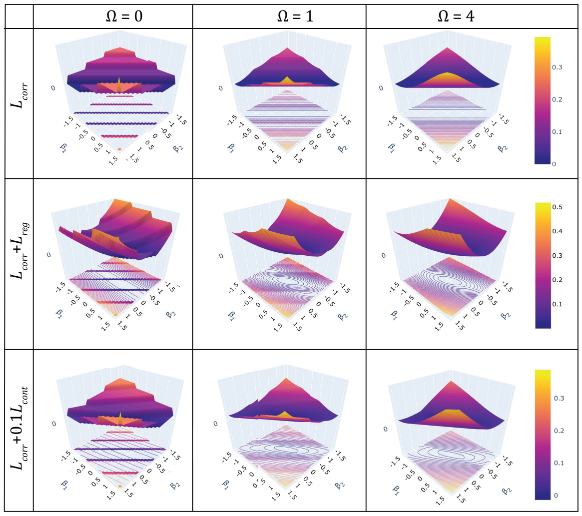

Figure 4. Objective function surface visualization ($\Delta t = 2$).

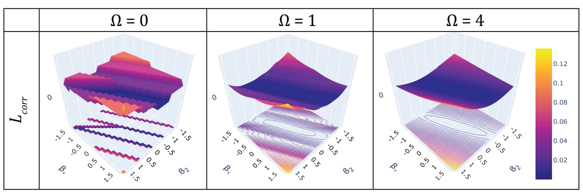

Figure 5. Objective function surface visualization of L_{corr} when $\Delta t = 4$.

$$\text{rect}(j) = \begin{cases} 1 & j < Y(\boldsymbol{\beta}) \leq j+1 \\ 0 & \text{otherwise} \end{cases} \tag{20}$$

It explains the objective function surface (shown in Fig. 4) is not as smooth when $\Omega = 1$. By contrast, when $\Omega = 4$, *i.e.* quadratic basis function, all three objective function surfaces are smoothed out compared to $\Omega = 0$ and $\Omega = 1$.

Convexity of the objective function. Furthermore, we notice that when no penalty functions are added ($\Omega = 4$, L_{corr}), a series of $\boldsymbol{\beta}$ are found for minimum L_{corr} owing to the problem that several sets pf parameters $\boldsymbol{\beta}$ can correspond to the same minimum $Y(\boldsymbol{\beta})$. This problem can be solved by increasing Δt, as shown in Fig. 5. Another alternative solution to be add L_{reg} and L_{cont} , which will perform as constrained conditions to prevent too many possible solutions, which is demonstrated in Fig. 4. Despite displaying convexity through this simple simulation, our objective function varies with images and may not be convex in most cases.

Robustness The noise tolerance study is presented in Suppl. Section 4

5. Numerical experiments

5.1. Dataset

We tested our methods in various practical scenarios where parameterization is critical. Among them, we use single protein/molecule tracking (SPT/SMT) to demonstrate the capability of tracing along the temporal axis (X-Y-T, Eq. (5)), whereas the rest are static 2D images (X-Y, Eq. (18)).

SPT/SMT. SPT/SMT, which is a process of finding the trajectory of a fluorescent emitter (*i.e.* protein/molecule) in a diffraction-limited system, has become an important tool to study biological problems, since it allows direct observation of protein dynamics. Parametric tracing can be especially essential in SPT/SMT, providing kinetic information such as trajectories, velocity and acceleration of individual molecule. Furthermore, SPT/SMT traces emitters, which often emit a few photons, through an EM-CCD camera that detects single-photon events. Such a high sensitivity often leads to low SNR. In spite of intensive studies focusing on kinetic classification via deep learning after tracing, including [2, 55], only a few methods are proposed for tracing itself, such as UNet-based detection [25] and statistical method [12]. To test our method's performance, we use the synthetic SPT/SMT dataset, a benchmark method to test SPT/SMT algorithms [12]. 1000 traces, resembling random walk, are employed with diffusion coefficient as $D = 3\mu m/s^2$; Each emitter, or molecule, is convoluted with point spread function [39] stemming from optical diffract limit. Different amplitudes of Gaussian noise (σ = 0.0, 0.1, 0.3 and 0.55) are added to test the robustness of our method under low SNR.

Highway dataset. We perform parametric road following on Jiqing Expressway Dataset[1]. Lanes were annotated by a combination of segment-wise polynomial fitting and manual labeling. Since our goal is to test our method under more challenging conditions, we collected the frames recorded in the tunnels where various lighting conditions are observed. Our testing set comprises more than 1000 frames from several tunnels. Faint lanes are commonly observed due to shadows cast by other cars, low-lighting, or strong reflection from the ground.

Other datasets We also employ seismic images (Suppl. Section 7), remote sensing images [11] (Unlabeled) and microscopy images [35] (Unlabeled).

5.2. Experiment settings

$\Omega = 4$ and RMSprop optimization method are used in all experiments. **Competing methods** for comparison, predominately three types, are listed as follows:

[1]https://github.com/vonsj0210/Multi-Lane-Detection-Dataset-with-Ground-Truth

Conventional methods: Non-deep methods involve classical Otsu's method [44] for all experiments. Denoising methods, such as NLM [6] and BM3D [14], are implemented when testing Otsu's method since Otsu's method is not designed for noisy images. The state-of-art method reported for low SNR conditions, Faint Curved Edge Detection (FCED) [41] is also tested for some experiments.

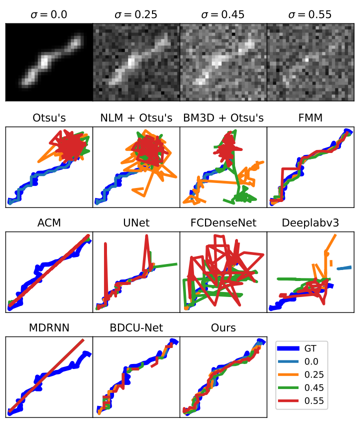

Figure 6. Visual comparison of SPT/SMT. Projected trajectories into X-Y plane are shown with different amplitudes of noise being tested. Note that images are normalized to 1.

FMM and ACM: FMM used here is implemented following reference [45]. Isotropic metric is defined according to pixel intensity f: $W(x) = \frac{1}{\epsilon+|f(x_0)-f(x)|}$ where x_0 is the starting seed. The learning rate $\tau = 0.8$ for all experiments. Specific values of ϵ are provided for each experiment. We apply original ACM [32] which has no restriction on closed contour. In all experiment, weights for continuity term and smoothness term are $\alpha = 0.1$ and $\beta = 1.0$. Two seed points required by FMM and ACM are taken from the ground truth.

DCN: Deep segmentation learning methods include UNet [48], FC-DenseNet [29] and DeepLabv3-MobileNetv2 [10, 50]. For SPT/SMT, we also added Multi-Dimensional Recurrent Networks (MDRNN) [23] and a convolutional-LSTM-based method [52]: Bi-Directional ConvLSTM U-Net (BDCU-Net) [3]. For each experiment, deep learning models are trained from scratch using each of the corresponding dataset. Both low and high SNR images are included in the training.

5.3. Metric

We defined three metrics to quantify the accuracy of all the methods: detected length percentage $\mathscr{L}$, distance between curves $\mathscr{D}$ and distance between tangent vectors $\mathscr{V}$:

$$\mathscr{L} = \frac{|T_{\text{ground truth}} \cap T|}{|T_{\text{ground truth}}|} \tag{21}$$

$$\mathscr{D} = \|\boldsymbol{c}(t) - \boldsymbol{c}_{\text{ground truth}}(t)\|, \forall t \in T_{\text{ground truth}} \tag{22}$$

$$\mathscr{V} = \|\boldsymbol{c}'(t) - \boldsymbol{c}'_{\text{ground truth}}(t)\|, \forall t \in T_{\text{ground truth}} \tag{23}$$

where $T_{\text{ground truth}}$ and T are the sets of X coordinates in the ground-truth image and output image, $|T|$ denotes the number of elements in set T and $\boldsymbol{c}'(t)$ is the differentiation of path $\boldsymbol{c}(t)$ with respect to t. $\mathscr{L}$ describes how capable a method can recognize the curve. $\mathscr{D}$ and $\mathscr{V}$ indicate how accurately a method can trace the curves and measure the tangent vector respectively. Because start and endpoints are given in FMM and ACM, so $\mathscr{L}$ will not be included for comparison.

5.4. Results

SPT/SMT. Results and visual comparison are shown in Tab. 1 and Fig. 6, respectively. All methods show comparable results when $\sigma = 0.0$. However, when it comes to low SNR tracing, our method shows 1 - 15% accuracy improvement in $\mathscr{D}$ and $\mathscr{V}$. Among all competing methods, UNet [48] and its improved model, BDCU-Net [3], have better accuracy. MDRNN's [23] poor performance is attributed to the frame-to-frame random walk behaviors. It is worth noting that despite that $\mathscr{L}$ are close to 1 across most methods in various SNR, $\mathscr{D}$ and $\mathscr{V}$ degrade significantly with noise. Such results indicate that those methods are mistakenly tracing the noise instead of signals.

Lane following. As shown in Fig. 7 and Tab. 2, result provided from our method are among the best in terms of accuracy in various lighting conditions. $\mathscr{L}$ values are significantly smaller compared with those from SPT/SMT. This seeming contradiction is stemming from the much more nonuniform distribution of noise and lighting across the image. Such heterogeneity poses a challenge for competing methods to segment accurately. Faint lane separation lines, for instance, are often observed near the horizon.

Ablation study. The results using different sets of the parameter are listed in Suppl. Section 3.

6. Discussion and Conclusion

We present a novel end-to-end parametric curve tracing method and demonstrate that our approach delivers good results under low SNR conditions. We note that our proposed approach has several advantages as well as limitations.

Advantages. Lesser performance of existing methods is partially attributed to their multi-step procedure. This shared drawback will lead to not only a sophisticated parameter tuning process, but also a failure to consider detailed features such as ridges and valleys during parameterization, since such information is replaced by mask after curve identification or segmentation step.

Noise level	$\sigma = 0$			$\sigma = 0.1$			$\sigma = 0.3$			$\sigma = 0.55$		
Methods	$\mathscr{L}$	$\mathscr{D}$	$\mathscr{V}$	$\mathscr{L}$	$\mathscr{D}$	$\mathscr{V}$	$\mathscr{L}$	$\mathscr{D}$	$\mathscr{V}$	$\mathscr{L}$	$\mathscr{D}$	$\mathscr{V}$
NLM [6] + Otsu's [44]	-	-	-	**1.0**	2.635	2.350	**1.0**	3.741	2.344	**1.0**	4.281	2.484
BM3D [14] + Otsu's [44]	-	-	-	**1.0**	2.743	1.929	**1.0**	3.160	2.096	**1.0**	5.086	2.644
FMM [45] $\epsilon = 0.01$	-	0.482	0.696	-	0.535	0.711	-	0.691	0.747	-	0.849	0.773
FMM [45] $\epsilon = 1$	-	0.484	0.700	-	0.517	0.708	-	0.644	0.737	-	0.813	0.781
ACM [32]	-	2.180	1.291	-	2.182	3.274	-	2.184	3.558	-	2.296	3.275
UNet [48]	**1.0**	0.303	0.653	0.999	0.411	0.653	0.995	0.558	0.705	0.996	0.832	1.572
FC-DenseNet [29]	**1.0**	0.260	**0.562**	**1.0**	0.415	0.577	**1.0**	1.434	1.878	**1.0**	5.060	4.163
DeepLabv3 + MobileNetv2 [10, 50]	**1.0**	0.348	0.650	0.992	0.629	0.648	0.968	1.538	1.252	0.903	2.004	1.941
MDRNN [23]	**1.0**	5.630	0.990	**1.0**	5.594	3.218	**1.0**	5.300	2.955	**1.0**	5.611	3.437
BDCU-Net [3]	0.814	0.256	0.536	0.854	**0.398**	0.577	0.873	0.560	0.675	0.836	0.732	0.735
Ours	**1.0**	0.238	0.570	**1.0**	0.402	**0.529**	**1.0**	**0.537**	**0.590**	**1.0**	**0.719**	**0.623**

Table 1. Evaluation results of SPT/SMT. Best in bold.

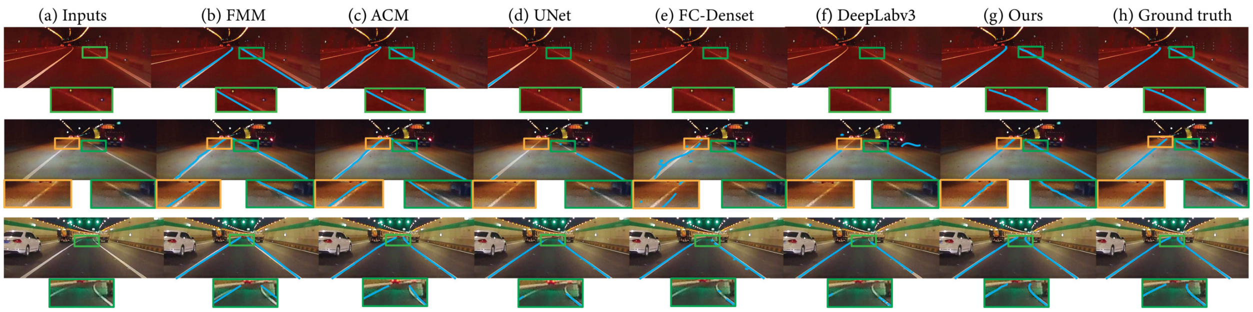

Figure 7. Visual comparison of a few examples under various lighting conditions, including bright and dim lighting: (a) Inputs, (b) FMM ($\epsilon = 0.01$), (c) ACM, (d) UNet, (e) FC-Densenet, (f) DeepLabv3, (g) Our method and (h) Ground truth. (Best viewed with zoom). The segmentation results from DCN methods are shown in Supple. Fig. 2. (Best viewed with zoom)

Noise level	Bright light $\sigma_{est} = 0.136$ [17]			Low light $\sigma_{est} = 0.335$ [17]		
Methods	$\mathscr{L}$	$\mathscr{D}$	$\mathscr{V}$	$\mathscr{L}$	$\mathscr{D}$	$\mathscr{V}$
NLM [6] + Otsu's [44]	-	-	-	0.424	4.642	0.332
BM3D [14] + Otsu's [44]	-	-	-	0.491	2.221	0.357
FCED [41]	0.603	3.540	1.586	0.515	3.241	2.514
FMM [45] $\epsilon = 0.01$	-	3.801	2.648	-	3.195	1.573
FMM [45] $\epsilon = 1$	-	3.942	2.433	-	3.961	5.579
ACM [32]	-	10.609	0.488	-	6.651	0.400
UNet [48]	0.795	1.781	0.135	0.622	**1.998**	0.224
FC-DenseNet [29]	0.835	1.948	0.151	0.718	2.464	0.350
DeepLabv3 + MobileNetv2 [10, 50]	0.890	1.748	0.250	0.789	2.129	0.248
Ours	**0.932**	**1.635**	**0.110**	**0.823**	2.171	**0.188**

Table 2. Evaluation results of lane following. Best in bold.

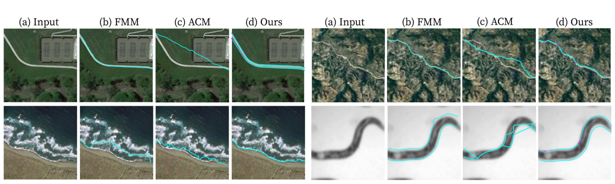

Figure 8. Visual comparison of parametric tracing in other applications, including remote sensing images [11] and C. elegans imaged under microscope [35]. (Best viewed with zoom)

Furthermore, the cosine similarity term in our objective function is noise-tolerant (Suppl. Sec. 4). Such robustness to noise has also been tested in speech verification [31] and word embedding [40]. In contrast, the performance of DCN is highly dependent on the presence of noise in training data. Our method, similar to text and speech matching, adopts the vector representation of pixel intensities, which enables parametrically searching for the shortest "distance" defined as cosine similarity in this high dimensional space.

Limitation and Caution. Our method needs boundary conditions, *i.e.* seed point. Our method requires only one seed point. The tracing can stop when the objective value is beyond the user-defined threshold or it goes beyond the pre-defined region of interest. The start seed points used in this paper were all identified automatically. However, we do not rule out the possibility that some images may require labeling seed points manually.

The time cost is dominated by the number of iterations and integral length of the curve(Suppl. Sec. 5). However, in the situation where temporal correlation is high (e.g., video in autonomous driving), the iterations required for convergence can be reduced by reasonable initialization.

Lastly, our method is designed to trace curves that are definable by $\boldsymbol{c}(t)$ and its signals exhibit similarity along curves. More examples of parametric curve tracing across different imaging domains are shown in Fig. 8. We note that parametric tracking for more complex "objects" awaits further study.

Acknowledgment

We thank TGS (Freedom and Independence surveys) and Christian Sutton for providing the seismic images and generating the corresponding ground truth.

References

[1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Gregory S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian J. Goodfellow, and et. al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *CoRR*, abs/1603.04467, 2016. 5

[2] Marloes Arts, Ihor Smal, Maarten Paul, Claire Wyman, and Erik Meijering. Particle mobility analysis using deep learning and the moment scaling spectrum. *Scientific Reports*, 9:17160, 11 2019. 6

[3] Reza Azad, Maryam Asadi, Mahmood Fathy, and Sergio Escalera. Bi-directional convlstm u-net with densley connected convolutions. pages 406–415, 10 2019. 7, 8

[4] Erik J Bekkers. B-spline {cnn}s on lie groups. In *International Conference on Learning Representations*, 2020. 4

[5] Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann Lecun. Spectral networks and locally connected networks on graphs. In *International Conference on Learning Representations (ICLR2014), CBLS, April 2014*, 2014. 4

[6] A. Buades, B. Coll, and J. . Morel. A non-local algorithm for image denoising. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 2, pages 60–65 vol. 2, 2005. 1, 6, 8

[7] J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(6):679–698, 1986. 1, 2

[8] Vicent Caselles, Ron Kimmel, and Guillermo Sapiro. Geodesic active contours, 1997. 3

[9] Fu Chang, Chun-Jen Chen, and Chi-Jen Lu. A linear-time component-labeling algorithm using contour tracing technique. *Computer Vision and Image Understanding*, 93(2):206 – 220, 2004. 1, 2

[10] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *ECCV*, 2018. 1, 3, 7, 8

[11] G. Cheng, J. Han, and X. Lu. Remote sensing image scene classification: Benchmark and state of the art. *Proceedings of the IEEE*, 105(10):1865–1883, 2017. 6, 8

[12] Nicolas Chenouard, Ihor Smal, de Fabrice Chaumont, Martin Maška, Jean-Christophe Olivo-Marin, Erik Meijering, de Fabrice Chaumont, Ihor Smal, Ihor Smal, Martin Maška, Ortiz de Carlos Solórzano, F Ivo Sbalzarini, Yuanhao Gong, Janick Cardinale, Craig Carthel, Stefano Coraluppi, Mark Winter, R Andrew Cohen, J William Godinez, Karl Rohr, J William Godinez, Yannis Kalaidzidis, Liang Liang, James Duncan, Hongying Shen, Yingke Xu, E G Klas Magnusson, Joakim Jaldén, M Helen Blau, Perrine Paul-Gilloteaux, François Waharte, Philippe Roudot, Charles Kervrann, Jean-Yves Tinevez, L Spencer Shorte, Joost Willemse, Katherine Celler, P van Gilles Wezel, Han-Wei Dan, and Yuh-Show Tsai. Objective comparison of particle tracking methods. *Nature methods*, pages 281–289, 2014. 6

[13] O. Chum and J. Matas. Matching with prosac - progressive sample consensus. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 220–226 vol. 1, 2005. 1

[14] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian. Image denoising by sparse 3-d transform-domain collaborative filtering. *IEEE Transactions on Image Processing*, 16(8):2080–2095, 2007. 1, 6, 8

[15] Carl de Boor. Package for calculating with b-splines. *SIAM Journal on Numerical Analysis*, 14:57, 10 1973. 4, 5

[16] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, page 3844–3852, Red Hook, NY, USA, 2016. Curran Associates Inc. 4

[17] David L Donoho and Iain M Johnstone. Ideal spatial adaptation by wavelet shrinkage. *Biometrika*, 81(3):425–455, 09 1994. 8

[18] M. Fey, J. E. Lenssen, F. Weichert, and H. Müller. Splinecnn: Fast geometric deep learning with continuous b-spline kernels. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 869–877, 2018. 4

[19] Martin A. Fischler and Robert C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, June 1981. 1

[20] V. Fragoso, P. Sen, S. Rodriguez, and M. Turk. Evsac: Accelerating hypotheses generation by modeling matching scores with extreme value theory. In *2013 IEEE International Conference on Computer Vision*, pages 2472–2479, 2013. 1

[21] Jun Gao, Chengcheng Tang, Vignesh Ganapathi-Subramanian, Jiahui Huang, Hao Su, and Leonidas J. Guibas. Deepspline: Data-driven reconstruction of parametric curves and surfaces. *CoRR*, abs/1901.03781, 2019. 3

[22] C. Grana, D. Borghesani, and R. Cucchiara. Optimized block-based connected components labeling with decision trees. *IEEE Transactions on Image Processing*, 19(6):1596–1609, 2010. 1, 2

[23] Alex Graves, Santiago Fernández, and Jürgen Schmidhuber. Multi-dimensional recurrent neural networks. In Joaquim Marques de Sá, Luís A. Alexandre, Włodzisław Duch, and Danilo Mandic, editors, *Artificial Neural Networks – ICANN 2007*, pages 549–558, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg. 7, 8

[24] Charles A Hall and W.Weston Meyer. Optimal error bounds for cubic spline interpolation. *Journal of Approximation Theory*, 16(2):105 – 122, 1976. 1

[25] Anders S Hansen, Maxime Woringer, Jonathan B Grimm, Luke D Lavis, Robert Tjian, and Xavier Darzacq. Robust model-based analysis of single-particle tracking experiments with spot-on. *eLife*, 7:e33125, jan 2018. 6

[26] Lifeng He, Xiwei Ren, Qihang Gao, Xiao Zhao, Bin Yao, and Yuyan Chao. The connected-component labeling problem: A review of state-of-the-art algorithms. *Pattern Recognition*, 70:25 – 43, 2017. 1

[27] T. Huang, G. Yang, and G. Tang. A fast two-dimensional median filtering algorithm. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 27(1):13–18, 1979. 1

[28] Y. Jang, J. Mun, K. Oh, and Jaeseok Kim. Block-based connected component labeling algorithm with block prediction. In *2017 40th International Conference on Telecommunications and Signal Processing (TSP)*, pages 578–581, 2017. 1, 2

[29] S. Jégou, M. Drozdzal, D. Vazquez, A. Romero, and Y. Bengio. The one hundred layers tiramisu: Fully convolutional densenets for semantic segmentation. In *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1175–1183, 2017. 1, 3, 7, 8

[30] P. Jiang and J. A. Shackleford. Cnn driven sparse multi-level b-spline image registration. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9281–9289, 2018. 4

[31] Kuruvachan K George, C. Kumar, Ramachandran K I, and Ashish Panda. Cosine distance features for robust speaker verification. 09 2015. 7

[32] Michael Kass, Andrew Witkin, and Demetri Terzopoulos. Snakes: Active contour models. *INTERNATIONAL JOURNAL OF COMPUTER VISION*, 1(4):321–331, 1988. 3, 7, 8

[33] Satyanad Kichenassamy, Arun Kumar, Peter Olver, Allen Tannenbaum, and Anthony Yezzi. Conformal curvature flows: From phase transitions to active vision. *Archive for Rational Mechanics and Analysis*, 134:275–301, 01 1996. 3

[34] R. Kimmel and J. A. Sethian. Computing geodesic paths on manifolds. *Proceedings of the National Academy of Sciences*, 95(15):8431–8435, 1998. 2

[35] Vebjorn Ljosa, Katherine Sokolnicki, and Anne Carpenter. Annotated high-throughput microscopy image sets for validation. *Nature methods*, 9:637, 06 2012. 6, 8

[36] R. Malladi and J. A. Sethian. Level set and fast marching methods in image processing and computer vision. In *Proceedings of 3rd IEEE International Conference on Image Processing*, volume 1, pages 489–492 vol.1, 1996. 3

[37] Soham Mandal and Virginie Uhlmann. A learning-based formulation of parametric curve fitting for bioimage analysis. *bioRxiv*, 2020. 3

[38] Z. Miao, B. Wang, W. Shi, and H. Zhang. A semi-automatic method for road centerline extraction from vhr images. *IEEE Geoscience and Remote Sensing Letters*, 11(11):1856–1860, 2014. 3

[39] Michael J. Nasse and Jörg C. Woehl. Realistic modeling of the illumination point spread function in confocal scanning optical microscopy. *J. Opt. Soc. Am. A*, 27(2):295–302, Feb 2010. 6

[40] Kim Anh Nguyen, Sabine Schulte im Walde, and Ngoc Thang Vu. Neural-based noise filtering from word embeddings. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2699–2707, Osaka, Japan, Dec. 2016. The COLING 2016 Organizing Committee. 7

[41] N. Ofir, M. Galun, S. Alpert, A. Brandt, B. Nadler, and R. Basri. On detection of faint edges in noisy images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(4):894–908, 2020. 1, 2, 6, 8

[42] N. Ofir, M. Galun, B. Nadler, and R. Basri. Fast detection of curved edges at low snr. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 213–221, 2016. 1

[43] Stanley Osher and James A Sethian. Fronts propagating with curvature-dependent speed: Algorithms based on hamilton-jacobi formulations. *Journal of Computational Physics*, 79(1):12 – 49, 1988. 2

[44] N. Otsu. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1):62–66, 1979. 1, 2, 6, 8

[45] Gabriel Peyré, Mickaël Péchaud, Renaud Keriven, and Laurent D. Cohen. Geodesic Methods in Computer Vision and Graphics. *Foundations and Trends in Computer Graphics and Vision*, 5(3-4):197–397, 2010. 2, 3, 7, 8

[46] Hartmut Prautzsch, Wolfgang Boehm, and Marco Paluszny. *Bezier and B-Spline Techniques*. Springer-Verlag, Berlin, Heidelberg, 2002. 4

[47] Mayra Z. Rodriguez, Cesar H. Comin, Dalcimar Casanova, Odemir M. Bruno, Diego R. Amancio, Luciano da F. Costa, and Francisco A. Rodrigues. Clustering algorithms: A comparative approach. *PLOS ONE*, 14(1):1–34, 01 2019. 2

[48] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In Nassir Navab, Joachim Hornegger, William M. Wells, and Alejandro F. Frangi, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pages 234–241, Cham, 2015. Springer International Publishing. 1, 3, 7, 8

[49] Marie-Hélène Roy Cardinal, Gilles Soulez, Jean-Claude Tardif, Jean Meunier, and Guy Cloutier. Fast-marching segmentation of three-dimensional intravascular ultrasound images: A pre- and post-intervention study. *Medical physics*, 37:3633–47, 07 2010. 3

[50] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *CVPR*, 2018. 1, 3, 7, 8

[51] J A Sethian. A fast marching level set method for monotonically advancing fronts. *Proceedings of the National Academy of Sciences*, 93(4):1591–1595, 1996. 2

[52] Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-kin Wong, and Wang-chun Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'15, page 802–810, Cambridge, MA, USA, 2015. MIT Press. 7

[53] Irwin Sobel. An isotropic 3x3 image gradient operator. *Presentation at Stanford A.I. Project 1968*, 02 2014. 1, 2

[54] Richard Szeliski and James M. Coughlan. Spline-based image registration. *International Journal of Computer Vision*, 22:199–218, 1997. 4

[55] Johannes Thomsen, Magnus Berg Sletfjerding, Simon Bo Jensen, Stefano Stella, Bijoya Paul, Mette Galsgaard Malle, Guillermo Montoya, Troels Christian Petersen, and Nikos S Hatzakis. Deepfret, a software for rapid and automated single-molecule fret data classification using deep learning. *eLife*, 9:e60404, nov 2020. 6

[56] J. Wu, Z. Yin, and Y. Xiong. The fast multilevel fuzzy edge detection of blurry images. *IEEE Signal Processing Letters*, 14(5):344–347, 2007. 2

[57] Kesheng Wu, Ekow Otoo, and Kenji Suzuki. Optimizing two-pass connected-component labeling algorithms. *Pattern Anal. Appl.*, 12(2):117–135, Feb. 2009. 1

[58] Chenyang Xu and J.L. Prince. Gradient vector flow: a new external force for snakes. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 66–71, 1997. 3