

Vid2Curve: Simultaneous Camera Motion Estimation and Thin Structure Reconstruction from an RGB Video

PENG WANG, The University of Hong Kong

LINGJIE LIU, Max Planck Institute for Informatics

NENGLUN CHEN, The University of Hong Kong

HUNG-KUO CHU, National Tsing Hua University

CHRISTIAN THEOBALT, Max Planck Institute for Informatics

WENPING WANG, The University of Hong Kong



Fig. 1. This figure shows three thin structured objects reconstructed using our method, together with selected frames of the input videos. Using an RGB video as input, our method performs curve-based camera pose estimation and reconstructs complex 3D thin structures in better quality than other existing methods. See the comparisons with the other methods on the bucket model and the hanger model in Section 5.5.

Thin structures, such as wire-frame sculptures, fences, cables, power lines, and tree branches, are common in the real world. It is extremely challenging to acquire their 3D digital models using traditional image-based or depth-based reconstruction methods, because thin structures often lack distinct point features and have severe self-occlusion. We propose the first approach that simultaneously estimates camera motion and reconstructs the geometry of complex 3D thin structures in high quality from a color video captured

by a handheld camera. Specifically, we present a new curve-based approach to estimate accurate camera poses by establishing correspondences between featureless thin objects in the foreground in consecutive video frames, without requiring visual texture in the background scene to lock on. Enabled by this effective curve-based camera pose estimation strategy, we develop an iterative optimization method with tailored measures on geometry, topology as well as self-occlusion handling for reconstructing 3D thin structures. Extensive validations on a variety of thin structures show that our method achieves accurate camera pose estimation and faithful reconstruction of 3D thin structures with complex shape and topology at a level that has not been attained by other existing reconstruction methods.

CCS Concepts: • Computing methodologies → Parametric curve and surface models.

ACM Reference Format:

Peng Wang, Lingjie Liu, Nenglun Chen, Hung-Kuo Chu, Christian Theobalt, and Wenping Wang. 2020. Vid2Curve: Simultaneous Camera Motion Estimation and Thin Structure Reconstruction from an RGB Video. *ACM Trans. Graph.* 00, 0, Article 0 (2020), 12 pages. https://doi.org/0000001.0000001_2

Authors' addresses: Peng Wang, The University of Hong Kong; Lingjie Liu, Max Planck Institute for Informatics; Nenglun Chen, The University of Hong Kong; Hung-Kuo Chu, National Tsing Hua University; Christian Theobalt, Max Planck Institute for Informatics; Wenping Wang, The University of Hong Kong.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2020 Association for Computing Machinery.

0730-0301/2020/0-ART0 \$15.00

https://doi.org/0000001.0000001_2

1 INTRODUCTION

The real world is full of objects made of thin elements, such as wire-frame sculptures, fences, power lines, leafless trees, etc. Reconstructing 3D geometry of these thin objects is a critical task in various applications. For example, wire art is widely used in the design of furniture, jewelry, and sculptures. Accurate 3D scanning of wire art is important for cultural heritage preservation and for filling virtual worlds with authentic contents. Reconstruction of (leafless) trees helps robot-assisted fruit tree pruning in agriculture, and power line reconstruction helps aerial drones avoid collisions.

It is well known that reconstructing thin structures is very difficult with either image-based or depth-based methods. Traditional image-based methods perform 3D reconstruction by estimating camera poses using visual feature points in the scene and then identifying the dense point correspondences between texture features. Since thin structures lack texture features and are only a few pixels wide, classical correspondence matching methods perform poorly or fail. Besides, even small camera calibration errors may severely compromise the reconstruction accuracy of thin structures. To address this issue, some recent works leverage high-order curve primitives for image matching [Fabbri and Kimia 2010; Nurutdinova and Fitzgibbon 2015; Usumezbas et al. 2016; Xiao and Li 2005]. However, these works assume input images are pre-calibrated. This assumption is hard to meet for thin structure scanning because the rich texture features in the background needed for camera calibration often makes it hard to segment the thin structure in the foreground for reconstruction. Therefore, it would be ideal not to separate the step of camera calibration from object reconstruction, and also be able to accommodate a texture-less background for the ease of segmenting the object in the foreground.

Another class of 3D scanning methods uses RGB-D depth sensors [Dai et al. 2017; Newcombe et al. 2011]. Most of them align and integrate depth scans using a truncated signed distance field (TSDF) representation, from which a final surface can be extracted. These methods successfully scan relatively large structures and environments. However, because of high noise and low resolution of most depth cameras, and because of the limited discretization resolution of TSDFs (e.g., on voxel grids), they fail to capture thin structures. Liu et al. [2018] presented a new fusion primitive, curve skeletons, tailored to reliable thin structure scanning with depth cameras. However, infrared-based active RGB-D cameras are not applicable under all scene conditions, such as in strong light outdoors or when the object has black surface color.

In this work, we present the first approach that simultaneously estimates camera poses, as well as geometry and structure of complex thin 3D structures from handheld RGB video. See Figure 1 for some thin structures reconstructed by our method. Our method uses a new iterative optimization scheme to compute camera poses and 3D geometry and structure in a progressive manner. It first extracts 2D curves in each input video frame. After proper initialization, we solve for the camera poses and the 3D curve network by minimizing the total distance between the projections of the curve networks in each input view and their corresponding curves extracted from the same input images. Note that we add the input view one by one progressively. As more frames are processed, the camera poses

are updated, and the 3D curve network is refined in an alternating manner. In this process, we retain the junctions and the connection of points to recover the structure of the reconstruction. Note that our method estimates camera poses solely based on thin structure objects, i.e., without requiring textured backgrounds.

To summarize, we propose a novel method for reconstructing thin structures with an RGB video as input. Our approach has two main technical contributions.

- We develop a new curve-based method for efficiently and accurately computing camera parameters. This method automatically establishes the correspondence between curves in different image frames, without requiring any initial camera poses or assuming the presence of point features or texture features. This solution to the problem of automatic camera pose estimation from curves is significant in its own right.
- Equipped with this camera pose estimation technique, we design an effective iterative optimization method for reconstructing 3D wire models based on a point cloud representation. This method achieves high-fidelity reconstruction of very complex 3D wire models that have previously not been possible using a single commodity RGB camera.

2 RELATED WORK

2.1 Thin Structure Reconstruction

The rapid development of scanner technology (e.g., structured light, LiDAR, and commodity RGBD camera) has motivated a large body of work on surface scanning and reconstruction (see [Berger et al. 2014] for a comprehensive survey). While most methods work on *extended surfaces*, some methods relax this assumption to enable the reconstruction of *thin surfaces* [Aroudj et al. 2017; Savinov et al. 2016; Ummenhofer and Brox 2013]. Although previous reconstruction methods have shown impressive results on smooth and textured extended surfaces, they are fragile when dealing with the thin structures that lack sufficient surface details. Moreover, scanning thin structures proves a challenging task even with advanced acquisition technologies, particularly due to limited sensor resolution [Fan et al. 2016; Wu et al. 2014; Yan et al. 2014].

Reconstructing delicate structures. For reconstructing objects made of delicate thin structures, Li et al. [2010] propose to reconstruct objects from high-quality 3D scans using a deformable model named *arterial snakes*. An alternative approach reconstructs the surface by fitting generalized cylinders to input image [Chen et al. 2013] or point cloud data [Yin et al. 2014] where the fitting process is either defined manually on the 2D image plane when the skeletal curve has a simple shape or based on a 3D point cloud [Huang et al. 2013]. However, for the thin structures with small radius, it is extremely difficult to perform extrusion to reconstruct the curve surface as in the 3-Sweep [Chen et al. 2013]. [Yin et al. 2014] requires heavy user interactions and often fails to reconstruct complex thin structures when the initial 3D curve skeleton extracted by L1-axis [Huang et al. 2013] contains topological and geometric errors.

Image-based reconstruction. Another line of works reconstructs thin structures from multiple color images. Tabb [2013] reconstructs thin structures from multiple images using silhouette probability

maps by solving a pseudo-Boolean optimization problem in a volumetric representation. The capturing system they used consists of 30 cameras, including 20 industrial-grade cameras, that are fixed on walls and ceilings surrounding the object to be scanned. [Tabb and Medeiros 2017] proposes a method that uses a robotic vision system for image acquisition to reconstruct tree branches in a real field outdoor. Both of these two methods require pre-computed camera poses as input. Martin et al. [2014] propose to reconstruct thin tubular structures (e.g., cables) using physics-based simulation of rods. A special voxel grid, called occupancy grid, is used to resolve crossing ambiguities observed in the 2D image plane. Hsiao et al. [2018] reconstruct a 3D wire art from several views of 2D line-drawings. This method uses constrained 3D path finding in a volumetric grid to resolve spatial ambiguities due to inconsistency between input line drawings. They demonstrate impressive results for a wide set of wire art models, but the voxel representation precludes reconstruction at very high accuracy. Li et al. [2018] use spatial curves generated from image edges to reconstruct thin structures. Liu et al. [2017] reconstruct 3D wire models from a small number of images using a candidate selection strategy for image correspondence. While these image-based methods produce impressive reconstructions of wire models of moderate complexity, they suffer from the error-prone camera pose prediction and their inability to handle self-occlusion of 3D wires. They thus perform poorly when reconstructing complex 3D thin structures. Yücer et al. [2016a] exploit the local gradient information in captured dense light fields to segment out thin structures [Yücer et al. 2016b] and estimate a per-view depth map for voxel-based carving and Poisson surface reconstruction [Kazhdan et al. 2006]. This method assumes that input objects have sufficient texture details for a valid depth estimation. In contrast, our method employs a curve-based method for accurate camera pose estimation and effectively handles textureless objects and self-occlusion to achieve high-fidelity reconstruction of complex 3D thin structures from a handheld color video, which has not been possible before.

Depth-based reconstruction. Several approaches use depth sensors for thin structure reconstruction. They can, therefore, resort to actively measured depth rather than difficultly matched image correspondence to infer 3D structure. However, depth camera data come with their own specific challenges, as the measured depth is often of low resolution, very noisy, and often exhibits systematic measurement biases. Depth-based reconstruction algorithms need to address these challenges specifically.

Liu et al. [2018] present *CurveFusion* that directly explores a dense 3D point cloud obtained from commodity depth sensors and utilizes curve skeleton as a fusion primitive for thin structure reconstruction given the aforementioned depth data challenges. The infrared-based depth sensors used in this method make it applicable only in some scene conditions, such as indoor environment without strong sunlight and objects with non-black surface color, which significantly limits its application. Moreover, the maximum reconstruction accuracy is bounded by the limited depth resolution of depth sensors, which often leads to missing curves or incorrect topology on the complex junctions. In contrast, our method works directly on a higher resolution RGB video to establish the correspondence using

a curve-based matching approach. Therefore it can reconstruct thin structures with higher spatial resolution.

2.2 Curve-based Structure-from-Motion

Our core algorithm builds upon the concepts of 3D reconstruction using structure-from-motion (SfM) [Schönberger and Frahm 2016; Snavely et al. 2006] and multi-view stereo (MVS) [Furukawa and Ponce 2010; Goesele et al. 2007; Huang et al. 2018; Kuhn et al. 2017; Schönberger et al. 2016]. These methods follow a general principle to establish point correspondences across images and then reconstruct a sparse set of 3D points alongside camera poses. While impressive results were achieved in reconstructing objects with reliable textured surfaces, these methods perform poorly when there are insufficient easy-to-discriminate point features in the scene. Therefore, several approaches exploit higher order features such as lines [Hofer et al. 2014; Jain et al. 2010] or curves [Fabbri and Kimia 2010; Nurutdinova and Fitzgibbon 2015; Usumezbas et al. 2016; Xiao and Li 2005] as reconstruction primitives. We refer the reader to the work by Fabbri et al. [2016] for a fundamental theory of the multi-view geometry on curves.

Existing curve-based reconstruction methods can be classified according to the assumptions they made. For instance, Berthilsson et al. [2001] assume that the reconstructed 3D curve should be occlusion-free in each view. Hong et al. [2004] focus on reflective and symmetric curves. Rao et al. [2012] present a curve-based SLAM method that is closely related to our method in the spirit of jointly estimating camera poses and thin structures from curve primitives. However, their method assumes the input of stereo images and the curves of interest are constrained to planar curves on the ground, while our method takes input images from a hand-held RGB camera and works for general 3D spatial curve networks in the setting of thin structure reconstruction. Similarly, Nurutdinova et al. [2015] assume that the correspondences between image curves are given and use curve as primitive to improve the accuracy of SfM. Recent works [Fabbri and Kimia 2010; Usumezbas et al. 2016] relax the assumptions and follow a typical workflow that reconstructs 3D curves by aggregating detected 2D curve segments across images using epipolar constraints. All the above methods share the limitation that they consider each line or curve segment individually, which possibly suffer from noise and reconstruction gaps, and do not reason about the global curve structure and connectivity of objects. In contrast, our method estimates camera poses and reconstructs a continuous 3D curve with an effective measure to handle self-occlusion. It produces faithful reconstruction of the complex thin structures and their connectivity way beyond the capability of existing methods.

3 OVERVIEW

Our goal is to extract the 3D geometry as well as structure and topology of objects composed of complex networks of thin tubular-shaped or wire-like structures of varying radius from a handheld color video. While we assume the intrinsic parameters of the camera to be known and fixed, the motion (i.e., the sequence of poses) of the camera is unknown and needs to be computed. We follow the definition used in previous approaches [Liu et al. 2017, 2018] to

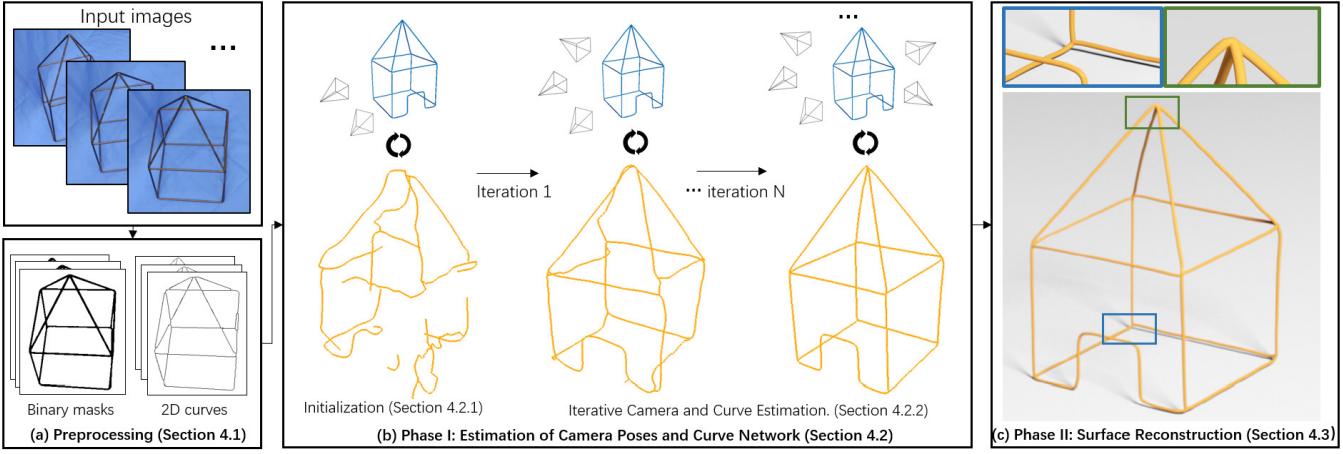


Fig. 2. Method overview. Given a sequence of RGB images of a 3D thin structure, we first segment out the structure in foreground to obtain a sequence of binary masks and corresponding one-pixel wide 2D curves in the preprocessing step (a). To solve the optimization problem formulated in Section 4.2, we first initialize the camera poses and a 3D curve network using two properly selected image frames from the input video (Section 4.2.1). (b) Then we adopt an iterative structure optimization strategy that adds and processes the remaining image frames progressively to update the camera poses of observed views so far and refine the estimated curve network in an alternating manner (Section 4.2). (c) The surface of the thin structure is modeled as a sweep surface along the constructed 3D curve with a circular section whose radius estimated from image observations (Section 4.3). The final reconstructed output is a clean, smooth thin structure with high geometric and topological fidelity to the original wire model.

define a thin structure as a collection of connected generalized cylinders, where each generalized cylinder is a sweeping surface of varying radius along its *skeletal curve*. Therefore, we represent a thin structure as a network of connected skeletal curves, to be called the *curve network* for short, with a radius function defined on the skeletal curves. Figure 2 illustrates the pipeline of our method. Given a sequence of RGB images, we first segment out the thin structure in the foreground to obtain a sequence of binary masks and corresponding one-pixel-width 2D curves in a preprocessing step (see Figure 2(a), Section 4.1). Our method then runs in two phases: *Phase I - camera pose estimation and curve network reconstruction*; and *Phase II - surface geometry reconstruction*. Phase I computes two sets of variables in an iterative optimization approach (Section 4.2): the set of extrinsic parameters defining the camera poses of all input frames, and the set of 3D points defining the curve network in 3D, along with accurate connectivity classification at junctions of the curve network. After an initialization step (Section 4.2.1), we adopt an iterative structure optimization to compute the values of the aforementioned variables by minimizing the difference between the projection of the curve network and the corresponding 2D curves observed in all input image frames (Section 4.2.2). Frames of the input video are progressively added and processed in their temporal order. When a new frame is added, a new set of 3D points and refined camera poses based on all images seen so far is estimated in an iterative optimization process that alternates between camera pose and 3D geometry computation (see Figure 2(b)).

To enable this, we developed two techniques that are critical to the success of our method. The first technique is a new correspondence finding algorithm that succeeds on textureless thin objects by operating on curve features instead of salient keypoints. It is

designed to reliably establish correspondences between the estimated 3D curve network and observed 2D curves. The second is an efficient and effective method to detect self-occlusion of thin structures in input views. It enables us to prevent unreliable 2D curve segments observed from a self-occluding perspective from being considered in the update of the corresponding 3D curve points. This significantly improves the quality of the final reconstructed curve network (Section 4.2.3).

Once all frames are processed, Phase II reconstructs the final surface of the thin structure by sweeping along skeletal curves a disc whose varying radius (or thickness) is estimated by fusing the 2D curve radius estimates across all the input image observations (see Figure 2(c), Section 4.3).

4 ALGORITHM

4.1 Preprocessing

During preprocessing, we first segment each input image to obtain a binary mask of the pixels showing the thin structure in the foreground. Reliable segmentation of thin structures from a general background is extremely challenging. We assume that the 3D thin structure is filmed in front of a simple uniform background so it can be segmented using color keying method or advanced video object segmentation tool such as Rotobrush in Adobe After Effects [Bai et al. 2009]. For the shapes of small to medium size, this can be easily achieved by filming in front of a monochromatic wall or cloth. Let $I = \{I_k, k = 1, 2, \dots, K\}$ denote the input video as a sequence of RGB frames, and let $M = \{M_k, k = 1, 2, \dots, K\}$ denote the corresponding foreground segmentation masks. We use the image thinning method to extract one pixel wide medial axis curves

from \mathbf{M} , which are called *skeletal curves* and henceforth denoted by $\mathbf{c} = \{c_k \subset \mathbb{R}^2, k = 1, 2, \dots, K\}$.

4.2 Phase I: Estimation of Camera Poses and Curve Network

We develop an optimization framework to simultaneously compute all camera poses and a 3D skeletal curve network. The camera pose of input image I_k is parameterized by the respective camera rotation R_k and translation T_k . The 3D curve network C of the entire object to be reconstructed is represented as a graph $G = (\mathbf{P}, \mathbf{E})$, where $\mathbf{P} = \{P_i \in \mathbb{R}^3, i = 1, 2, \dots, m\}$ represents a sequence of 3D points appropriately sampled on the 3D curve and the edge set \mathbf{E} encodes the connectivity of the 3D points.

We compute the camera poses, (R_k, T_k) , and the curve network, C , by minimizing an objective function that measures the sum of the squared 2D distances between the projection of the curve network, C , in an input view I_k and the corresponding 2D skeletal curve c_k , across all input views:

$$F(\{R_k, T_k\}; C) = \sum_k \text{dist}^2(c'_k, c_k), \quad (1)$$

where $c'_k = \pi(R_k, T_k; C)$ is the projection of C onto view I_k using camera extrinsic parameters (R_k, T_k) . Further, $\text{dist}^2(c'_k, c_k)$ indicates the one-sided integrated squared distance between the curve c'_k and the curve c_k in the 2D image plane, which is defined as:

$$\text{dist}^2(c'_k, c_k) = \int_{p \in c'_k} \min_{q \in c_k} \|p - q\|_2^2,$$

where $\|\cdot\|_2$ is the Euclidean norm. In the following, we explain an iterative scheme to minimize this objective function. Starting from two input images, our method progressively adds input frames one-by-one to a set of active images to estimate (R_k, T_k) , and refines C on the currently active image set in an alternating manner.

4.2.1 Initialization. Now we discuss how to initialize the camera poses and the 3D curve network in order to solve the optimization problem formulated in Equation 1. Since a proper initialization will largely facilitate the later incremental reconstruction of thin structure, we propose to select two image frames that meet the criterion of presenting sufficient camera movement such that the resulting parallax could provide sufficient depth information. Specifically, we prepare a sequence of image pairs, denoted as $\{(I_1, I_i), i = 2, 3, \dots\}$ and evaluate the image pairs successively in the following way. Given an image pair, we compute a pixel-wise correspondence between the curve segments using a 2D curve matching method [Kraevoy et al. 2009] that exploits optical flow [Kroeger et al. 2016] to estimate an initial coarse alignment between two images. Based on the correspondence induced by the matching curves, we jointly estimate the camera poses of these two frames and a 3D point cloud using the bundle adjustment [Triggs et al. 1999]. We then choose the first pair where the distance between the two initially estimated camera locations is larger than a threshold β ($\beta = 0.03$ in our experiments). Note that the distance is normalized by the average depth of the points relative to the first camera.

Curve network generation. After the initial 3D point set is generated, we connect the points into a 3D curve network using a

variant of Kruskal's algorithm to compute the minimum spanning tree [Kruskal 1956]. The goal is to build a general curve structure rather than a tree-like structure for representing a wide range of object. The network formation algorithm has the following steps:

- (1) Given a set of 3D points, \mathbf{P} , we first form a set of candidate edges \mathbf{E}' by pairing points with a mutual distance below a preset threshold θ , that is, $\mathbf{E}' = \{(P_i, P_j) | P_i, P_j \in \mathbf{P}, i \neq j, \|P_i - P_j\| < \theta\}$. We empirically set $\theta = 5\delta_0$, where δ_0 is the sampling distance defined below.
- (2) A graph $G = (\mathbf{V}, \mathbf{E})$ is maintained, with $\mathbf{V} = \mathbf{P}$ and $\mathbf{E} = \emptyset$ (empty set).
- (3) Check, in ascending order of the edge length, if the edges of \mathbf{E}' shall be added to \mathbf{E} . An edge $(P_i, P_j) \in \mathbf{E}'$ is added to \mathbf{E} if it does not form a loop in the graph G or the length of the loop formed by the edge is greater than a threshold L . We set $L = 20\delta_0$ in our experiments, where δ_0 is the sampling distance defined below.
- (4) Repeat step (3) until all edges in \mathbf{E}' have been processed.

However, only loops larger than a minimal size threshold shall be allowed in the curve network in order to avoid the formation of erroneous small loops due to noise in the point cloud.

Curve re-sampling. Since the 3D points in \mathbf{P} that define the geometry of curve network G are directly "lifted" from pixels in an input view, the distribution of the points is in general not uniform. We improve this by re-sampling points on the curve network with an even distribution. We suppose that the center of the object being scanned has unit distance to the camera center. Then the inter-point sampling distance is set to be $\delta_0 = 1/f_0$, where f_0 is the camera focus length, which is identical across all the input frames. The rationale for setting the sampling distance in this way is to ensure that, when projected onto the input view, the sample points on the 3D curve have nearly the same density as the image pixels.

Note that both aforementioned steps, curve network generation and curve re-sampling, are performed not only during initialization but also when the points are updated in every iteration of minimizing the objective function defined in Equation 1.

4.2.2 Iterative Camera and Curve Estimation.

Curve matching. A key step in solving the minimization problem in Equation 1 is our *curve matching* step that establishes the correspondence between the points of 3D curve C and those on the observed image curves c_k in each input image. A naive distance-based criterion would be finding the closest point $q_j \in c_k$ to the projection of any given point $P_j \in C$, and setting q_j as the matching point of P_j . Clearly, this greedy approach may often yield wrong matching points when the observed image curve c_k is dense and cluttered (see Figure 3(a)).

We solve this problem by combining a distance-based criterion with a constraint on curve consistency, which indicates that two consecutive sample points $\{P_i, P_{i+1} \in C\}$ should have their corresponding points $\{q_j, q_{j+1}\}$ lie on the same local branch of the image curve c_k and be close to each other (see Figure 3(b,c)).

Specifically, suppose that the 3D curve C is represented by the point sequence $\mathbf{P} = \{P_i \in \mathbb{R}^3, i = 1, 2, \dots, m\}$ and the observed image curve c_k in the image I_k is represented by the pixel sequence

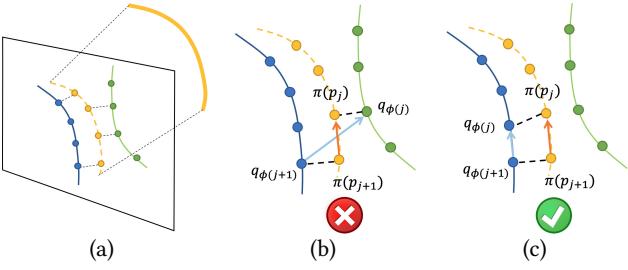


Fig. 3. **Illustration of curve matching.** (a) A naive closest point method may lead to (b) inconsistency matching between the projection of 3D curve and the 2D curves; (c) Correct matching by considering both distance and curve consistency.

$Q = \{q_i \in \mathbb{R}^2, i = 1, 2, \dots, n\}$. A matching between P and Q is then defined by an injection ϕ from $[1, m]$ into $[1, n]$. Let Φ denote the set of all such injections. Then we seek the optimal matching ϕ_0 that minimizes an objective function defined as follows:

$$E_{match}(C, c_k) = \alpha \sum_j \|\pi(P_j) - q_{\phi(j)}\| + \sum_j \|(\pi(P_j) - \pi(P_{j+1})) - (q_{\phi(j)} - q_{\phi(j+1)})\|. \quad (2)$$

We empirically set $\alpha = 0.1$ in our experiments. This minimization problem is solved efficiently with dynamic programming. Given a point $P_i \in C$, we select the candidates for P_j 's matching points only among those pixels $q_j \in c_k$ lying within a circle of radius of 10 pixels centered at the projected point $\pi(P_i)$.

Optimization strategy. There are two sets of variables in the objective function (see Equation 1): (1) the camera pose parameters (R_k, T_k) and (2) the sample points of the curve network C . To simplify the optimization task, we process the camera poses in a progressive manner so that a new camera view is added after the camera poses of all the preceding views have been estimated. Similarly, the curve network is also refined in a progressive manner.

Based on the discrete representation of the curve network, we can rewrite the Equation 1 as:

$$\tilde{F}(\{R_k, T_k\}; C) = \sum_k \sum_{P_j \in C} \mathbb{I}_{occ}(P_j, I_k) \cdot e_{k,j} \quad (3)$$

where the distance error term

$$e_{k,j} = \min_{q \in c_k} \|\pi(R_k P_j + T_k) - q\|_2^2 \quad (4)$$

is the squared distance from the projection of a variable point, denoted as $p'_j = \pi(R_k P_j + T_k)$, to the observed image curve c_k . And $\mathbb{I}_{occ}(P_j, I_k)$ is an indicator function that returns value 0 if the point P_j fails to pass the self-occlusion test with respect to the input view I_k , and otherwise returns value 1. The details of self-occlusion handling process are elaborated in Section 4.2.3.

In practice, we adopt a commonly used formulation for curve fitting [Pottmann et al. 2002] to efficiently minimize the distance error term $e_{k,j}$ in Equation 4 by expressing it as a special linear combination of the squared distances of p'_j to the image curve c_k along the normal direction, n_j , and tangent direction, t_j , at the

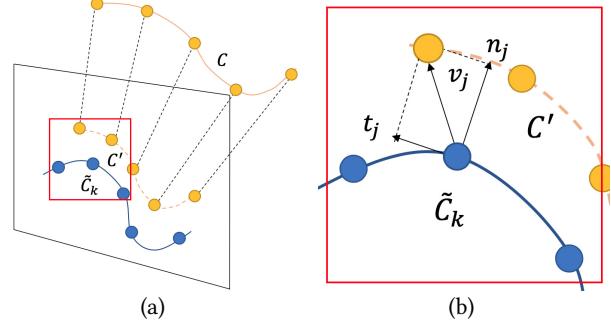


Fig. 4. (a) The projection c' of 3D curve C and its corresponding image curve c_k ; (b) The distance error term $e_{k,j}$ is expressed as a linear combination of squared distances along the tangent direction t_j and the normal direction n_j , respectively.

corresponding point $q_j \in c_k$ (see Figure 4). Thus, by denoting $v_j = \pi(R_k P_j + T_k) - q_j$, Equation 3 can be further expressed as:

$$\tilde{F}^*(\{R_k, T_k\}; C) = \sum_k \sum_{P_j \in C} \mathbb{I}_{occ}(P_j, I_k) \cdot e_{k,j}^* + F_s(C) \quad (5)$$

where the distance error term

$$e_{k,j}^* = \|v_j \cdot n_j\|^2 + w \|v_j \cdot t_j\|^2 \quad (6)$$

and the regularization term

$$F_s(C) = \lambda \sum_k \sum_{P_j \in C} \|P_{j+1} - 2P_j + P_{j-1}\|^2 \quad (7)$$

We set the weight parameter $\lambda = (\frac{2.5}{\delta_0})^2$ and w with value 0.5 in our experiments for a trade-off between efficiency and stability of convergence.

We minimize the objective function \tilde{F}^* iteratively. In each iteration, \tilde{F}^* is minimized in an alternating fashion that first optimizes the camera poses $\{(R_k, T_k)\}$ while fixing the curve points $\{p_j\}$, and then optimizes the curve points while fixing the camera poses. The process iterates until the curve points stop moving. Note that, once

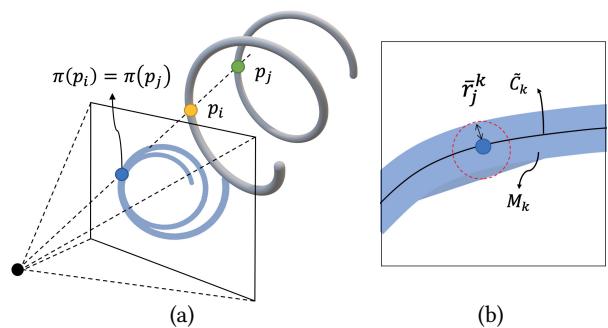


Fig. 5. (a) Illustration of self-occlusion: Two distinct points, p_i and p_j , on a non-self-intersecting 3D curve C are projected to the same intersecting point $\pi(p_i) = \pi(p_j)$ in the view plane; (b) The width $2\bar{r}_j^k$ of the projected point of 3D thin structure is estimated by the corresponding 2D curve c_k and segmentation mask M_k in the view of I_k .



Fig. 6. **Reconstruction results.** The gallery of real world 3D thin structures reconstructed using our method. Our method reconstructs a wide variety of wire objects in high quality.

the camera poses $\{(R_k, T_k)\}$ and the points $\{P_j\}$ have been updated, we re-compute the corresponding point q_j on the image curve c_k to the updated projected point p'_j by our matching algorithm before entering the next iteration.

4.2.3 Handling Self-occlusion. A *self-occlusion* occurs when distinct parts of a 3D curve network C are projected onto the same location in a 2D view I_k . When this happens, the 3D curve C intersects itself in the projected view where the intersecting pixels correspond to multiple points of C , causing ambiguity during the reconstruction (see Figure 5(a)). Hence, it is critical to detect self-occlusion to ensure that image observations involving self-occlusion are not used for 3D curve reconstruction.

To determine whether the points of 3D curve C are subject to self-occlusion in a certain image I_k , we perform the following steps:

- (1) For each point $P_j \in C$ and its matching point $q_{\phi(j)} \in c_k$ paired in the curve matching step, we examine the neighboring pixels of $q_{\phi(j)}$ using a 3×3 local window. We then generate a 3D point set \hat{P} that contains whatever points of C that match the pixels within the local window.
- (2) We then compute a spatial compactness factor as the standard deviation σ_j derived from the average L2 distance between the points in \hat{P} and their centroid.
- (3) Point P_j is labeled as self-occluded (i.e., $\mathbb{I}_{occ}(P_j, I_k) = 0$) if $\sigma_j \geq \sigma_0$. Otherwise if $\sigma < \sigma_0$ and all the points in \hat{P} lie on the same local branch of the 3D curve C , the point is not self-occluded (i.e., $\mathbb{I}_{occ}(P_j, I_k) = 1$).

We use a preset threshold $\sigma_0 = 10\delta_0$ in our experiments, where δ_0 is the sampling distance defined in Sec. 4.2.1. We employ this

self-occlusion handling not only during the 3D curve reconstruction but also during the surface reconstruction in Phase II in order to obtain reliable radius estimates as discussed below.

4.3 Phase II: Surface Geometry Reconstruction

To generate the generalized cylinder (or sweeping surface) representing the surface of the reconstructed thin structure, we compute the radius at each point of the 3D curve using the corresponding image observations from all the input views. Specifically, for each point P_j of 3D curve C , we first find its matching point $q_{\phi(j)} \in c_k$. Note that the projection of a generalized cylinder is a 2D strip in an input image. Therefore we compute the width of the strip at the point $q_{\phi(j)}$ by computing the distances from $q_{\phi(j)}$ to both sides of the strip defined in the foreground segmentation mask M_k . We denote the estimated width by $2\bar{r}_j^k$ (see Figure 5(b)). Then the radius of the thin structure at P_j in the image view I_k is defined as:

$$r_j^k = \frac{\bar{r}_j^k}{f_0} \cdot \text{depth}(P_j, I_k), \quad (8)$$

where f_0 is the focal length of the camera and $\text{depth}(P_j, I_k)$ is the depth of P_j with respect to the image I_k . The final estimated radius at P_j is thus defined as the average of the r_j^k over all the input images, excluding those image observations that involve self-occlusion at P_j , i.e., $\mathbb{I}_{occ}(P_j, I_k) = 0$.

5 EXPERIMENTAL RESULTS

We evaluate our method on a wide variety of real world thin structures with varying scale and complexity, e.g., from man-made wire sculptures to leafless tree branches. The input data is captured by a



Fig. 7. Reconstruction of synthetic models. The synthetic dataset contains nine wire models. Four of these models, shown in the bottom row, have a varying thickness and are used for validation as summarized in Table 1 and Table 2.

hand-held camera with known intrinsic parameters. The length of each video ranges from 20 to 30 seconds, depending on the scale and complexity of the objects, and we downsample the videos with a sampling rate of 5 frames to produce 100 ~ 300 RGB images as input frames to our method. To facilitate the foreground segmentation, all the objects were captured against a clean background.

Figure 6 shows a gallery of real world 3D thin structures reconstructed using our method. We refer the reader to the supplementary video for more results, with close-ups. These results demonstrate the effectiveness of our method in estimating accurate cameras poses as well as producing high-fidelity reconstructions of 3D thin structures with complex geometry and topology.

5.1 Quantitative Evaluations

To quantitatively evaluate our method, we prepared a dataset of synthetic wire models, among which Cube, Text, Bimbo and Bimbo_Thin models were created by ourselves and the others are from Mehra et al. [2009] and Xu et al. [2014]. Four wire models (Cube, Text, Bimbo and Bimbo_Thin) have varying thickness while the others have uniform thickness. Figure 7 shows the reconstruction of these digital models from the rendered images of the objects using a virtual camera moving along a simulated motion path. Only the intrinsic camera parameters are used as input; the camera path is estimated by our method and compared against the ground truth.

The following metrics are used for quality assessment:

- *Reconstruction error (RE)* measures the average of normalized closest point distances between the reconstructed model and the ground-truth in 3D.
- *Relative reconstruction error (RRE)* measures the average of normalized closest point distances between the reconstructed model and the ground-truth relative to the corresponding tube diameter of the closest point.
- *Relative pose error (RPE)* measures the pose error of an estimated camera path against its ground truth, following [Sturm et al. 2012].
- *Projection error (PE)* measures the average of normalized distances between the projection of points on the central curve and the closest points sampled on 2D central curves over all the frames.

Table 1. Quantitative evaluation on reconstruction error (RE); relative reconstruction error (RRE); projection error (PE); and topology error, including precision (TPE) and recall (TRE).

	RE	RRE	PE	TPE	TRE
Blender	0.000596	0.0775	0.0009	86/87	86/86
Cup	0.000530	0.0689	0.0008	25/26	25/25
Game Pad	0.000450	0.0505	0.0015	43/43	43/43
Japan House	0.000557	0.0690	0.0024	48/49	48/50
Pencil Holder	0.000578	0.0831	0.0011	45/46	45/45
Cube	0.000511	0.0835	0.0010	98/98	98/98
Text	0.000871	0.0960	0.0012	13/13	13/13
Bimbo	0.000539	0.0636	0.0009	172/172	172/175
Bimbo_Thin	0.000298	0.1726	0.0008	174/174	174/175

Table 2. Quantitative evaluation on estimating camera pose error (RPE).

	RPE ($\Delta = 30$)	Path length ($\Delta = 30$)
Blender	0.0078	2.4848
Cup	0.0031	1.1782
Game Pad	0.0125	1.7534
Japan House	0.0401	2.9084
Pencil Holder	0.0083	2.7369
Cube	0.0109	1.3501
Text	0.0119	0.6947
Bimbo	0.0066	1.9390
Bimbo_Thin	0.0031	1.9453

- *Topology error* includes two metrics: *precision (TPE)* and *recall (TRE)*. TPE measures the fraction of correctly reconstructed junctions in the reconstructed model and TRE measures the fraction of correctly reconstructed junctions in the ground-truth model.

Note that the above distance measures in 3D and 2D are normalized by the diagonal length of the bounding box of the synthetic model in 3D and in the 2D projected view, respectively.

Table 1 reports the statistics on the reconstruction errors. It shows that our algorithm produces faithful 3D reconstruction with correct recovery of 3D topology. The reconstruction errors (RE) are less than 0.001 and the projection errors (PE) are less than 0.003. The relative reconstruction errors (RRE) are less than 10% except for the Bimbo_Thin model, whose curve width, after projecting to the images, is 1-3 pixels, while the width of the other models ranges from 5 to 10 pixels. Table 2 reports the relative pose errors (RPE) [Sturm et al. 2012] measured on the nine estimated camera paths. Note that the RPE is defined with respect to the frame interval Δ , which is set to 30 in this evaluation. That is, the RPE here is the averaged camera pose error (i.e. drifting) during the span of 30 frames along the camera motion path. Table 2 also reports the averaged length of camera displacement during the span of 30 frames. It follows that, on average, the relative camera pose error accumulated over 30 frames is less than 2% of the average length of camera motion during 30 frames.

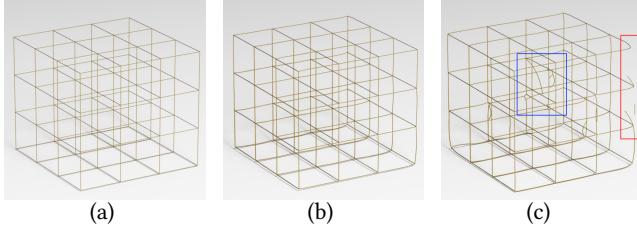


Fig. 8. Effect of curve matching. (a) Input synthetic $3 \times 3 \times 3$ grid model. (b) Reconstruction using our curve matching algorithm. (c) Reconstruction using a naïve closest point search algorithm, yielding notable artifacts, such as missing parts (red box) and redundant parts (blue box).

5.2 Ablation Studies

Effect of Curve Matching. The tailored curve matching algorithm discussed in Section 4.2.2 plays a crucial part in reconstructing accurate curve skeletons. To validate its effectiveness, we conducted an experiment that compares our curve matching method with a naïve closest point search algorithm on reconstructing a synthetic $3 \times 3 \times 3$ grid model (see Figure 8(a)). As shown in Figure 8(c), the naïve method caused obvious artifacts such as missing and redundant parts, while our method produces high-fidelity reconstruction (see Figure 8(b)).

Effect of Self-occlusion Detection. As discussed in Section 4.2.3, the ability to detect and handle self-occlusion is key to robust reconstruction of thin structures. To show the impact of self-occlusion detection on the final reconstruction, we took the video of a real basket model. This input video has severe self-occlusion in over half of its frames (see Figure 9(a)). We tested our system with this input video with and without enabling the self-occlusion detection, and show the respective reconstruction results in Figure 9(b) and Figure 9(c). Clearly, there are significant reconstruction errors in both geometry and topology when self-occlusion detection is turned off.

5.3 Sensitivity to Camera Shake

Camera shake caused by unsteady hands or other motions during video capturing leads to blurry input images. To quantitatively evaluate how well our method could resist camera shake, we conducted an experiment using synthetic 3D wire models. Specifically, we rendered 200 images of the Japan House model and used a non-uniform blurring model [Whyte et al. 2012] to simulate camera shake effect

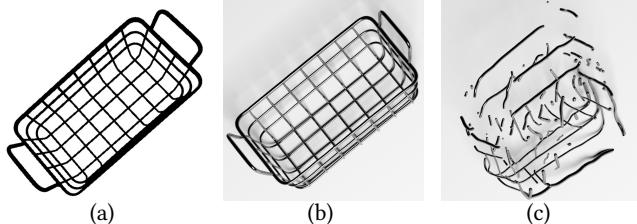


Fig. 9. Effect of self-occlusion detection. (a) An input frame with severe self-occlusion. (b) Reconstruction result with self-occlusion detection. (c) Reconstruction result without self-occlusion detection.

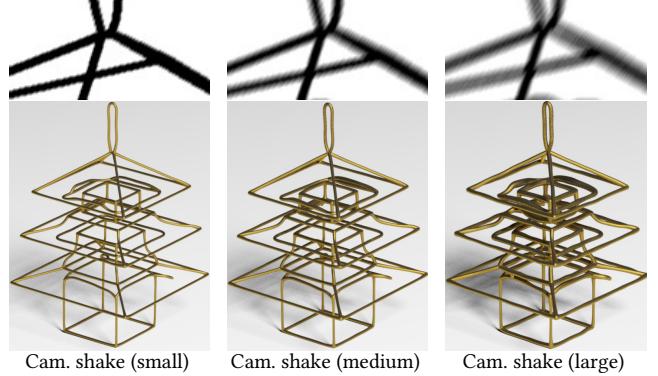


Fig. 10. Sensitivity to camera shake. (Top) Blurry input frames due to different degrees of camera shake. (Bottom) Corresponding reconstruction results.

as shown in Figure 10(top). With this blurring model, the rotation of a camera is represented as Euler angle and we added different degrees of perturbations to each angle by sampling a Gaussian distribution with zero mean and varying standard deviations of 0.1° (small), 0.3° (medium) and 0.5° (large). Figure 10 shows the reconstruction results with different degrees of image blurring and the quantitative results are reported in Table 3. We can see that our method can resist a certain degree of camera shake. In the case of large camera shake, the blurry boundaries cause relatively large estimation errors in image segmentation, which in turn results in increased reconstruction errors in terms of both RE and RRE.

5.4 Sensitivity to Segmentation Noise

To evaluate the sensitivity of our method to the quality of the segmentation mask, we conducted an experiment that adds different levels of noise to the segmentation boundary of the Japan House model (see Figure 11(top)). The noise is added by sampling a Gaussian distribution with zero mean and standard deviations of 0.3 (low), 0.5 (medium) and 0.7 (high). The reconstruction results are shown

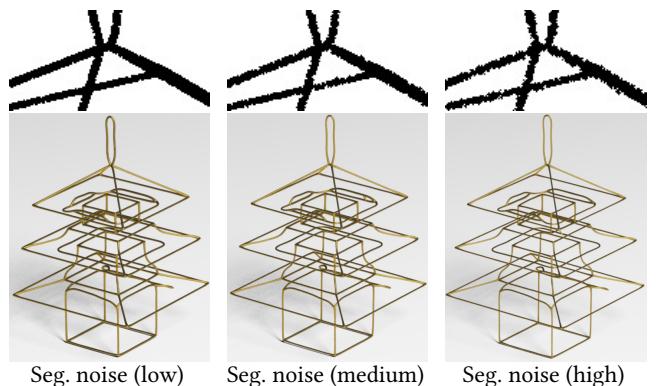


Fig. 11. Sensitivity to segmentation noise. (Top) The input segmentation masks after applying different amounts of boundary noise. (Bottom) Our reconstruction results.

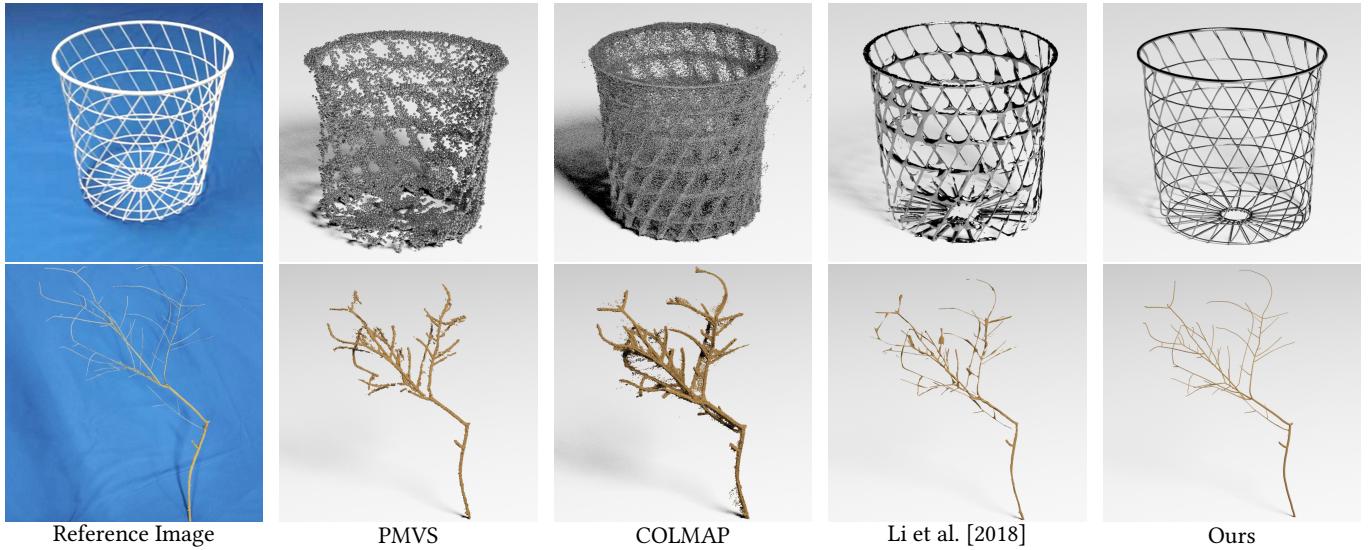


Fig. 12. Comparisons with three multi-view stereo (MVS) algorithms: PMVS [Furukawa and Ponce 2010], COLMAP [Schönberger and Frahm 2016; Schönberger et al. 2016], and Li et al. [Li et al. 2018].

Table 3. Quantitative evaluation on the sensitivity to different degrees of camera shake (upper block) and segmentation noises (lower block). Please refer to Section 5.1 for the definition of error metrics.

	RE	RRE	PE	TPE	TRE
Cam. shake (small)	0.000651	0.0807	0.0014	48/48	48/50
Cam. shake (medium)	0.001144	0.1418	0.0024	44/47	44/50
Cam. shake (large)	0.003349	0.4163	0.0043	41/47	41/50
Seg. noise (low)	0.000765	0.0947	0.0009	48/49	48/50
Seg. noise (medium)	0.000891	0.1102	0.0009	50/52	50/50
Seg. noise (high)	0.001168	0.1445	0.0008	50/51	50/50

in Figure 11(bottom) and the quantitative results are reported in Table 3. The results demonstrate the robustness of our method to a certain level of segmentation noise.

5.5 Comparisons with Baselines

We first compare our method with three multi-view stereo (MVS) algorithms, PMVS [Furukawa and Ponce 2010], COLMAP [Schönberger and Frahm 2016; Schönberger et al. 2016], and Li et al. [Li et al. 2018], on a bucket model and a model of leafless tree branches. The number of input frames used as input for all these methods is 182 for each object. Note that these methods all require rich background texture features for camera pose estimation. Therefore, for fair comparison, their input videos are taken to contain some textured objects in the background. As shown in Figure 12, the reconstructions by these methods contain significant noise and missing parts. In addition to visual comparison, we also conducted a quantitative evaluation on the bucket model shown in Figure 12 (top row). For the quality assessment, we employ the commonly used re-projection error that measures the average distance between the projected 3D reconstruction and 2D image segmentation over all the input frames. The error of our reconstruction is less than 0.0007,

which is smaller than the errors of the other three methods: Li et al. (0.0015), PMVS (0.0034), and COLMAP (0.0023). Here the error value is normalized by the diagonal length of the 2D bounding box of the wire model in each 2D projected view.

We further compare our method with a recent image-based method for 3D wire reconstruction [Liu et al. 2017], which uses three images along with intrinsic camera parameters and camera poses as input. In our experiment the three input images were captured using the same viewpoint setting reported in their paper to avoid self-occlusion as much as possible in these views.

As shown in Figure 13 (middle), the method in [Liu et al. 2017] fails for a wire model with a repetitive curve structure, while our method reconstructs the model successfully.

Figure 14 shows a comparison to CurveFusion [Liu et al. 2018], an RGBD video-based method for thin structure reconstruction, and a state-of-the-art visual-silhouette based method by Tabb [Tabb 2013]. For the CurveFusion method, the input is an RGBD sequence consisting of 261 frames. For the visual-silhouette based method, the author helped us to take photos of the wire model from 144 pre-calibrated camera positions. It can be seen that the reconstruction by [Tabb 2013] is noisy and irregular. Although the CurveFusion method shows impressive reconstruction quality, its result contains obvious defects, such as missing curves and incorrect topology, caused by

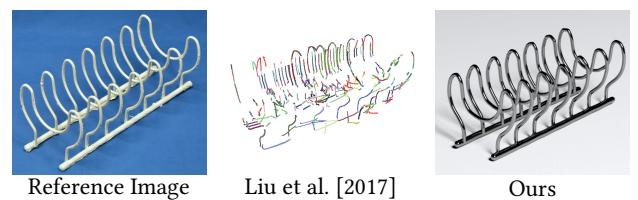


Fig. 13. Comparison with Liu et al. [2017]

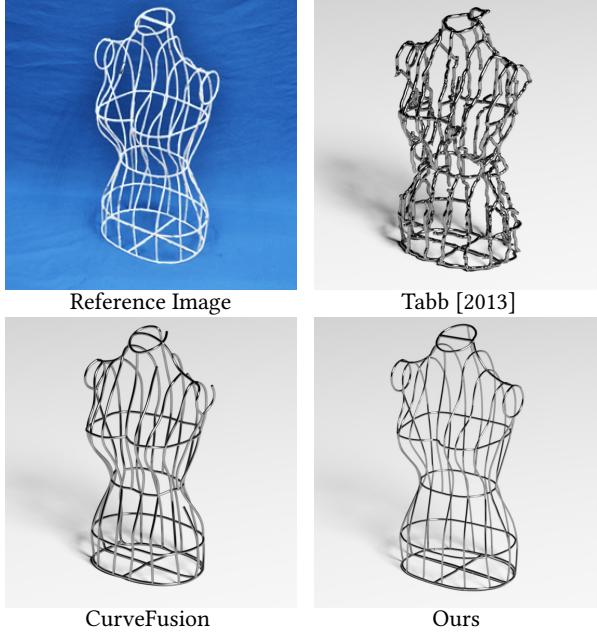


Fig. 14. Comparison with Tabb [Tabb 2013] and CurveFusion [Liu et al. 2018].

the limited resolution of the commodity depth sensor used. Moreover, well-known limitations of the infrared-based depth sensors are that it does not work outdoor in strong sunlight and that objects with black surface cannot be scanned. In contrast, our method is able to handle these situations and produces reconstruction results with superior quality.

5.6 Performance

Here we report the performances of our method on a machine with Intel i5-8300H 2.3 GHz CPU and 16G RAM. The overall computation time is proportional to the number of input images and the complexity of wire models. For a simple wire model, such as the Fat Cat model (see Figure 6(bottom-left)) with 124 input frames, our method took 138 seconds to reconstruct the model. The computation times increases to 25 minutes for reconstructing a more complex example, such as the Bucket model (see Figure 1(left)) with 229 input frames. For a breakdown, the initialization step (Section 4.2.1) and surface reconstruction step (Section 4.3) take only 3% and 2% of the total time, respectively. The main bottleneck lies in the process of iterative structure optimization (Section 4.2.2), which takes about 95% of the total running time.

5.7 Limitations

Our method requires that an input video be captured with a simple or clean background for easy and accurate foreground segmentation. Therefore, future work is needed to enable robust foreground segmentation of wire models against a general background. Another limitation is that our method assumes that wire models are made of tubular wires with circular sections. Therefore it cannot accurately reconstruct wire models with non-circular sections.

6 CONCLUSIONS

We have proposed a new method for high quality reconstruction of 3D thin structures from an RGB video. Our method does not require camera poses as input but uses only curve feature correspondence to accurately estimate the camera poses of the input image frames. This allows us to reconstruct more complex thin structure in higher quality than the previous method. As future work, we would like to study the segmentation of thin structures against a general background so to be able to apply the proposed reconstruction pipeline in a natural setting. Another extension is to consider the reconstruction of general objects consisting of thin structure components as well as extended surfaces.

ACKNOWLEDGMENTS

We would like to thank the reviewers for their valuable and insightful comments. We also thank Shiwei Li, Amy Tabb, Raleeb Zayer for their help with experiments. This work was partially funded by the Research Grant Council of Hong Kong (GRF 17210718), ERC Consolidator Grant 770784, Lise Meitner Postdoctoral Fellowship, Ministry of Science and Technology of Taiwan (108-2218-E-007-050 and 107-2221-E-007-088-MY3).

REFERENCES

- Samir Aroudj, Patrick Seemann, Fabian Langguth, Stefan Guthe, and Michael Goesele. 2017. Visibility-Consistent Thin Surface Reconstruction Using Multi-Scale Kernels. *ACM SIGGRAPH Asia* 36, 6 (2017), 187:1–187:13.
- Xue Bai, Jue Wang, David Simons, and Guillermo Sapiro. 2009. Video SnapCut: Robust Video Object Cutout Using Localized Classifiers. *ACM SIGGRAPH* 28, 3, Article 70 (2009), 11 pages.
- M. Berger, A. Tagliasacchi, L. Seversky, P. Alliez, J. Levine, A. Sharf, and C. Silva. 2014. State of the Art in Surface Reconstruction from Point Clouds. In *CGF Eurographics - State of the Art Reports*. <https://doi.org/10.2312/egst.20141040>
- Rikard Berthilsson, Kalle Astrom, and Anders Heyden. 2001. Reconstruction of General Curves, Using Factorization and Bundle Adjustment. *IJCV* 41, 3 (2001), 171–182.
- Tao Chen, Zhe Zhu, Ariel Shamir, Shi-Min Hu, and Daniel Cohen-Or. 2013. 3-Sweep: Extracting Editable Objects from a Single Photo. *ACM SIGGRAPH Asia*, Article 195 (2013), 10 pages.
- Angela Dai, Matthias Nießner, Michael Zollhöfer, Shahram Izadi, and Christian Theobalt. 2017. BundleFusion: Real-Time Globally Consistent 3D Reconstruction Using On-the-Fly Surface Reintegration. *ACM TOG* 36, 3, Article 24 (2017), 18 pages.
- R. Fabbri and B. Kimia. 2010. 3D Curve Sketch: Flexible Curve-based Stereo Reconstruction and Calibration. In *IEEE CVPR*. 1538–1545.
- Ricardo Fabbri and Benjamin B. Kimia. 2016. Multiview Differential Geometry of Curves. *IJCV* 120, 3 (2016), 324–346.
- Xinyi Fan, Linguang Zhang, Benedict Brown, and Szymon Rusinkiewicz. 2016. Automated View and Path Planning for Scalable Multi-object 3D Scanning. *ACM SIGGRAPH Asia*, Article 239 (2016), 13 pages.
- Yasutaka Furukawa and Jean Ponce. 2010. Accurate, Dense, and Robust Multiview Stereopsis. *IEEE PAMI* 32, 8 (2010), 1362–1376.
- M. Goesele, N. Snavely, B. Curless, H. Hoppe, and S. M. Seitz. 2007. Multi-View Stereo for Community Photo Collections. In *IEEE ICCV*. 1–8.
- Manuel Hofer, Michael Maurer, and Horst Bischof. 2014. Improving Sparse 3D Models for Man-Made Environments Using Line-Based 3D Reconstruction. In *International Conference on 3D Vision (3DV)*.
- Wei Hong, Yi Ma, and Yizhou Yu. 2004. Reconstruction of 3-D Symmetric Curves from Perspective Images without Discrete Features. In *EECV*. 533–545.
- Kai-Wen Hsiao, Jia-Bin Huang, and Hung-Kuo Chu. 2018. Multi-view Wire Art. *ACM SIGGRAPH Asia* 37, 6, Article 242 (2018), 11 pages.
- Hui Huang, Shihao Wu, Daniel Cohen-Or, Minglin Gong, Hao Zhang, Guiqing Li, and Baquan Chen. 2013. L1-medial Skeleton of Point Cloud. *ACM SIGGRAPH*, Article 65 (2013), 8 pages.
- Po-Han Huang, Kevin Matzen, Johannes Kopf, Narendra Ahuja, and Jia-Bin Huang. 2018. DeepMVS: Learning Multi-view Stereopsis. In *IEEE CVPR*.
- Arjun Jain, Christian Kurz, Thorsten Thormählen, and Hans-Peter Seidel. 2010. Exploiting Global Connectivity Constraints for Reconstruction of 3D Line Segment from Images. In *IEEE CVPR*.

- Michael Kazhdan, Matthew Bolitho, and Hugues Hoppe. 2006. Poisson Surface Reconstruction. In *SGP*, 61–70.
- Vladislav Kraevoy, Alla Sheffer, and Michiel van de Panne. 2009. Modeling from Contour Drawings. In *CGF Eurographics*. ACM, 37–44.
- Till Kroeger, Radu Timofte, Dengxin Dai, and Luc Van Gool. 2016. Fast Optical Flow Using Dense Inverse Search. In *ECCV*. Springer, 471–488.
- Joseph B Kruskal. 1956. On the Shortest Spanning Subtree of a Graph and the Traveling Salesman Problem. *Proceedings of the American Mathematical Society* 7, 1 (1956), 48–50.
- Andreas Kuhn, Heiko Hirschmüller, Daniel Scharstein, and Helmut Mayer. 2017. A TV Prior for High-quality Scalable Multi-view Stereo Reconstruction. *IJCV* 124, 1 (2017), 2–17.
- Guo Li, Ligang Liu, Hanlin Zheng, and Niloy J. Mitra. 2010. Analysis, Reconstruction and Manipulation using Arterial Snakes. *ACM SIGGRAPH Asia* 29, 6, Article 152 (2010), 10 pages.
- Shiwei Li, Yao Yao, Tian Fang, and Long Quan. 2018. Reconstructing Thin Structures of Manifold Surfaces by Integrating Spatial Curves. In *IEEE CVPR*. 2887–2896.
- Lingjie Liu, Duygu Ceylan, Cheng Lin, Wenping Wang, and Niloy J. Mitra. 2017. Image-based Reconstruction of Wire Art. *ACM SIGGRAPH* 36, 4, Article 63 (2017), 11 pages.
- Lingjie Liu, Nenglun Chen, Duygu Ceylan, Christian Theobalt, Wenping Wang, and Niloy J. Mitra. 2018. CurveFusion: Reconstructing Thin Structures from RGBD Sequences. *ACM SIGGRAPH Asia* 37, 6 (2018).
- Tobias Martin, Juan Montes, Jean-Charles Bazin, and Tiberiu Popa. 2014. Topology-aware Reconstruction of Thin Tubular Structures. In *SIGGRAPH Asia Technical Briefs*. Article 12, 4 pages.
- Ravish Mehra, Qingnan Zhou, Jeremy Long, Alla Sheffer, Amy Gooch, and Niloy J. Mitra. 2009. Abstraction of Man-made Shapes. In *ACM SIGGRAPH Asia 2009 papers*. 1–10.
- Richard A. Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J. Davison, Pushmeet Kohli, Jamie Shotton, Steve Hodges, and Andrew Fitzgibbon. 2011. KinectFusion: Real-time Dense Surface Mapping and Tracking. In *IEEE ISMAR*. 127–136.
- Irina Nurutdinova and Andrew Fitzgibbon. 2015. Towards Pointless Structure from Motion: 3D Reconstruction and Camera Parameters from General 3D Curves. In *IEEE ICCV*. 2363–2371.
- Helmut Pottmann, Stefan Leopoldseder, and Michael Hofer. 2002. Approximation with Active B-spline Curves and Surfaces. In *10th Pacific Conference on Computer Graphics and Applications, 2002. Proceedings*. IEEE, 8–25.
- Dushyant Rao, Soon-Jo Chung, and Seth Hutchinson. 2012. CurveSLAM: an Approach for Vision-based Navigation Without Point Features. In *International Conference on Intelligent Robots and Systems*. 4198–4204.
- Nikolay Savinov, Christian Hane, Lubor Ladicky, and Marc Pollefeys. 2016. Semantic 3D Reconstruction with Continuous Regularization and Ray Potentials Using a Visibility Consistency Constraint. In *IEEE CVPR*.
- Johannes Lutz Schönberger and Jan-Michael Frahm. 2016. Structure-from-Motion Revisited. In *IEEE CVPR*.
- Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. 2016. Pixelwise View Selection for Unstructured Multi-View Stereo. In *ECCV*.
- Noah Snavely, Steven M. Seitz, and Richard Szeliski. 2006. Photo Tourism: Exploring Photo Collections in 3D. In *ACM SIGGRAPH*. 835–846.
- Jürgen Sturm, Nikolas Engelhard, Felix Endres, Wolfram Burgard, and Daniel Cremers. 2012. A Benchmark for the Evaluation of RGB-D SLAM Systems. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 573–580.
- A. Tabb. 2013. Shape from Silhouette Probability Maps: Reconstruction of Thin Objects in the Presence of Silhouette Extraction and Calibration Error. In *IEEE CVPR*. 161–168. <https://doi.org/10.1109/CVPR.2013.28>
- A. Tabb and H. Medeiros. 2017. A Robotic Vision System to Measure Tree Traits. In *International Conference on Intelligent Robots and Systems (IROS)*. 6005–6012.
- Bill Triggs, Philip F McLauchlan, Richard I Hartley, and Andrew W Fitzgibbon. 1999. Bundle Adjustment—A Modern Synthesis. In *International Workshop on Vision Algorithms*. Springer, 298–372.
- B. Ummenhofer and T. Brox. 2013. Point-Based 3D Reconstruction of Thin Objects. In *IEEE ICCV*. 969–976. <https://doi.org/10.1109/ICCV.2013.124>
- Anil Usumezbas, Ricardo Fabbri, and Benjamin B. Kimia. 2016. From Multiview Image Curves to 3D Drawings. In *ECCV*. 70–87.
- Oliver Whyte, Josef Sivic, Andrew Zisserman, and Jean Ponce. 2012. Non-uniform Deblurring for Shaken Images. *IJCV* 98, 2 (2012), 168–186.
- Shihao Wu, Wei Sun, Pinxin Long, Hui Huang, Daniel Cohen-Or, Minglun Gong, Oliver Deussen, and Baoquan Chen. 2014. Quality-driven Poisson-guided Autoscaning. *ACM SIGGRAPH Asia*, Article 203 (2014), 12 pages.
- Yi Jun Xiao and Youfu Li. 2005. Optimized Stereo Reconstruction of Free-form Space Curves Based on a Nonuniform Rational B-spline Model. *J. Opt. Soc. Am. A* 22, 9 (2005), 1746–1762.
- Baoxuan Xu, William Chang, Alla Sheffer, Adrien Bousseau, James McCrae, and Karan Singh. 2014. True2Form: 3D Curve Networks from 2D Sketches via Selective Regularization. (2014).
- Feilong Yan, Andrei Sharf, Wenzhen Lin, Hui Huang, and Baoquan Chen. 2014. Proactive 3D Scanning of Inaccessible Parts. *ACM SIGGRAPH Asia*, Article 157 (2014), 8 pages.
- Kangxue Yin, Hui Huang, Hao Zhang, Minglun Gong, Daniel Cohen-Or, and Baoquan Chen. 2014. Morfit: Interactive Surface Reconstruction from Incomplete Point Clouds with Curve-driven Topology and Geometry Control. *ACM SIGGRAPH Asia*, Article 202 (2014), 12 pages.
- K. Yücer, C. Kim, A. Sorkine-Hornung, and O. Sorkine-Hornung. 2016a. Depth from Gradients in Dense Light Fields for Object Reconstruction. In *International Conference on 3D Vision (3DV)*. 249–257.
- Kaan Yücer, Alexander Sorkine-Hornung, Oliver Wang, and Olga Sorkine-Hornung. 2016b. Efficient 3D Object Segmentation from Densely Sampled Light Fields with Applications to 3D Reconstruction. *ACM TOG* 35, 3, Article 22 (2016), 15 pages.