Vision-based Discovery of Nonlinear Dynamics for 3D Moving Target

Zitong Zhang[1] , **Yang Liu**[2] and **Hao Sun**[1,*]
[1]Gaoling School of Artificial Intelligence, Renmin University of China, Beijing, China
[2]School of Engineering Science, University of Chinese Academy of Sciences, Beijing, China
zhangzitong@ruc.edu.cn; liuyang22@ucas.ac.cn; haosun@ruc.edu.cn

Abstract

Data-driven discovery of governing equations has kindled significant interests in many science and engineering areas. Existing studies primarily focus on uncovering equations that govern nonlinear dynamics based on direct measurement of the system states (e.g., trajectories). Limited efforts have been placed on distilling governing laws of dynamics directly from videos for moving targets in a 3D space. To this end, we propose a vision-based approach to automatically uncover governing equations of nonlinear dynamics for 3D moving targets via raw videos recorded by a set of cameras. The approach is composed of three key blocks: (1) a target tracking module that extracts plane pixel motions of the moving target in each video, (2) a Rodrigues' rotation formula-based coordinate transformation learning module that reconstructs the 3D coordinates with respect to a predefined reference point, and (3) a spline-enhanced library-based sparse regressor that uncovers the underlying governing law of dynamics. This framework is capable of effectively handling the challenges associated with measurement data, e.g., noise in the video, imprecise tracking of the target that causes data missing, etc. The efficacy of our method has been demonstrated through multiple sets of synthetic videos considering different nonlinear dynamics.

1 Introduction

Nonlinear dynamics is ubiquitous in nature. Data-driven discovery of underlying laws or equations that govern complex dynamics has drawn great attention in many science and engineering areas such as astrophysics, aerospace science, biomedicine, etc. Existing studies primarily focus on uncovering governing equations based on direct measurement of the system states, e.g., trajectory time series, [Bongard and Lipson, 2007; Schmidt and Lipson, 2009; Brunton *et al.*, 2016; Rudy *et al.*, 2017; Chen *et al.*, 2021b; Sun *et al.*, 2021]. Limited efforts have been placed on distilling governing laws of dynamics directly from videos for moving targets in a 3D space, which represents a novel and interdisciplinary research domain. This challenge calls for a solution of fusing various techniques, including computer stereo vision, visual object tracking, and symbolic discovery of equations.

We consider a moving object in a 3D space, recorded by a set of horizontally positioned, calibrated cameras at different locations. Discovery of the governing equations for the moving target first requires accurate estimation of its 3D trajectory directly from the videos, which can be realized based on computer stereo vision and object tracking techniques. Computer stereo vision, which aims to reconstruct 3D coordinates for depth estimation of a given target, has shown immense potential in the fields of robotics [Nalpantidis and Gasteratos, 2011; Li *et al.*, 2021], autonomous driving [Ma *et al.*, 2019; Peng *et al.*, 2020], etc. Disparity estimation is a crucial step in stereo vision, as it computes the distance information of objects in a 3D space, thereby enabling accurate perception and understanding of the scene. Recent advances of deep learning has kindled successful techniques for visual object tracking e.g., DeepSORT [Wojke *et al.*, 2017] and YOLO [Redmon *et al.*, 2016]. The aforementioned techniques lay a critical foundation to accurately estimate the 3D trajectory of a moving target for distilling governing equations, simply based on videos recorded by multiple cameras in a complex scene.

We assume that the nonlinear dynamics of a moving target can be described by a set of ordinary differential equations, e.g., $d\mathbf{y}/dt = \mathcal{F}(\mathbf{y})$, where $\mathcal{F}$ is a nonlinear function of the d-dimensional system state $\mathbf{y} = \{y_1(t), y_2(t), \ldots, y_d(t)\} \in \mathbb{R}^d$. The objective of equation discovery is to identify the closed form of $\mathcal{F}$ given observations of $\mathbf{y}$. This could be achieved via symbolic regression [Bongard and Lipson, 2007; Schmidt and Lipson, 2009; Sahoo *et al.*, 2018; Petersen *et al.*, 2021; Mundhenk *et al.*, 2021; Sun *et al.*, 2023] or sparse regression [Brunton *et al.*, 2016; Rudy *et al.*, 2017; Rao *et al.*, 2023]. When the data is noisy and sparse, differentiable models (e.g., neural networks (NN) [Chen *et al.*, 2021b], cubic splines [Sun *et al.*, 2021; Sun *et al.*, 2022]) are employed to reconstruct the system states, thereby forming physics-informed learning for more robust discovery.

Recently, attempts have been made toward scene understanding and prediction grounding physical concepts [Jaques *et al.*, 2020; Chen *et al.*, 2021a]. Although a number of efforts have been placed on distilling the unknown governing laws of dynamics from videos for moving targets [Champion *et al.*,

[*] Corresponding author

2019; Udrescu and Tegmark, 2021; Luan *et al.*, 2022], the system dynamics was assumed in plane (e.g., in a 2D space). To our knowledge, distilling governing equations for a moving object in a 3D space (e.g., $d = 3$) directly from raw videos remains scant in literature. To this end, we introduce a unified vision-based approach to automatically uncover governing equations of nonlinear dynamics for a moving target in a predefined reference coordinate system, based on raw video data recorded by a set of horizontally positioned, calibrated cameras at different locations.

Contributions. The proposed approach is composed of three key blocks: (1) a target tracking module based on YOLO-v8 that extracts plane pixel motions of the moving target in each video data; (2) a coordinate transformation model based on Rodrigues' rotation formula, which allows the conversion of pixel coordinates obtained through target tracking to 3D spatial/physical coordinates in a predefined reference coordinate system given the calibrated parameters of only one camera; and (3) a spline-enhanced library-based sparse regressor that uncovers a parsimonious form of the underlying governing equations for the nonlinear dynamics. Through the integration of these components, it becomes possible to extract spatiotemporal information of a moving target from 2D video data and subsequently uncover the underlying governing law of dynamics. This integrated framework excels in effectively addressing challenges associated with measurement noise and data gaps induced by imprecise target tracking. Results from extensive experiments demonstrate the efficacy of the proposed method. This endeavor offers a novel perspective for understanding the complex dynamics of moving targets in a 3D space.

2 Related Work

Computer stereo vision. Multi-view stereo aims to reconstruct a 3D model of the observed scene from images with different viewpoints [Schönberger *et al.*, 2016; Galliani *et al.*, 2016], assuming the intrinsic and extrinsic camera parameters are known. Recently, deep learning has been employed to tackle this challenge, such as convolutional neural networks [Flynn *et al.*, 2016; Huang *et al.*, 2018] and adaptive modulation network with co-teaching strategy [Wang *et al.*, 2021].

Target tracking. Methods for vision-based target tracking can be broadly categorized into two main classes: correlation filtering and deep learning. Compared to traditional algorithms, correlation filtering-based approaches offer faster target tracking [Mueller *et al.*, 2017], while deep learning-based methods [Ciaparrone *et al.*, 2020; Marvasti-Zadeh *et al.*, 2021] provide higher precision.

Governing equation discovery. Data-driven discovery of governing equations can be realized through a number of symbolic/sparse regression techniques. The most popular symbolic regression methods include genetic programming [Koza, 1994; Bongard and Lipson, 2007; Schmidt and Lipson, 2009], symbolic neural networks [Sahoo *et al.*, 2018], deep symbolic regression [Petersen *et al.*, 2021; Mundhenk *et al.*, 2021], and Monte Carlo tree search [Lu *et al.*, 2021; Sun *et al.*, 2023]. Sparse regression techniques such as SINDy [Brunton *et al.*, 2016; Rudy *et al.*, 2017; Rao *et al.*, 2023] leverage a predefined library that includes a limited number of candidate terms, which search for the underlying equations in a compact solution space.

Physics-informed learning. Physics-informed learning has been developed to deal with noisy and sparse data in the context of equation discovery. Specifically, differentiable models (e.g., NN [Raissi *et al.*, 2019; Chen *et al.*, 2021b], cubic splines [Sun *et al.*, 2021; Sun *et al.*, 2022]) are employed to reconstruct the system states and approximate the required derivative terms required to form the underlying black equations.

Vision-based discovery of dynamics. Very recently, attempts have been made to discover the governing of equations for moving objects directly from videos. These methods are generally based on autoencoders that extract the latent dynamics for equation discovery [Champion *et al.*, 2019; Udrescu and Tegmark, 2021; Luan *et al.*, 2022]. Other related works include the discovery of intrinsic dynamics [Floryan and Graham, 2022] or fundamental variables [Chen *et al.*, 2022] based on high-dimensional data such as videos.

3 Methodology

We here elucidate the basic concept and approach of vision-based discovery of nonlinear dynamics for a moving target in a 3D space. Figure 1 shows the schematic architecture of our method. The target tracking module serves as the foundational stage, which extracts pixel-level motion information from the target's movements across consecutive frames in a video sequence. The coordinate transformation module utilizes Rodrigues' rotation formula with respect to a predefined reference coordinate origin, which lays the groundwork for the subsequent analysis of the object's dynamics. The final crucial component is the spline-enhanced library-based sparse regressor, essential for revealing the fundamental dynamics governing object motion.

3.1 Coordinates Transformation

In this paper, we employ three cameras with known and fixed positions oriented in different directions to independently capture the motion of an object (see Figure 1a). With the constraint of calibrating only one camera, our coordinate learning module becomes essential (e.g., the 3D trajectory of the target and other camera parameters can be simultaneously learned). In particular, it is tasked with learning the unknown parameters of the other two cameras, including scaling factors and the rotation angle on each camera plane. These parameters enable to reconstruct the motion trajectory of the object in the reference coordinate system. We leverage Rodrigues' rotation formula to compute vector rotations in three-dimensional space, which enables the derivation of the rotation matrix, describing the rotation operation from a given initial vector to a desired target vector. This formula finds extensive utility in computer graphics, computer vision, robotics, and 3D rigid body motion problems.

In a 3D space, a rotation matrix is used to represent the transformation of a rigid body around an axis. Let $\mathbf{v}_0$ represent the initial vector and $\mathbf{v}_1$ denote the target vector. We denote the rotation matrix as $\mathbf{R}$. The relationship between

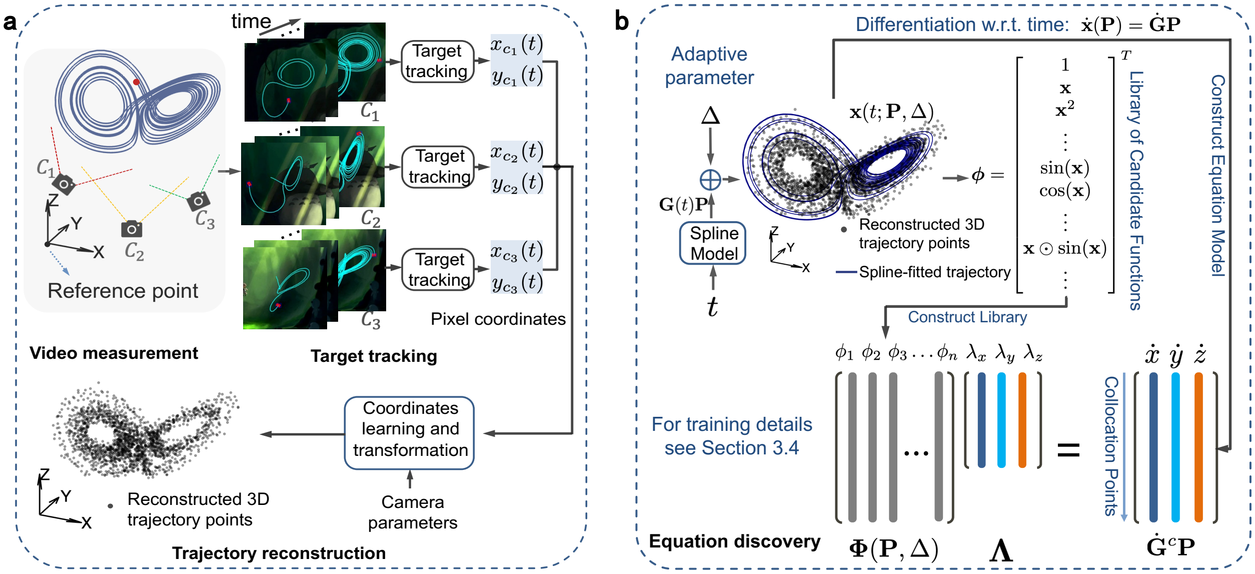

a
time
C_1
C_2
C_3
Z
Y
X
Reference point
Target tracking
Target tracking
Target tracking
$x_{c_1}(t)$
$y_{c_1}(t)$
$x_{c_2}(t)$
$y_{c_2}(t)$
$x_{c_3}(t)$
$y_{c_3}(t)$
Pixel coordinates
Video measurement
Target tracking
Coordinates learning and transformation
Camera parameters
Reconstructed 3D trajectory points
Trajectory reconstruction
b
Differentiation w.r.t. time: $\dot{\mathbf{x}}(\mathbf{P}) = \dot{\mathbf{G}}\mathbf{P}$
Adaptive parameter
Δ
$\mathbf{G}(t)\mathbf{P}$
Spline Model
t
$\mathbf{x}(t;\mathbf{P},\Delta)$
Reconstructed 3D trajectory points
Spline-fitted trajectory
$\phi = [1, \mathbf{x}, \mathbf{x}^2, \vdots, \sin(\mathbf{x}), \cos(\mathbf{x}), \vdots, \mathbf{x}\odot\sin(\mathbf{x}), \vdots]^T$
Library of Candidate Functions
Construct Equation Model
Construct Library
$\phi_1\ \phi_2\ \phi_3 \dots \phi_n\ \lambda_x\ \lambda_y\ \lambda_z$
$\dot{x}\ \dot{y}\ \dot{z}$
For training details see Section 3.4
Collocation Points
Equation discovery
$\mathbf{\Phi}(\mathbf{P},\Delta)$
$\mathbf{\Lambda}$
$\dot{\mathbf{G}}^c\mathbf{P}$

Figure 1: Schematic of vision-based discovery of nonlinear dynamics for 3D moving target. Firstly, we record the motion trajectory of the object in a 3D space using multiple cameras in a predefined reference coordinate system (see **a**). Pixel trajectory coordinates are obtained through target identification and tracking. Note that camera parameters include the camera's position, the normal vector of the camera's view plane, and the calibrated camera parameters, which comprise the scaling factor and tilt angle. In particular, we use coordinate learning and transformation to obtain the spatial motion trajectory in the reference coordinate system. Secondly, for each dimension of the trajectory, we introduce a spline-enhanced library-based sparse regressor to uncover the underlying governing law of dynamics. The differentiation for the trajectory and spline curve with respect to time are respectively given by $\dot{\mathbf{x}} = d\mathbf{x}/dt$, $\dot{\mathbf{G}} = d\mathbf{G}/dt$ (see **b**).

the pre- and post-rotation vectors can be expressed as $\mathbf{v}_1 = \mathbf{R}\mathbf{v}_0$. The rotation angle, denoted as θ, can be calculated via $\cos\theta = \frac{\mathbf{v}_0\cdot\mathbf{v}_1}{\|\mathbf{v}_0\|\|\mathbf{v}_1\|}$. The rotation axis is represented by the unit vector $\mathbf{u} = [u_x, u_y, u_z]$, namely, $\mathbf{u} = \frac{\mathbf{v}_0\times\mathbf{v}_1}{\|\mathbf{v}_0\times\mathbf{v}_1\|}$. Having defined the rotation angle θ and the unit vector $\mathbf{k}$, we construct the rotation matrix $\mathbf{R}$ using Rodrigues' rotation formula:

$$\mathbf{R} = \mathbf{I} + \sin\theta\mathbf{U} + (1-\cos\theta)\mathbf{U}^2. \tag{1}$$

where $\mathbf{I}$ is a 3×3 identity matrix, and $\mathbf{U}$ is a 3×3 skew-symmetric matrix representing the cross product of the rotation axis vector $\mathbf{u}$ expressed as

$$\mathbf{U} = \begin{bmatrix} 0 & -u_z & u_y \\ u_z & 0 & -u_x \\ -u_y & u_x & 0 \end{bmatrix}. \tag{2}$$

When projecting an object onto a plane, denoting the coordinates of the projection as $\mathbf{x}_p = (x_p, y_p, z_p)^T$. The procedure for projecting a 3D object onto a plane is elaborated in Appendix A. The object's projected shape on a plane is determined solely by plane's normal vector. Refer to Appendix B for a detailed proof, and Appendix C for calculation of the camera's offsets from a camera plane.

3.2 Cubic B-Splines

B-splines are differentiable, and constructed using piecewise polynomial functions called basis functions. When the measurement data is noisy and sparse, cubic B-splines could serve as a differentiable surrogate model to form robust physics-informed learning for equation discovery [Sun *et al.*, 2021]. We herein adopt this approach to tackle challenges associated with data noise and gaps induced by the imprecise target tracking for discovering laws of 3D dynamics. The i-th cubic B-spline basis function of degree k, written as $G_{i,k}(u)$, can be defined recursively as:

$$\begin{aligned} G_{i,0}(u) &= \begin{cases} 1 & \text{if } u_i \le u < u_{i+1} \\ 0 & \text{otherwise} \end{cases}, \\ G_{i,k}(u) &= \frac{u-u_i}{u_{i+k}-u_i}G_{i,k-1}(u) + \frac{u_{i+k+1}-u}{u_{i+k+1}-u_{i+1}}G_{i+1,k-1}(u), \end{aligned} \tag{3}$$

where u_i represents a knot that partitions the computational domain. By selecting appropriate control points and combinations of basis functions, cubic B-splines with $\mathbb{C}^2$ continuity can be customized to meet specific requirements. In general, a cubic B-spline curve of degree p defined by $n+1$ control points $\mathbf{P} = \{\mathbf{p}_0, \mathbf{p}_1, ..., \mathbf{p}_n\}$ and a knot vector $U = \{u_0, u_1, ..., u_m\}$ is given by: $C(u) = \sum_{i=0}^{n} G_{i,3}(u)\cdot\mathbf{p}_i$. To ensure the curve possesses continuous and smooth tangent directions at the starting and ending nodes, meeting the first derivative interpolation requirement, we use Clamped cubic B-spline curves for fitting.

3.3 Network Architecture

We utilized the YOLO-v8 for object tracking in the recorded videos (see Figure 1a). Regardless of whether the captured object has an irregular shape or is in a rotated state, we only need to capture their centroid positions and track them to obtain pixel data. Subsequently, leveraging Rodrigues' rotation formula and based on the calibrated camera, we derive the scaling and rotation factors of the other two cameras. These factors enable the conversion of the object trajectory's pixel coordinates into the world coordinates deducing the physical trajectory. For the trajectory varying with time in each dimension, we use the cubic B-splines to fit the trajectory and a library-based sparse regressor to uncover the underlying governing law of dynamics in the reference coordinate system. This approach is capable of dealing with data noise, multiple instances of data missing and gaps.

Learning 3D trajectory. In this work, we use a three-camera setup to capture and represent the object's 2D motion trajectory in each video scene, yielding the 2D coordinates denoted as (x_{rp}, y_{rp}). The rotation matrix $\mathbf{R}$ is decomposed to retain only the first two rows, denoted as $\mathbf{R}^{-}$, to suitably handle the projection onto the image planes. Under the condition of calibrating only one camera, we can reconstruct the coordinates of a moving object in the reference 3D coordinate system using three fixed cameras capturing an object's motion in a 3D space. The assumed given information includes normal vectors $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3$ of camera planes for all three cameras, the positions of the cameras, as well as a scaling factor s_1 and rotation angles ϑ_1 for one of the cameras. We define the scaling factor vector as $\mathbf{s} = \{s_1, s_2, s_3\}$ and the rotation angle vector as $\vartheta = \{\vartheta_1, \vartheta_2, \vartheta_3\}$. The loss function for reconstructing the 3D coordinates of the object in the reference coordinate system is given by

$$\mathcal{L}_r\left(\mathbf{s}^*; \vartheta^*\right) = \frac{1}{N_m} \left\|\tilde{\mathbf{x}} - \left(s_1 \mathcal{T}\left(\vartheta_1\right) \mathbf{x}_{c_1} + \Delta_{c_1}\right)\right\|_2^2, \tag{4}$$

where

$$\tilde{\mathbf{x}} = \mathbf{R}_1^{-} \begin{bmatrix} \mathbf{R}_2^{-} \\ \mathbf{R}_3^{-} \end{bmatrix}^{-1} \begin{bmatrix} s_2 \mathcal{T}\left(\vartheta_2\right) \mathbf{x}_{c_2} + \Delta_{c_2} \\ s_3 \mathcal{T}\left(\vartheta_3\right) \mathbf{x}_{c_3} + \Delta_{c_3} \end{bmatrix}. \tag{5}$$

Here, $\mathbf{s}^* = \{s_2, s_3\}$ and $\vartheta^* = \{\vartheta_2, \vartheta_3\}$. Note that $\mathbf{x}_{c_i} = (x_{c_i}, y_{c_i})^T$ represents the pixel coordinates, black Δ_{c_i}denotes deviation of object position from image coordinate origin. N_m the number of effectively recognized object coordinate points, $\mathcal{T}(\vartheta)$ the transformation matrix induced by rotation angle ϑ expressed as $\mathcal{T}(\vartheta) = [\cos\vartheta \;\; \sin\vartheta; \; -\sin\vartheta \;\; \cos\vartheta]$. When using three cameras, the transformation between the object's coordinates $\mathbf{x}_{ref}$ in the reference coordinate system and the pixel coordinates $\mathbf{x}_c$ in the camera setups reads

$$\mathbf{x} = \begin{bmatrix} \mathbf{R}_1^{-} \\ \mathbf{R}_2^{-} \\ \mathbf{R}_3^{-} \end{bmatrix}^{-1} \begin{bmatrix} s_1 \mathcal{T}(\vartheta_1) \mathbf{x}_{c_1} + \Delta_{c_1} \\ s_2 \mathcal{T}(\vartheta_2) \mathbf{x}_{c_2} + \Delta_{c_2} \\ s_3 \mathcal{T}(\vartheta_3) \mathbf{x}_{c_3} + \Delta_{c_3} \end{bmatrix}. \tag{6}$$

Solving for parameter values $(\mathbf{s}^*, \vartheta^*)$ via optimization of Eq. (4), we can subsequently compute the reconstructed 3D physical coordinates via the calculation provided in Eq. (6).

Equation discovery. Given the potential challenges in target tracking, e.g., momentary target loss, noise, or occlusions, we leverage physics-informed spline learning to address these issues (see Figure 1b). In particular, cubic B-splines are employed to approximate the 3D trajectory. Given three sets of control points denoted as $\mathbf{P} = \{\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3\} \in \mathbb{R}^{r \times 3}$. Given that the coordinate system is arbitrarily defined, and to enhance the fitting of data $\mathcal{D}_r$, we introduce the learnable adaptive offset parameter $\Delta = \{\Delta_1, \Delta_2, \Delta_3\}$. The 3D parametric curves where $\mathbf{x}(t; \mathbf{P}, \Delta)$ are defined by the control point vectors $\mathbf{P}$, the cubic B-spline basis functions $\mathbf{G}(t)$ and the offset parameter Δ, namely, $\mathbf{x}(t; \mathbf{P}, \Delta) = \mathbf{G}(t)\mathbf{P} + \Delta$. Since the basis functions consist of differentiable polynomials, the expression of its differential equation is given by $\dot{\mathbf{x}}(\mathbf{P}) = \dot{\mathbf{G}}\mathbf{P}$. Generally, the dynamics is governed by a limited number of significant terms, which can be selected from a library of l candidate functions, *v.i.z.*, $\boldsymbol{\phi}(\mathbf{x}) \in \mathbb{R}^{1 \times l}$ [Brunton *et al.*, 2016]. The governing equations can be written as:

$$\dot{\mathbf{x}}(\mathbf{P}) = \boldsymbol{\phi}(\mathbf{P}, \Delta)\boldsymbol{\Lambda}, \tag{7}$$

where $\boldsymbol{\phi}(\mathbf{P}, \Delta) = \boldsymbol{\phi}(\mathbf{x}(t; \mathbf{P}, \Delta))$, and $\boldsymbol{\Lambda} = \{\boldsymbol{\lambda}_1, \boldsymbol{\lambda}_2, \boldsymbol{\lambda}_3\} \in \mathcal{S} \subset \mathbb{R}^{l \times 3}$ is the sparse coefficient matrix belonging to a constraint subset $\mathcal{S}$ (only the terms active in $\boldsymbol{\phi}$ are non-zero).

Accordingly, the task of equation discovery can be formulated as follows: when provided with reconstructed 3D trajectory data $\mathcal{D}_r = \{\mathbf{x}_1^m, \mathbf{x}_2^m, \mathbf{x}_3^m\} \in \mathbb{R}^{N_m \times 3}$. In other words, $\mathcal{D}_r$ is presented as effectively tracking the object movements in a video and subsequently transforming them into a 3D trajectory, where N_m is the number of data points. Our goal is to identify the suitable set of $\mathbf{P}$ and a sparse $\boldsymbol{\Lambda}$ that fits the trajectory data meanwhile satisfying Eq. (7). Considering that the reconstructed trajectory $\mathcal{D}_r$ might exhibit noise or temporal discontinuity, we use collocation points denoted as $\mathcal{D}_c = \{t_0, t_1, \ldots, t_{n_c - 1}\}$ to compensate data imperfection, where $\mathcal{D}_c$ denotes the randomly sampled set of N_c number of collocation points ($N_c \gg N_m$). These points are strategically employed to reinforce the fulfillment of physics constraints at all time instances (see Figure 1b).

3.4 Network Training

The loss function for this network comprises three main components, namely, the data component $\mathcal{L}_d$, the physics component $\mathcal{L}_p$, and the sparsity regularizer, given by:

$$\arg \min_{\{\mathbf{P}, \boldsymbol{\Lambda}, \Delta\}} \left[\mathcal{L}_d + \alpha \mathcal{L}_p + \beta \|\boldsymbol{\Lambda}\|_0\right], \tag{8}$$

where

$$\mathcal{L}_d\left(\mathbf{P}, \Delta; \mathcal{D}_r\right) = \frac{1}{N_m} \sum_{i=1}^{3} \left\|\mathbf{G}_m \mathbf{p}_i + \Delta_i - \mathbf{x}_i^m\right\|_2^2, \tag{9a}$$

$$\mathcal{L}_p\left(\mathbf{P}, \Delta, \boldsymbol{\Lambda}; \mathcal{D}_c\right) = \frac{1}{N_c} \sum_{i=1}^{3} \left\|\boldsymbol{\Phi}(\mathbf{P}, \Delta)\boldsymbol{\lambda}_i - \dot{\mathbf{G}}^c \mathbf{p}_i\right\|_2^2. \tag{9b}$$

Here, $\mathbf{G}_m$ denotes the spline basis matrix evaluated at the measured time instances, $\mathbf{x}_i^m$ the coordinates in each dimension after 3D reconstruction in the reference coordinate system (may be sparse or exhibit data gaps whereas $\dot{\mathbf{G}}_c$ the derivative of the spline basis matrix evaluated at the collocation instances. The term $\mathbf{G}_m \mathbf{p}_i$ is employed to fit the measured trajectory in each dimension, while $\dot{\mathbf{G}}^c \mathbf{p}_i$ is used to reconstruct the potential equations evaluated at the collocation instances. Additionally, $\boldsymbol{\Phi} \in \mathbb{R}^{N_c \times l}$ represents the collocation library matrix encompassing the collection of candidate terms, $\|\boldsymbol{\Lambda}\|_0$ the sparsity regularizer, α and β the relative weighting parameters.

Since the regularizer $\|\boldsymbol{\Lambda}\|_0$ leads to an NP-hard optimization issue, we apply an Alternate Direction Optimization (ADO) strategy (see Appendix D) to optimize the loss function [Chen *et al.*, 2021b; Sun *et al.*, 2021]. The interplay of spline interpolation and sparse equations yields subsequent effects: the spline interpolation ensures accurate modeling of the system's response, its derivatives, and the candidate function terms, thereby laying the foundation for constructing the governing equations. Simultaneously, the equations represented in a sparse manner synergistically constrain spline interpolation and facilitate the projection of accurate candidate functions. Ultimately, this transforms the extraction of a 3D trajectory of an object from video into a closed-form differential equation.

$$\dot{\mathbf{x}} = \phi(\mathbf{x} - \Delta^*)\mathbf{\Lambda}^*. \tag{10}$$

After applying ADO to execute our model, resulting in the optimal control point matrix $\mathbf{P}^*$, sparse matrix $\mathbf{\Lambda}^*$, and adaptive parameter Δ^*, an affine transformation is necessary to eliminate Δ^* in the identified equations. We replace $\mathbf{x}$ with $\mathbf{x} - \Delta^*$, as shown in Eq. (10), to obtain the final form of equations. We then assign a small value to prune equation coefficients, yielding the discovered governing equations in a predefined 3D coordinate system.

4 Experiments

In this section, we evaluate our method for uncovering 3D governing equations of a moving target automatically from videos using nine datasets[1]. The nonlinear dynamical equations for these chaotic systems and their respective trajectories can be found in Appendix E (see Figure S1). We generate 3D trajectories based on the governing equations of the dataset and subsequently produce corresponding video data captured from various positions. Our analysis encompasses the method's robustness across distinct video backgrounds, varying shapes of moving objects, object rotations, levels of data noise, and occlusion scenarios. We further validate the identified equations demonstrating their interpretability and generalizability. The proposed computational framework is implemented in PyTorch. All simulations in this study are conducted on an Intel Core i9-13900 CPU workstation with an NVIDIA GeForce RTX 4090 GPU.

Data generation. The videos in this study are synthetically generated using MATLAB to simulate real dynamic systems captured by cameras. To commence, the dynamic system is pre-defined, and its trajectory is simulated utilizing the 4th-order Runge-Kutta method in MATLAB. Leveraging the generated 3D trajectory, a camera's orientation is established within a manually defined 3D coordinate system to simulate the 2D projection of the object onto the camera plane. The original colored images featuring the moving object are confined to dimensions of 512×512 pixels at 25 frames per second (fps). Various shapes are employed as target markers in the video along with local dynamics (e.g., with self-rotation) to emulate the motion of the object (see Appendix F). Subsequently, a set of background images are randomly selected to mimic the real-world video scenarios. The resultant videos generated within the background imagery comprise color content, with each frame containing RGB channels (e.g., see Appendix Figure S2). After obtaining the video data, it becomes imperative to perform object recognition and tracking on the observed entities based on the YOLO-v8 method.

[1] The datasets are derived from instances introduced in [Gilpin, 2021], where we utilize the following examples: Lorenz, SprottE, RayleighBenard, SprottF, NoseHoover, Tsucs2 and WangSun.

Cases	Methods	Terms Found?	False Positives	ℓ_2 Error ($\times 10^{-2}$)	P (%)	R (%)
Lorenz	Ours	Yes	1	**1.50**	**92.31**	**100**
	PySINDy	Yes	1	4.17	**92.31**	**100**
SprottE	Ours	Yes	**0**	**0.15**	**100**	**100**
	PySINDy	Yes	3	3.48	72.73	**100**
RayleighBenard	Ours	Yes	**1**	**2.00**	**91.67**	**100**
	PySINDy	Yes	2	2.74	84.62	**100**
SprottF	Ours	**Yes**	**0**	**0.16**	**100**	**100**
	PySINDy	No	1	7.51	90	90
NoseHoover	Ours	Yes	**0**	**6.22**	**100**	**100**
	PySINDy	Yes	3	824.44	75	**100**
Tsucs2	Ours	Yes	1	**5.39**	**93.75**	**100**
	PySINDy	Yes	1	12.29	93.75	**100**
WangSun	Ours	**Yes**	**1**	**0.16**	**93.33**	**100**
	PySINDy	No	3	856.47	86.67	92.86

Table 1: The performance of our method compared to the PySINDy in reconstructing three-dimensional coordinates from videos (see Appendix Table S3 for comparisons with other methods.)

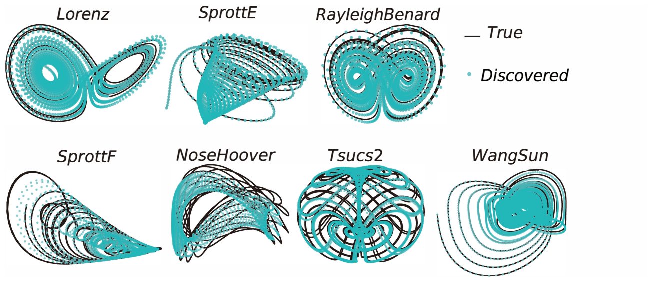


Figure 2: Discovered 3D trajectories *vs.* the ground truth.

4.1 Results

Evaluation metrics. We employ both qualitative and quantitative metrics to assess the performance of our method. Our goal is to identify all equation terms as accurately as possible while eliminating irrelevant terms (False Positives) to the greatest extent. The error ℓ_2, represented as $||\mathbf{\Lambda}_{id} - \mathbf{\Lambda}_{true}||_2/||\mathbf{\Lambda}_{true}||_2$, quantifies the relative difference between the identified coefficients $\mathbf{\Lambda}_{id}$ and the ground truth $\mathbf{\Lambda}_{true}$. To avoid the overshadowing of smaller coefficients when there is a significant disparity in their magnitudes, we introduce a non-dimensional measure to obtain a more comprehensive evaluation.

The discovery of governing equations can be framed as a binary classification task [Rao *et al.*, 2022], determining whether a particular term exists or not, given a candidate library. Hence, we introduce precision and recall as metrics for evaluation, which quantify the proportion of correctly identified coefficients among the actual coefficients, expressed as: $P = \|\mathbf{\Lambda}_{\text{id}} \odot \mathbf{\Lambda}_{\text{true}}\|_0 \, / \, \|\mathbf{\Lambda}_{\text{id}}\|_0$ and $R = \|\mathbf{\Lambda}_{\text{id}} \odot \mathbf{\Lambda}_{\text{true}}\|_0 \, / \, \|\mathbf{\Lambda}_{\text{true}}\|_0$, where $\odot$ denotes element-wise product. Successful identification is achieved when both the entries in the identified and true vectors are non-zero.

Discovery results. Based on our evaluation metrics (e.g., the ℓ_2 error, the number of correct and incorrect equations terms found, precision, and recall), a detailed analysis of the experimental results obtained by our method is found in Table S3 (without data noise). After reconstructing the 3D tra-

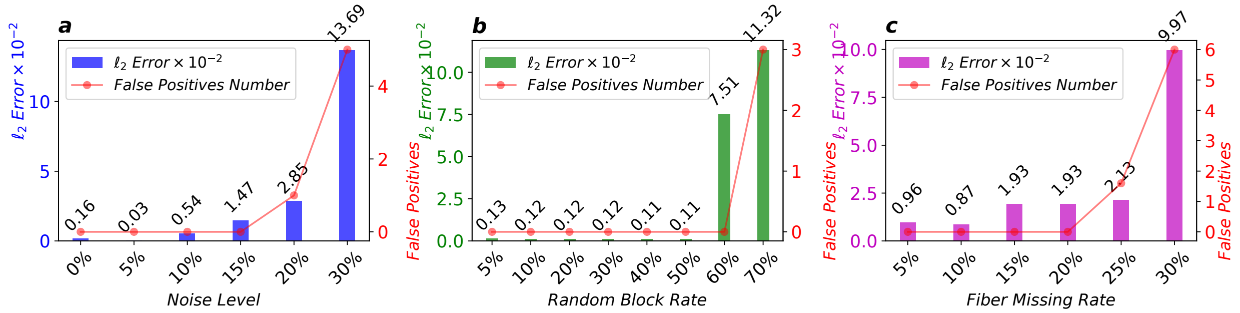
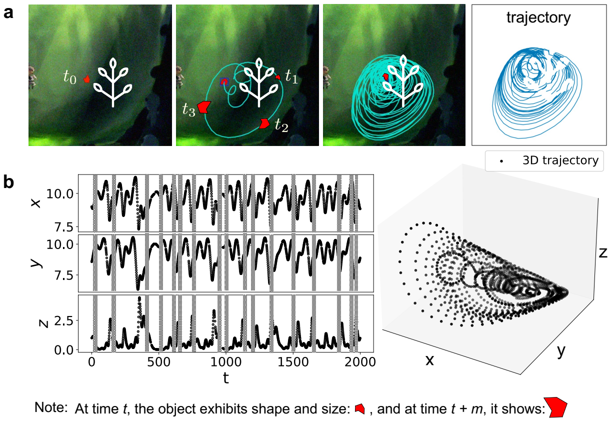


Figure 3: The influence of noisy and missing data (e.g., random block and fiber missing) on the experimental results, using the sprootF video data as an example black (other systems can be found in Appendix H). The evaluation metrics include the ℓ_2 relative error and the number of incorrectly identified equation coefficients. We analyzed the effect of (**a**) noise levels, (**b**) random block missing rates, and (**c**) fiber missing rates, respectively, to test the model's robustness.

jectories in the world coordinate system, we also compare our approach with PySINDy [Brunton *et al.*, 2016] as the baseline model. The library of candidate functions includes combinations of system states with polynomials up to the third order. The listed results are averaged over five trials. It demonstrates that our method outperforms PySINDy on each dataset in the pre-defined coordinate system. The explicit forms of the discovered governing equations for 3D moving objects obtained using our approach can be further found in Appendix G (e.g., Table S2). It is evident from Appendix Table S2 that the discovered equations by our method align better with the ground truth. We also reconstructed the motion trajectories in a 3D space using our discovered equations compared with the actual trajectories under the same coordinate system, as shown in Figure 2. These two trajectories nearly coincide, demonstrating the feasibility of our method.

It is noted that we also tested the variations and rotations of the moving object shapes in the recorded videos (e.g., see Appendix Figure S2) and found that they have little impact on the performance of our algorithm, primarily affecting the tracking efficiency. In fact, encountering noise and situations where moving objects are occluded during the measurement process can significantly impact our experimental results. To assess the robustness of our algorithm, we selected the SprottF instance for in-depth analysis and conducted experiments under various noise levels and different data occlusion scenarios. The experimental results are detailed in Figure 3. It is seen that our approach is robust against data noise and missing, discussed in detail as follows.

Noise effect. The Gaussian noise with zero mean and unit variance at a given level (e.g., 0%, 5%, ..., 30%) is added to the generated video data. To address the issue of small coefficients being overshadowed due to significant magnitude differences, we use two evaluation metrics in a standardized coordinate system: the ℓ_2 error and the count of incorrectly identified equation coefficients. Figure 3a showcases our method's performance across various noise levels. We observe that up to a 20% noise interference, our method almost accurately identifies all correct coefficients of the governing equation. However, beyond a 30% noise level, our method's performance begins to decline.



Note: At time t, the object exhibits shape and size: , and at time $t + m$, it shows:

Ground truth: $\dot{x} = y + z - 10$
$\dot{y} = -x + 0.5y + 5.0$
$\dot{z} = x^2 - 20x - z + 100$

Discovered: $\dot{x} = y + z - 9.98$
$\dot{y} = -x + 0.5y + 5.0$
$\dot{z} = x^2 - 19.97x - z + 99.94$

Figure 4: Example of a synthetic dataset simulating real-world scenarios. **a**. An example of the generated video for an object with an irregular shape undergoing random self-rotational motion and size variations. The video frames were perturbed with a zero mean Gaussian noise (variance = 0.01), and a tree-like obstruction was introduced to further simulate real-world complexity. **b**. We reconstructed the 3D trajectory of the observed target under conditions of occlusion-induced data missing. The shading areas indicate the regions impacted by the obstruction. Our approach can reconstruct the 3D point trajectories from sparse observation points, revealing accurate discovery of the underlying governing equations. Note that the video file can be found in the supplementary material.

Random block missing data effect. To evaluate our algorithm's robustness in the presence of missing data, we consider two missing scenarios (e.g., the target is blocked in the video scene), namely, random block missing and fiber missing (see Appendix Figure S3 for example). Firstly, we randomly introduce non-overlapping occlusion blocking on the target in the video during the observation period. Each block covers 1% of the total time periods. We validate our method's performance as the number of occlusion blocks increases. The "random block rate" represents the overall occlusion time as a percentage of the total observation time. We showcase our algorithm's robustness by introducing occlusion blocks that temporarily obscure the moving object, rendering it unidentifiable (see Figure 3b). These non-overlapping occlusion blocks progressively increase in num-

Conditions	Rate (%)	Methods	Terms Found?	False Positives	ℓ_2 Error ($\times 10^{-2}$)	P (%)	R (%)
Noise	10	Ours	**Yes**	**0**	**0.77**	**100**	**100**
		Model-A	No	1	8.78	90	90
	20	Ours	**Yes**	1	**2.85**	**100**	**100**
		Model-A	No	1	17.79	80	80
Random Block	10	Ours	**Yes**	**0**	**0.77**	**100**	**100**
		Model-A	No	3	9.49	75	90
	20	Ours	**Yes**	**0**	**2.19**	**100**	**100**
		Model-A	No	1	7.67	90	90
Fiber Missing	10	Ours	**Yes**	**0**	**0.87**	**100**	**100**
		Model-A	No	3	7.88	75	90
	20	Ours	**Yes**	**0**	**1.71**	**100**	**100**
		Model-A	No	4	10.79	66.67	80

Table 2: Test results for the ablated model named Model-A (i.e., spline + SINDy) under varying noise levels, random block rates, and fiber missing rates on discovering the SprottF equations.

ber, simulating higher occlusion rates. Remarkably, our algorithm remains highly robust even with up to 50% data loss due to occlusion.

Fiber missing data effect. Additionally, we conducted tests for scenarios involving continuous missing data (defined as fiber missing). By introducing 5 non-overlapping occlusion blocks randomly throughout the observation period, we varied the occlusion duration of each block, quantified by the "fiber missing rate" – the ratio of continuous missing data to the overall data volume. In Figure 3c, we explore the impact of increasing occlusion duration per block while maintaining a constant number of randomly selected occlusion blocks. All results are averaged over five trials. Our model demonstrates strong stability even when the fiber missing rate is about 20%.

Simulating real-world scenario. Furthermore, we generated a synthetic video dataset simulating real-world scenarios. Here, we modeled the observed object as an irregular shape undergoing random self-rotational motion and size variations, as shown in Figure 4a. Note that the size variations simulate changes in the camera's focal length when capturing the moving object in depth. The video frames were perturbed with a zero mean Gaussian noise (variance = 0.01). Moreover, a tree-like obstruction was introduced to further simulate the real-world complexity (e.g., the object might be obscured during motion) as depicted in Figure 4b. Despite these challenges, our method can discover the governing equations of the moving object in the reference coordinate system, showing its potential in practical applications. Please refer to Appendix I for more details.

Overall, our algorithm proves robust in scenarios with unexpected data noise, multiple instances of data loss, and continuous data gaps, for uncovering governing laws of dynamics for a moving object in a 3D space based on raw videos.

4.2 Ablation Study

We performed an ablation study to validate whether the physics component in the spline-enhanced library-based sparse regressor module is effective. Hence, we introduced an ablated model named Model-A (e.g., fully decoupled "spline + SINDy" approach). We first employed the cubic splines to interpolate the 3D trajectory in each dimension and then computed the time derivatives of the fitted trajectory points based on spline differentiation. These trajectories and the estimated derivatives are then fed into the SINDy model for equation discovery. Taking the instance of SprootF as an example, we show in Table **??** the performance of the ablated model under varying noise levels, random block rates, and fiber missing rates. It is observed that the performance of the ablated model deteriorates in all considered cases. Hence, we can ascertain that the physics-informed spline learning in the library-based sparse regressor module plays a crucial role in equation discovery under imperfect data conditions.

4.3 Discussion and Limitations

The above results show that our approach can effectively uncover the governing equations of a moving target in a 3D space directly from a set of recorded videos. The false positives of identification, when in the presence (e.g., see Appendix Table S2), are all small constants. We consider these errors to be within a reasonable range. This is because the camera pixels can only take approximate integer values, and factors such as the size of pixels captured by the camera and the number of cameras capturing the moving object can affect the reconstruction of the 3D coordinates in the reference coordinate system. The experimental results can be further improved when high-resolution videos are recorded and more cameras are used. There is an affine transformation relationship between the artificially set reference coordinate system and the actual one. Potential errors in learning such a relationship also lead to false positives in equation discovery.

Despite efficacy, our model has some limitations. The library-based sparse regression technique encounters a bottleneck when identifying very complex equations black (e.g., power or division terms) when the a priori knowledge of the candidate terms is deficient. We plan to integrate symbolic regression techniques to tackle this challenge. Furthermore, the present study only focuses on discovering the 3D dynamics of a single moving target in a video scene. In the future, we will try discovering dynamics for multiple moving objects (inter-coupled or independent).

5 Conclusion

We proposed a vision-based method to distill the governing equations for nonlinear dynamics of a moving object in a 3D space, solely from video data captured by a set of three cameras. By leveraging geometric transformations in a 3D space, combined with Rodrigues' rotation formula and computer vision techniques to track the object's motion, we can learn and reconstruct the 3D coordinates of the moving object in a user-defined coordinate system with the calibration of only one camera. Building upon this, we introduced an adaptive spline learning framework integrated with a library-based sparse regressor to identify the underlying law of dynamics. This framework can effectively handle challenges posed by partially missing and noisy data, successfully uncovering the governing equations of the moving target in a predefined reference coordinate system. The efficacy of this method has been validated on synthetic videos that record the behavior of different nonlinear dynamic systems. This approach offers a novel perspective for understanding the complex dynamics of moving objects in a 3D space. We will test it on real-world recorded videos in our future study.

Acknowledgments

The work is supported by the National Natural Science Foundation of China (No. 92270118) and the Strategic Priority Research Program of the Chinese Academy of Sciences (No. XDB0620103). The authors would like to acknowledge the support from the Fundamental Research Funds for the Central Universities (No. 202230265 and No. E2EG2202X2). The Appendix of this manuscript is found in the preprint version: https://arxiv.org/pdf/2404.17865.

References

[Bongard and Lipson, 2007] Josh Bongard and Hod Lipson. Automated reverse engineering of nonlinear dynamical systems. *Proceedings of the National Academy of Sciences*, 104(24):9943–9948, 2007.

[Brunton *et al.*, 2016] Steven L Brunton, Joshua L Proctor, and J Nathan Kutz. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the National Academy of Sciences*, 113(15):3932–3937, 2016.

[Champion *et al.*, 2019] Kathleen Champion, Bethany Lusch, J Nathan Kutz, and Steven L Brunton. Data-driven discovery of coordinates and governing equations. *Proceedings of the National Academy of Sciences*, 116(45):22445–22451, 2019.

[Chen *et al.*, 2021a] Z Chen, J Mao, J Wu, KKY Wong, JB Tenenbaum, and C Gan. Grounding physical concepts of objects and events through dynamic visual reasoning. In *International Conference on Learning Representations*, 2021.

[Chen *et al.*, 2021b] Zhao Chen, Yang Liu, and Hao Sun. Physics-informed learning of governing equations from scarce data. *Nature Communications*, 12(1):6136, 2021.

[Chen *et al.*, 2022] Boyuan Chen, Kuang Huang, Sunand Raghupathi, Ishaan Chandratreya, Qiang Du, and Hod Lipson. Automated discovery of fundamental variables hidden in experimental data. *Nature Computational Science*, 2(7):433–442, 2022.

[Ciaparrone *et al.*, 2020] Gioele Ciaparrone, Francisco Luque Sánchez, Siham Tabik, Luigi Troiano, Roberto Tagliaferri, and Francisco Herrera. Deep learning in video multi-object tracking: A survey. *Neurocomputing*, 381:61–88, 2020.

[Floryan and Graham, 2022] Daniel Floryan and Michael D Graham. Data-driven discovery of intrinsic dynamics. *Nature Machine Intelligence*, 4(12):1113–1120, 2022.

[Flynn *et al.*, 2016] John Flynn, Ivan Neulander, James Philbin, and Noah Snavely. Deepstereo: Learning to predict new views from the world's imagery. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5515–5524, 2016.

[Galliani *et al.*, 2016] Silvano Galliani, Katrin Lasinger, and Konrad Schindler. Gipuma: Massively parallel multi-view stereo reconstruction. *Publikationen der Deutschen Gesellschaft für Photogrammetrie, Fernerkundung und Geoinformation e. V*, 25(361-369):2, 2016.

[Gilpin, 2021] William Gilpin. Chaos as an interpretable benchmark for forecasting and data-driven modelling. *arXiv preprint arXiv:2110.05266*, 2021.

[Huang *et al.*, 2018] Po-Han Huang, Kevin Matzen, Johannes Kopf, Narendra Ahuja, and Jia-Bin Huang. Deepmvs: Learning multi-view stereopsis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2821–2830, 2018.

[Jaques *et al.*, 2020] Miguel Jaques, Michael Burke, and Timothy Hospedales. Physics-as-inverse-graphics: Unsupervised physical parameter estimation from video. In *International Conference on Learning Representations*, 2020.

[Koza, 1994] John R Koza. Genetic programming as a means for programming computers by natural selection. *Statistics and computing*, 4:87–112, 1994.

[Li *et al.*, 2021] Mingyang Li, Zhijiang Du, Xiaoxing Ma, Wei Dong, and Yongzhuo Gao. A robot hand-eye calibration method of line laser sensor based on 3d reconstruction. *Robotics and Computer-Integrated Manufacturing*, 71:102136, 2021.

[Lu *et al.*, 2021] Qiang Lu, Fan Tao, Shuo Zhou, and Zhiguang Wang. Incorporating actor-critic in monte carlo tree search for symbolic regression. *Neural Computing and Applications*, pages 1–17, 2021.

[Luan *et al.*, 2022] Lele Luan, Yang Liu, and Hao Sun. Distilling governing laws and source input for dynamical systems from videos. *Proceedings of the 31st International Joint Conference on Artificial Intelligence*, pages 3873–3879, 2022.

[Ma *et al.*, 2019] Xinzhu Ma, Zhihui Wang, Haojie Li, Pengbo Zhang, Wanli Ouyang, and Xin Fan. Accurate monocular 3d object detection via color-embedded 3d reconstruction for autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6851–6860, 2019.

[Marvasti-Zadeh *et al.*, 2021] Seyed Mojtaba Marvasti-Zadeh, Li Cheng, Hossein Ghanei-Yakhdan, and Shohreh Kasaei. Deep learning for visual tracking: A comprehensive survey. *IEEE Transactions on Intelligent Transportation Systems*, 23(5):3943–3968, 2021.

[Mueller *et al.*, 2017] Matthias Mueller, Neil Smith, and Bernard Ghanem. Context-aware correlation filter tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1396–1404, 2017.

[Mundhenk *et al.*, 2021] T Nathan Mundhenk, Mikel Landajuela, Ruben Glatt, Claudio P Santiago, Daniel M Faissol, and Brenden K Petersen. Symbolic regression via neural-guided genetic programming population seeding. *arXiv preprint arXiv:2111.00053*, 2021.

[Nalpantidis and Gasteratos, 2011] Lazaros Nalpantidis and Antonios Gasteratos. Stereovision-based fuzzy obstacle avoidance method. *International Journal of Humanoid Robotics*, 8(01):169–183, 2011.

[Peng *et al.*, 2020] Wanli Peng, Hao Pan, He Liu, and Yi Sun. Ida-3d: Instance-depth-aware 3d object detection from stereo vision for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13015–13024, 2020.

[Petersen *et al.*, 2021] Brenden K Petersen, Mikel Landajuela Larma, Terrell N Mundhenk, Claudio Prata Santiago, Soo Kyung Kim, and Joanne Taery Kim. Deep symbolic regression: Recovering mathematical expressions from data via risk-seeking policy gradients. In *International Conference on Learning Representations*, 2021.

[Raissi *et al.*, 2019] Maziar Raissi, Paris Perdikaris, and George E Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, 2019.

[Rao *et al.*, 2022] Chengping Rao, Pu Ren, Yang Liu, and Hao Sun. Discovering nonlinear pdes from scarce data with physics-encoded learning. In *The Tenth International Conference on Learning Representations*, 2022.

[Rao *et al.*, 2023] Chengping Rao, Pu Ren, Qi Wang, Oral Buyukozturk, Hao Sun, and Yang Liu. Encoding physics to learn reaction–diffusion processes. *Nature Machine Intelligence*, 5:765–779, 2023.

[Redmon *et al.*, 2016] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 779–788, 2016.

[Rudy *et al.*, 2017] Samuel H Rudy, Steven L Brunton, Joshua L Proctor, and J Nathan Kutz. Data-driven discovery of partial differential equations. *Science Advances*, 3(4):e1602614, 2017.

[Sahoo *et al.*, 2018] Subham Sahoo, Christoph Lampert, and Georg Martius. Learning equations for extrapolation and control. In *International Conference on Machine Learning*, pages 4442–4450, 2018.

[Schmidt and Lipson, 2009] Michael Schmidt and Hod Lipson. Distilling free-form natural laws from experimental data. *Science*, 324(5923):81–85, 2009.

[Schönberger *et al.*, 2016] Johannes L Schönberger, Enliang Zheng, Jan-Michael Frahm, and Marc Pollefeys. Pixelwise view selection for unstructured multi-view stereo. In *Computer Vision–ECCV 2016: 14th European Conference*, pages 501–518. Springer, 2016.

[Sun *et al.*, 2021] Fangzheng Sun, Yang Liu, and Hao Sun. Physics-informed spline learning for nonlinear dynamics discovery. *Proceedings of the 30th International Joint Conference on Artificial Intelligence*, pages 2054–2061, 2021.

[Sun *et al.*, 2022] Luning Sun, Daniel Zhengyu Huang, Hao Sun, and Jian-Xun Wang. Bayesian spline learning for equation discovery of nonlinear dynamics with quantified uncertainty. In *Advances in Neural Information Processing Systems*, 2022.

[Sun *et al.*, 2023] Fangzheng Sun, Yang Liu, Jian-Xun Wang, and Hao Sun. Symbolic physics learner: Discovering governing equations via monte carlo tree search. In *The Eleventh International Conference on Learning Representations*, 2023.

[Udrescu and Tegmark, 2021] Silviu-Marian Udrescu and Max Tegmark. Symbolic pregression: Discovering physical laws from distorted video. *Physical Review E*, 103(4):043307, 2021.

[Wang *et al.*, 2021] Hengli Wang, Rui Fan, and Ming Liu. Cot-amflow: Adaptive modulation network with coteaching strategy for unsupervised optical flow estimation. In *Conference on Robot Learning*, pages 143–155, 2021.

[Wojke *et al.*, 2017] Nicolai Wojke, Alex Bewley, and Dietrich Paulus. Simple online and realtime tracking with a deep association metric. In *2017 IEEE International Conference on Image Processing (ICIP)*, pages 3645–3649, 2017.