

Image-based Reconstruction of Wire Art

LINGJIE LIU, The University of Hong Kong, University College London

DUYGU CEYLAN, Adobe Research

CHENG LIN, The University of Hong Kong

WENPING WANG, The University of Hong Kong

NILOY J. MITRA, University College London



Fig. 1. We present an image-based method to reconstruct wire objects from a few input images (3 views in our experiments). Our reconstructions faithfully capture both the 3D geometry and the topology of the wires.

Objects created by connecting and bending wires are common in furniture design, metal sculpting, wire jewelry, etc. Reconstructing such objects with traditional depth and image based methods is extremely difficult due to their unique characteristics such as lack of features, thin elements, and severe self-occlusions. We present a novel image-based method that reconstructs a set of continuous 3D wires used to create such an object, where each wire is composed of an ordered set of 3D curve segments. Our method exploits two main observations: *simplicity* - wire objects are often created using only a small number of wires, and *smoothness* - each wire is primarily smoothly bent with sharp features appearing only at joints or isolated points. In light of these observations, we tackle the challenging image correspondence problem across featureless wires by first generating multiple candidate 3D curve

segments and then solving a global selection problem that balances between image and smoothness cues to identify the correct 3D curves. Next, we recover a decomposition of such curves into a set of distinct and continuous wires by formulating a *multiple traveling salesman problem*, which finds smooth paths, *i.e.*, wires, connecting the curves. We demonstrate our method on a wide set of real examples with varying complexity and present high-fidelity results using only 3 images for each object. We provide the source code and data for our work in the project website.

CCS Concepts: • **Computing methodologies** → **3D imaging; Reconstruction; Parametric curve and surface models;**

Additional Key Words and Phrases: delicate structure, wire art, image-based reconstruction

ACM Reference format:

Lingjie Liu, Duygu Ceylan, Cheng Lin, Wenping Wang, and Niloy J. Mitra. 2017. Image-based Reconstruction of Wire Art. *ACM Trans. Graph.* 36, 4, Article 63 (July 2017), 11 pages.

DOI: <http://dx.doi.org/10.1145/3072959.3073682>

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.

© 2017 ACM. 0730-0301/2017/7-ART63 \$15.00
DOI: <http://dx.doi.org/10.1145/3072959.3073682>

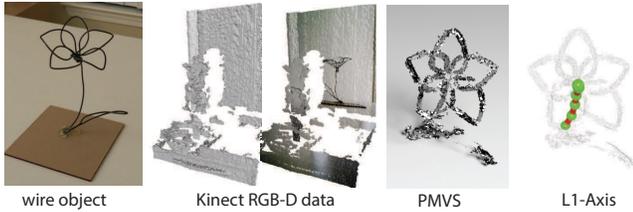


Fig. 2. While commodity depth sensors, such as the *Microsoft Kinect*, fail to provide reliable depth data for wire objects (notice how most of the flower is perceived to be planar), PMVS (Furukawa and Ponce 2010), a state-of-the-art multi-view stereo (MVS) method, generates a noisy and partial point cloud. Curve skeleton extraction methods (L1 medial axis (Huang et al. 2013)) are limited when applied to such MVS output.

1 INTRODUCTION

Wire art is an ancient art form that refers to creation of complex shapes by bending and connecting wires. First introduced by the Egyptians around 3000 BC (Ogden 1991), wire art is today practiced both by expert designers and hobbyists and is common in furniture design, metal sculptures, jewellery making, etc. Unlike other surface-based objects, wire-based compositions are fundamentally different as they consist entirely of 1D elements with ‘surfaces’ being perceptually filled in. This renders such objects more intriguing and creates a desire to digitize them to enrich virtual worlds.

Digitizing wiry objects remains a challenging problem, even though a multitude of depth and image based reconstruction methods exist. Such objects are composed of multiple thin wire structures that occlude each other and are thus difficult to scan even with high-end scanners. Moreover, the resolution of commodity depth sensors, *e.g.*, the *Microsoft Kinect*, is insufficient to produce reliable depth data for thin features (see Figure 2). Thus, we resort to image-based methods which provide higher resolution at a low-cost and a flexible acquisition process. While significant advances have been made in the domain of image based methods, there are two fundamental problems in reconstructing wiry objects that we address.

Correspondence problem. The success of any image based reconstruction method depends on detecting reliable correspondences, *i.e.*, identifying points across images that correspond to the same 3D point in the physical world. This is especially challenging in the case of wiry objects due to the lack of distinctive image features along thin wires of uniform color. As a result, traditional multi-view stereo (MVS) approaches suffer from significant noise and missing data (see Figure 2). Moreover, the resulting reconstructions have only a low level representation, *i.e.*, point clouds, without a global curve topology required for a faithful representation. Although there exist methods to extract curve skeletons from an input point cloud (Huang et al. 2013), the amount of noise in MVS output renders such methods impractical as shown in Figure 2.

We side-step from the issues of point-level correspondences by utilizing high order curve primitives for image matching. In the absence of distinctive image features, we rely on epipolar cues to find matches among 2D curve segments detected in each input view. For each 2D curve segment detected in a reference view, there are potentially multiple 2D curve segments in another view that

satisfy the epipolar constraints (Hartley and Zisserman 2004). The number of conforming candidate matches is especially high for wiry objects, which lack surface patches to hide the wires that are in the background. In other words, curves at various depths are simultaneously visible in each image and so result in prohibitively many candidate matches (in our experiments, for each 2D curve segment, we typically get 3-4 candidate matches). While it is possible to locally choose one of these candidate matches based on a *data* cost, *e.g.*, by measuring the reprojection error of each selected 3D curve segment to the input images, we observe that this is not sufficient to resolve all the ambiguities (see Figure 7). Instead we propose a strategy that explores an additional *smoothness* cost that favors consecutive 3D curve segments with smooth angles and small gaps. We formulate a global selection problem to choose the desired 3D curve segments from a set of candidate 3D curve segments by balancing between the data and smoothness costs.

Wire decomposition. Once we construct a set of 3D curve segments, the next critical step is to recover the global topology of these segments and obtain a decomposition of the input object into a set of continuous wires. We observe that objects made of many pieces of wires are often not stable and difficult to assemble, especially for hobbyists. On the other hand, bending only a single piece of wire to create sharp edges is also hard. In light of these observations, we define a graph where each (estimated) 3D curve segment is represented by a vertex and vertices are linked by edges that are assigned costs based on the angle and the distance between the curves they connect. We then formulate an instance of the *multiple traveling salesman problem* over this graph to find low-cost path(s) connecting all the graph vertices, where each vertex is covered by exactly one path. By associating a cost to each introduced new path, we balance between smoothness (*i.e.*, low cost paths) and simplicity (*i.e.*, as few paths as possible).

Having discovered the global curve topology, we perform a final refinement step to fit a cubic B-spline to each 3D wire. We optimize for the parameters of the splines so that consecutive curve segments along a path are smoothly connected and the reprojection error to the input images is minimized. This refinement step helps to fill small gaps along paths caused by insufficient image information and results in high fidelity reconstructions as shown in Figure 1. Such reconstructions enable easy manipulation or recreation using wire bending machines (Miguel et al. 2016).

We evaluate our method on a variety of real and synthetic examples composed of both single and multiple wires and provide comparisons with general purpose reconstruction methods and demonstrate superior results (see Figure 8 and Section 6). Our main contribution is a practical and effective method that reconstructs wiry objects from a very few input images (3 in all the presented examples). To our knowledge, no previous method can reconstruct wire sculptures at the complexity of our examples. Our method makes this possible by using high order curve primitives and global (*i.e.*, smoothness and simplicity) as well as local data cues.

2 RELATED WORK

Our work builds up on previous methods related to curve-based structure-from-motion and multi-view stereo as well as modeling of delicate structures from images and point clouds.

2.1 Modeling of delicate structures.

With the many advances in acquisition technologies (*e.g.*, structured light, LiDAR, and more recently commodity depth sensors), we have seen substantial progress in the area of surface reconstruction (Berger et al. 2014) to extract high fidelity surfaces from input data. An equally important problem is to improve the acquisition experience in terms of data coverage (Yan et al. 2014), quality (Wu et al. 2014), and speed (Fan et al. 2016). These methods, however, focus on objects with smooth surfaces while scanning thin wires remains challenging due to the limited sensor resolution. In the specific case of objects with delicate structures, several automatic and interactive modeling methods have been presented. Li et al. (2010) introduce the deformable model *arterial snakes* to reconstruct such objects from high-quality dense 3D scans. An alternative approach is to fit generalized cylinders to image data (Chen et al. 2013) or point clouds (Yin et al. 2014) where the fitting process is guided by a skeleton either defined manually as in 3-Sweep (Chen et al. 2013) or extracted automatically (Huang et al. 2013). However, the fact that wire objects do not have sufficient thickness makes it extremely challenging for a user to define a section plane or a non-planar extrusion curve in a 2D image manually. Furthermore, 3D point data (*e.g.*, obtained by MVS algorithms) is not reliable enough to automatically detect curve skeletons as we show in our evaluations. Martin et al. (2014) present a method to reconstruct thin tubular structures from a dense set of images using physics-based simulation of rods to improve accuracy. They use a 3D occupancy grid to disambiguate 2D crossings of cables. However, unlike the clean distribution of occupied cells obtained for cable structures, for typical wiry objects such a grid is very noisy: due to lack of surface occlusion, wires from the front and back of an object are simultaneously visible in each view and this results in ambiguities. Our method recovers the topology of the wires in this challenging case by using a global optimization.

More recently, some approaches to designing wire objects have been proposed with a specific focus on fabricating these designs. Iarussi et al. (2015) tackle the problem of extracting the topology of a wire jewelry from a single drawing. However, their method assumes that the curves are planar and fully visible in the single image input. Miguel et al. (2016) present an interactive system to convert a closed surface mesh to a self-supporting wire sculpture. While both approaches explore similar characteristics of wire objects, *e.g.*, simplicity, they fundamentally focus on a different problem of bringing digital designs into life. Thus, our approach is complementary and can provide input to these methods.

2.2 Curve-based Structure-from-Motion.

In recent years, a multitude of successful structure-from-motion (SfM) (Snavely et al. 2006; Wu et al. 2011) and multi-view stereo (MVS) (Furukawa and Ponce 2010; Goesele et al. 2007) algorithms have been developed. These methods work on the general principle of identifying point correspondences across images that represent the same world-space *surface* locations. While these methods produce impressive results for objects with reliable texture information, they perform poorly when there are insufficient discriminative point features. Some researchers have tackled this limitation by exploring

the presence of higher order features such as lines (Baillard et al. 1999; Hofer et al. 2014, 2013; Jain et al. 2010) and curves (Xiao and Li 2005). We refer the reader to the recent work of Fabbri et al. (2016) for a theoretical foundation of the multi-view geometry properties of curves.

Curve-based reconstruction methods can be grouped according to the simplifying assumptions they make: Berthilsson et al. (2001) assume that each 3D curve is *fully visible* in each view with no self-occlusions; Hong et al. (2004) focus on reflective symmetric curves that are fully visible in one or two images; assuming access to the range of depth values in the reconstructed scene, Kahl and August (2003) assign depth values to 2D curves, while Teney and Piater (2012) design a probability distribution to model the space of plausible curve reconstructions. Rao et al. (2012) explore known correspondences between endpoints of planar curves to develop a curve-based SLAM (simultaneous localization and mapping) system. Similarly, Nurutdinova et al. (2015) assume that image curves are already in correspondence and optimize for both camera calibration and parameters of the reconstructed space curves.

In the absence of prior knowledge, a typical workflow, which we also adopt, is to first detect 2D curve segments in each image and then utilize epipolar constraints to generate hypotheses of corresponding curves across image pairs. This process typically results in multiple ambiguous matches and an additional view is required for verification (Robert and Faugeras 1991). Recent work (Fabbri and Kimia 2010; Usumezbas et al. 2016) uses this workflow to reconstruct curvilinear structures in general scenes requiring a dense set of images. Our work, in contrast, reconstructs wireframe objects from as few as three input views.

All the above mentioned previous works obtain a reconstruction in the form of individual curve segments, which possibly suffers from noisy measurements and gaps. Instead, we globally extract continuous curve paths to produce more faithful representations of wire objects. A similar path finding approach has been proposed in the context of medical imaging (Delmas et al. 2015); however, no previous approach has demonstrated results at the complexity of the examples we show.

3 OVERVIEW

The input to our algorithm is a set of n (in our experiments $n = 3$) images, $\mathcal{I} = \{I^1, \dots, I^n\}$, of a 3D wire model captured from different viewpoints along with their camera parameters, $\mathcal{K} = \{\mathcal{K}^1, \dots, \mathcal{K}^n\}$. The output of our method is a set of k reconstructed 3D wires $\mathcal{W} = \{W_1, \dots, W_k\}$, where each wire is a continuous path represented as a B-spline curve. We provide an overview of this pipeline in Figure 3.

In a pre-processing stage, we extract a set of 2D curve segments in each image. Specifically, we first convert the images to the YCbCr color space with histogram equalization on Y-channel (luminance component) to increase the contrast of the wire regions relative to the image background. Then, starting from an arbitrary pixel in the perceived wire region, we utilize a flooding method to extract all the connected pixels that have a similar color and apply a thinning algorithm to extract a one-pixel wide curve. Finally, we decompose each such curve into segments such that no curve segment contains

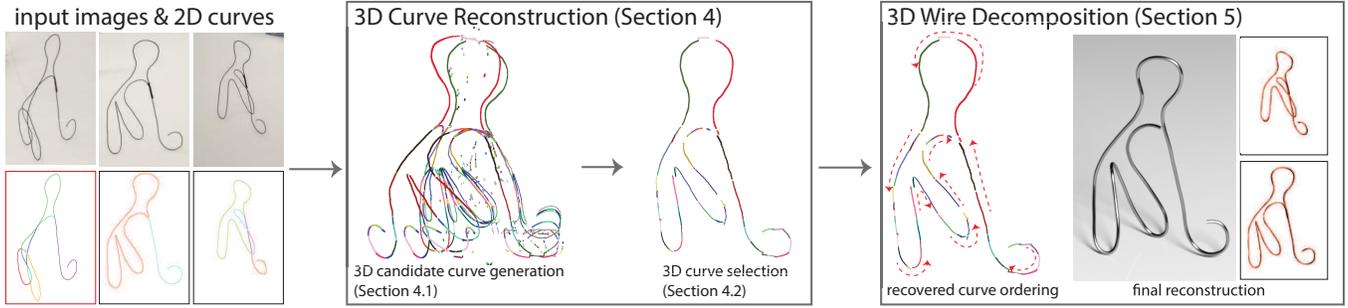
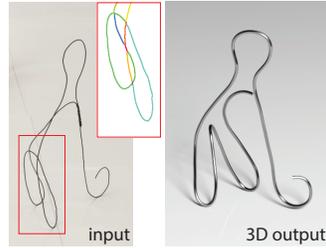


Fig. 3. Given a set of input images of a wire object, our method first detects 2D curve segments in each image. We then generate a set of candidate 3D curve segments corresponding to each 2D image curve (Section 4.1) and solve a global selection problem to choose one 3D curve from each candidate set (Section 4.2). Finally, we recover the global topology of the 3D curve segments (as denoted by the dashed red arrows) and decompose the object into a set of continuous wires (Section 5). We show this final reconstruction both in 3D and also how its projection (in red) aligns with the original image curves (in black).

any joint or branching point (Kovesi 2017). As a result, for each image I^j , we obtain a set of 2D curve segments denoted as $c^j = \{c_i^j\}$.

Given a set of 2D curves extracted from each image, our first goal is to generate a corresponding set of candidate 3D curve segments. To accomplish this goal, we establish continuous correspondences between matching 2D curve segments from different images. We use a conservative

strategy when decomposing 2D curves in each image into segments free of branching points to ensure that no 2D curve segment contains projections of disjoint 3D curves. The red highlighted region in the inset figure shows an example, where each 2D curve segment is denoted with a different color. Some of the branching points detected in the images may have been created due to self-occlusions and separate 2D curve segments that are projections of a single continuous 3D curve. While this conservative decomposition strategy results in the loss of such continuity relations, they are recovered in the subsequent stages of our algorithm. For example, the final 3D result shown in the inset consists of a single continuous wire.



For each 2D curve segment c_i^r detected in a reference view I^r , we generate a set of candidate 3D curve segments by establishing correspondences with 2D curve segments in a neighboring view that satisfy the epipolar constraints (Section 4.1). Once such sets of candidate 3D curve segments are generated for all views, we formulate a global selection problem that solves a quadratic optimization problem to choose one 3D curve segment from each candidate set (Section 4.2). This optimization not only minimizes the re-projection error of a selected 3D curve segment with respect to each view but also favors consecutive 3D curve segments that form smooth angles. As a result, we select a set of 3D curve segments, denoted as C_{opt} .

Next, to extract a global curve topology among the individual and disjoint 3D curve segments in C_{opt} , we solve a multiple traveling salesman problem (mTSP) to identify continuous and smooth paths such that each 3D curve segment is contained in exactly one

path (Section 5). Each such path corresponds to a continuous wire composed of a sequence of 3D curve segments which are a subset of C_{opt} . Finally, for a smooth and compact representation, we represent each wire as a smooth cubic B-spline curve and optimize for its parameters using a fitting procedure to minimize its projection error with respect to the input images.

Next, we describe the respective steps in detail.

4 3D CURVE RECONSTRUCTION

The 3D curve reconstruction step of our method first generates a set of candidate 3D curve segments and then solves a global selection optimization to identify the correct 3D curves among these candidates. Without loss of generality, we first explain this process for a given pair of reference and neighboring views (I^r, I^n) and then describe how it can be easily extended to multiple views.

4.1 3D Candidate Curve Generation

Given a set of curve segments $c^r = \{c_i^r\}$ detected in a reference view I^r , we represent each c_i^r as a sequence of densely sampled image points, $p_{i,j}^r, j = 1, 2, \dots, s$. For each c_i^r , starting from the first sample point $p_{i,1}^r$, we compute its corresponding epipolar line, $l_{i,1}^{r \rightarrow n}$, in a

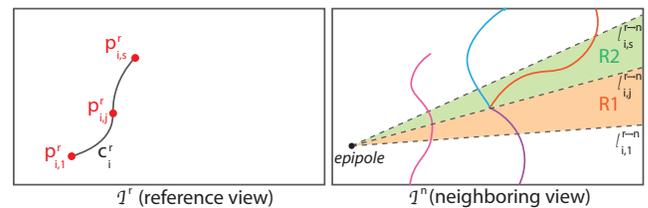


Fig. 4. We show a 2D curve segment c_i^r in I^r along with sample points ($p_{i,1}^r, p_{i,j}^r, p_{i,s}^r$) and their corresponding epipolar lines ($l_{i,1}^{r \rightarrow n}, l_{i,j}^{r \rightarrow n}, l_{i,s}^{r \rightarrow n}$) in I^n . Epipolar lines corresponding to sample points between $p_{i,1}^r$ and $p_{i,j}^r$ result in two intersection points in I^n (R1) and generate two candidate matching 2D curve segments (pink and purple). Starting with $p_{i,j}^r$, epipolar lines result in three intersection points (R2). Thus we break c_i^r into two segments at $p_{i,j}^r$ such that for the segment between $p_{i,j}^r$ and $p_{i,s}^r$ three candidate matching 2D curve segments (pink, blue, and orange) are generated.

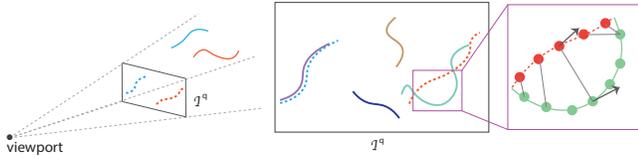


Fig. 5. Given a candidate 3D curve segment, we project it to a third view I^q and find the 2D curve segment in I^q that has the smallest aggregated closest point distance with its projection. We compute a *confidence cost* for the 3D curve segment which measures both the distance and the angular deviation between such closest points. 3D curve segments (in orange) which receive a high confidence cost can be pruned, while others (in blue) are kept as a candidate.

neighboring view I^n . Each epipolar line $l_{i,1}^{r \rightarrow n}$ is likely to intersect multiple disjoint 2D curve segments in I^n , each being a possible match for c_i^r (see Figure 4)¹. We expect the epipolar lines corresponding to consecutive sample points $p_{i,j}^r$ to result in intersections with the same set of candidate matching curve segments due to the continuity of image curves. Thus, we trace out these intersections as long as the number of intersections for consecutive sample points stays constant. If we reach a sample point $p_{i,j}^r$ on c_i^r whose epipolar line intersects a different number of 2D curve segments in I^n (e.g., due to a discontinuity resulting from self-occlusions), we break c_i^r into two segments consisting of sample points $\{p_{i,1}^r, p_{i,2}^r, \dots, p_{i,j-1}^r\}$ and $\{p_{i,j}^r, p_{i,j+1}^r, \dots, p_{i,s}^r\}$, respectively. By repeatedly tracing the intersection points of the consecutive epipolar lines and splitting c_i^r as necessary, we obtain a resulting set of 2D curve segments c^r in I^r , where for each c_i^r we have a set of matching candidate 2D curve segments $c_{i,j}^{r \rightarrow n}$, $j = 1, 2, \dots, m$ in the neighboring view I^n .

Given a curve segment c_i^r in I^r and its candidate matching curve segments $c_{i,j}^{r \rightarrow n}$ in I^n , $j = 1, 2, \dots, m$, we generate a set of 3D curve segments $C_{i,j}^r$, $j = 1, 2, \dots, m$, each represented by a sequence of 3D points reconstructed by triangulating the corresponding sample points in I^r and I^n . Each $C_{i,j}^r$ is potentially the correct 3D curve segment for $c_i^r \in I^r$ and is also compatible with I^n by construction. We use a third view I^q to help assess the likelihood of each of $C_{i,j}^r$ to be the correct 3D curve segment corresponding to c_i^r (see Figure 5).

In particular, we project each $C_{i,j}^r$ to I^q and find the 2D curve segment in I^q that is closest to its projection. We compute this closest 2D curve segment by establishing closest point correspondences between each sample on the projection of $C_{i,j}^r$ and the sample points on each $c_o^q \in I^q$ and choosing the c_o^q that results in the smallest aggregated distance between such correspondences. Given a set of point correspondences (u, v) , where u lies along the projection of the 3D curve segment $C_{i,j}^r$ to I^q and v is its closest point correspondence on the 2D curve segment $c_o^q \in I^q$, we compute a *confidence*

cost, $\mathcal{S}(C_{i,j}^r)$, as:

$$\mathcal{S}(C_{i,j}^r) = \frac{\sum_{(u,v)} \left(\left(\|u - v\| / \text{diag}(I^q) \right) + \eta(1 - |t_u \cdot t_v|) \right)}{s}, \quad (1)$$

where we measure both the distances between the corresponding points and the deviation between their tangential directions, (t_u, t_v) (see Figure 5). η denotes the relative weighting between the two measures and is set in the range (0.001 – 0.004) in our experiments. We normalize the distance measures by the length of the diagonal of the image, $\text{diag}(I^q)$, and normalize the aggregated distance and tangential deviation by s , the number of sample points on $C_{i,j}^r$. This confidence cost is used in the subsequent stage of our method to identify the correct 3D curve segments.

4.2 3D Curve Selection

In the previous step, for each 2D curve segment c_i^r , we reconstruct a set of 3D curve segments, $C_i^r = \{C_{i,j}^r\}$, composed of multiple candidates satisfying the epipolar constraints (see Figure 6) (in our experiments we often have 3-4 candidates for each curve segment). Next, we explore two type of cues to choose one 3D curve segment for each c_i^r as its true 3D proxy. First, for each c_i^r , we want to select the 3D curve segment that matches image observations well. However, due to many self-occlusions, image observations alone are not sufficient to resolve all the ambiguities. Therefore, we also explore pairwise relations between 3D curve segments. Since the true 3D curve segments are assumed to be taken from an object created by bending continuous wires, consecutive curve segments along this wire are smoothly connected. Thus, pairs of nearby 3D curve segments with smooth angles are more likely to exist together, which provides an effective pairwise cue. We formalize these cues in an optimization problem as follows. First, for ease of notation, we denote $\mathcal{A}^r = \cup_i C_i^r$, the union of all candidate 3D curve segment sets C_i^r . For each 3D curve segment in $\mathcal{A}_j \in \mathcal{A}^r$, we define a binary indicator variable x_j where $x_j = 1$ if the corresponding curve segment is selected and $x_j = 0$ otherwise. We denote by \mathbf{X} the vector formed by concatenating all such binary variables. Since, our goal

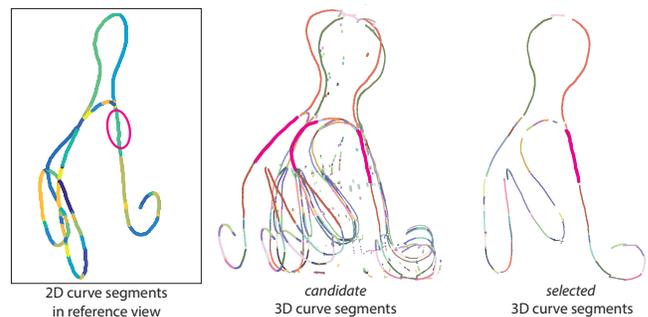


Fig. 6. After the 3D candidate curve generation step, for each 2D curve segment we obtain multiple candidates (e.g., for the highlighted 2D curve segment, we obtain 3 candidate curves in 3D (in pink)). The 3D curve selection process identifies the correct 3D curve among such candidates solving a global selection problem.

¹We note that there may be degenerate cases where an epipolar line is locally almost parallel to a 2D curve segment making it difficult to compute robust intersections. We refer the reader to the Appendix for a discussion of how we handle these cases.

is to select only one 3D curve segment from each candidate set, we define the linear constraint, $\sum_{\mathcal{A}_j \in \mathcal{C}_i^r} X(j) = 1$ for each \mathcal{C}_i^r .

We evaluate the compatibility of each 3D curve segment using the *confidence cost* defined in Equation 1. We concatenate the confidence scores of all the 3D curve segments $\mathcal{A}_j \in \mathcal{A}^r$ in a vector \mathbf{U} such that $\mathbf{U}_j = \mathcal{S}(\mathcal{A}_j)$.

Next, in order to evaluate the pairwise relations between 3D curve segments \mathcal{A}_i and \mathcal{A}_j , we define a pairwise cost $d(\mathcal{A}_i, \mathcal{A}_j)$:

$$d(\mathcal{A}_i, \mathcal{A}_j) = e(\mathcal{A}_i, \mathcal{A}_j) + \mu f(\mathcal{A}_i, \mathcal{A}_j). \quad (2)$$

$e(\mathcal{A}_i, \mathcal{A}_j)$ measures the distance in 3D between the closest endpoints of \mathcal{A}_i and \mathcal{A}_j and $f(\mathcal{A}_i, \mathcal{A}_j) = (1 - \cos \alpha)/2$, where α is the angle between the tangent directions of these closest endpoints. Here μ denotes the relative importance between the distance and angle terms ($\mu = 1$ in our experiments). We aggregate all pairwise relations in a square matrix \mathbf{V} of dimension $b \times b$ where b is the cardinality of \mathcal{A}^r (i.e., the total number of candidate 3D curve segments). We set $V_{ij} = d(\mathcal{A}_i, \mathcal{A}_j)$ for all non-diagonal entries and set the diagonal entries to zero.

Given \mathbf{U} and \mathbf{V} , we optimize for \mathbf{X}^* that minimizes the following:

$$\begin{aligned} \mathbf{X}^* := \underset{\mathbf{X}}{\operatorname{argmin}} \quad & \mathbf{U}^T \mathbf{X} + \lambda \mathbf{X}^T \mathbf{V} \mathbf{X}, \\ \text{subject to } \forall \mathcal{C}_i^r : \quad & \sum_{\mathcal{A}_j \in \mathcal{C}_i^r} X(j) = 1 \end{aligned} \quad (3)$$

where λ determines the relative weighting of the unary and binary cues and is set to 0.1-0.2 in our experiments (based on thickness and size of the object). We solve Equation 3 using the quadratic programming package in Matlab (The MathWorks 2016) and obtain an optimized set of 3D curve segments \mathcal{C}_{opt} which contains exactly one 3D curve segment as the true 3D proxy for each 2D curve segment c_i^r .

While it is possible to solve Equation 3 over all of the candidate 3D curve segments, in cases where this becomes computationally inefficient (due to a very large number of unknowns), it is possible to employ a pruning strategy where 3D curve segments with a confidence cost greater than a threshold (0.0015-0.008 in our experiments) are pre-filtered. Finally, there may be isolated 3D curve segments that have small confidence costs but are far away from the rest of the curve segments. Such isolated curves are typically

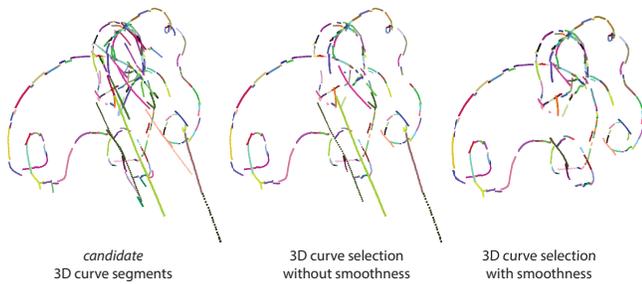


Fig. 7. For a given set of candidate 3D curve segments, using only the unary data term results in the selection of many wrong 3D curve segments. This shows the necessity of a global optimization with additional smoothness priors.

outliers and can be removed by thresholding its distance to its nearest 3D curve segment. Finally, in order to show the importance of utilizing both unary and binary cues, for a given set of candidate 3D curve segments, we solve Equation 3 twice, with and without the binary term. As shown in Figure 7, using only unary terms results in selection of many wrong candidates. This proves the effectiveness of our global selection strategy.

4.3 Taking Alternative Reference Views

We have described the 3D curve segment reconstruction step that generates 3D candidate curves from a pair of reference and neighbor views and uses a third view for verification. When there are more views, we utilize an incremental reconstruction strategy. Starting from a pair of reference and neighbor views, we reconstruct a set of 3D curves for each 2D curve segment in the reference view as just described. We then choose a new view as reference view and find a compatible neighboring view (a view with a good range of parallax as proposed by Gesele et al. (2007)). We project all the 3D curve segments reconstructed so far to the reference view and mark all (parts of) 2D curve segments that are covered by a projection as *processed*. A 2D curve segment is considered to be covered by a projection of a 3D curve segment if the aggregated distance and tangent deviation between the closest points among the 2D segment and the projection is below a certain threshold (0.02-0.05% compared to bounding box diagonal). We perform the same reconstruction step for all the *unprocessed* 2D curve segments in the new reference view. We repeat this step until all views have been considered as a reference view.

5 3D WIRE DECOMPOSITION

Once we have constructed a set of 3D curve segments, \mathcal{C}_{opt} , as explained in the previous section, our next goal is to recover the global topology of these curves and decompose them into a set of distinct continuous wires. To perform this decomposition, we exploit two unique features of wire objects we consider as input. First, such objects are often composed of only a small number of wires because it is not trivial to stably connect many pieces of wires. Second, it is hard to bend wires to create sharp angles due to physical resistance. Therefore, we expect each output wire to be as smooth as possible. In light of these considerations, we formulate the 3D wire decomposition as a multiple traveling salesman problem (mTSP) as described in the following.

Given the set of 3D curve segments, $\mathcal{C}_{\text{opt}} = \{C_i\}$, $i = 1, 2, \dots, b$, we construct a directed graph $G = (\mathcal{V}, \mathcal{E})$, where each C_i is represented as a vertex $V_i \in \mathcal{V}$. We define an additional *start node*, V_0 , such that each path defined over this graph starts and ends at this node. A directed edge $e_{ij} \in \mathcal{E}$ going from the node V_i to V_j is assigned a cost w_{ij} based on the pairwise relation between the corresponding 3D curve segments C_i and C_j . Precisely, $w_{ij} = d(C_i, C_j)$ as given in Equation 2. The edge costs are defined symmetrically, i.e., $w_{ij} = w_{ji}$.

Given $G = (\mathcal{V}, \mathcal{E})$, our goal is to find k distinct paths that start and end at V_0 such that each V_i ($i > 0$) is contained in exactly one path. Each of these paths corresponds to a continuous 3D wire used to create the input wire model. Since each path is uniquely defined by the edges it is composed of, each edge $e_{ij} \in \mathcal{E}$ is assigned a binary variable x_{ij} : $x_{ij} = 1$ if e_{ij} is contained in a path and $x_{ij} = 0$

otherwise. Furthermore, each node $V_i \in \mathcal{V}$ is assigned an auxiliary integer variable $u_i \in \mathbb{N}$ to denote the order in which V_i is visited along a path, $u_0 = 0$ for the dummy start node V_0 . We formulate our path finding problem as minimizing the total cost of the selected edges and the number of paths:

$$\{k^*, x_{ij}^*\} := \operatorname{argmin}_{\{k, x_{ij}\}} \sum_{i,j} x_{ij} w_{ij} + \xi k. \quad (4)$$

ξ is a fixed cost associated with using a unique path. In order to ensure each path is valid (*i.e.*, each vertex is visited by exactly one path and no path is composed of disconnected cycles), Equation 4 is solved subject to the following constraints.

To ensure each node V_i ($i > 0$) is visited by a path, exactly one of the incoming and outgoing edges of a node needs to be selected in the solution:

$$\begin{aligned} \forall i > 0 : \sum_{j, j \neq i} x_{ji} &= 1, \\ \forall i > 0 : \sum_{j, j \neq i} x_{ij} &= 1. \end{aligned} \quad (5)$$

Each of the k paths is required to start and end at the start node V_0 :

$$\begin{aligned} \sum_{j, j > 0} x_{j0} &= k, \\ \sum_{j, j > 0} x_{0j} &= k. \end{aligned} \quad (6)$$

In order to avoid paths composed of disconnected cycles, *subtour elimination constraints* as proposed by Kara and Bektas (2006) are defined:

$$\begin{aligned} \forall i > 0 : \\ u_i + (b - 1 - k)x_{0i} - x_{i0} &\leq b - k, \\ u_i + x_{0i} &\geq 2, \\ \forall i, j > 0, i \neq j : \\ u_i - u_j + (b - k + 1)x_{ij} + (b - 1 - k)x_{ji} &\leq b - k. \end{aligned} \quad (7)$$

We use the Gurobi optimization package (Gurobi Optimization 2016) to solve the binary-integer optimization problem given in Equation 4 with respect to the constraints defined in Equations 5, 6, and 7. As a result, we obtain a set of k continuous wires, $\mathcal{W} = \{\mathcal{W}_1, \dots, \mathcal{W}_k\}$, where each \mathcal{W}_i is composed of an ordered set of 3D curve segments.

We note that ξ , the cost of using a unique wire, provides a trade-off in the number of wires included in a decomposition. Increasing this cost prefers decompositions composed of fewer wires. In our experiments, we found setting $\xi = 1/10^{th}$ of the maximum edge cost provides a good balance. We also experimented with optimizing Equation 4 with known number of wires, *i.e.* manually setting k . In the special case of $k = 1$ (*i.e.*, for objects composed of a single wire), this formulation reduces to the standard travelling salesman problem where the goal is to find a single path that visits each node. Finally, while Equation 4 can be optimized over a fully connected graph G , some edges with very high costs can be pre-pruned for better computational efficiency.

5.1 Image-guided Curve Smoothing

At this stage, we have fully recovered all the 3D proxy information we need to represent the input object. Due to instabilities in lifting very small 2D curve segments to 3D, there may be small gaps between the 3D curve segments along a given wire decomposition. Thus, in a final stage, we represent each wire \mathcal{W}_i with a cubic B-spline and optimize for its parameters so that the projection of the resulting 3D wire matches the image observations. In particular given a wire \mathcal{W}_i , we represent it as:

$$\mathcal{W}_i = \mathcal{B}(t) = \sum_{j=1}^g \mathcal{X}_j \phi_j(t), \quad (8)$$

where \mathcal{X}_j are the control points and ϕ_j are fixed cubic B-spline basis functions. If the total number of points sampled along all the 3D curve segments contained in \mathcal{W}_i is h , we define $g = h/10$ control points. We first fit a B-spline to the set of all sample points along \mathcal{W}_i with each sample point u_a , $a = 1, 2, \dots, h$ associated with a fixed parameter value along this spline, $\mathcal{B}(t_a)$, $0 \leq t_a \leq 1$.

Given this initial B-spline fitting curve in 3D, we next perform an iterative optimization to obtain a refined set of control points \mathcal{X}_j^* that minimizes the following energy:

$$\begin{aligned} \mathcal{X}_j^* := \operatorname{argmin}_{\mathcal{X}_j} \sum_{a=1}^h \sum_{o=1}^n \left\| \mathcal{K}^o(\mathcal{B}(t_a)) - p^o \right\|^2 + \\ \omega \sum_{a=2}^{h-1} \left\| \mathcal{B}(t_{a-1}) - 2\mathcal{B}(t_a) + \mathcal{B}(t_{a+1}) \right\|^2. \end{aligned} \quad (9)$$

The first term minimizes the distance between $\mathcal{K}^o(\mathcal{B}(t_a))$, the projection of a 3D sample point u_a to image I^o , and its closest point correspondence $p^o \in I^o$. We project each sample point to the images it is visible in. The second term imposes a Laplacian smoothness penalty between consecutive sample points and ω (set to 10^3 in our experiments) defines the relative importance between the data and smoothness terms.

We solve Equation 9 in an iterative manner, re-establishing the correspondences between the projections of 3D sample points and 2D sample points in the images at each iteration. Typically, this process converges in 5-15 iterations and the resulting B-spline representation of each wire is free of gaps and provides an accurate, smooth, and compact reconstruction of the object.

6 RESULTS

We evaluate our method on a wide set of examples with varying complexity. For each of these examples we utilize 3 images of the wire object taken from different viewpoints as input, together with their camera parameters. We use the open-source structure-from-motion (SfM) tool, VisualSfM (Wu 2011; Wu et al. 2011), to obtain the camera parameters in a pre-calibration step. Note that we make use of nearby textured objects to ensure that a sufficient number of image correspondences are detected for reliable SfM computation.

For each of our examples, we provide a sample input image and a rendering of our final reconstruction in Figure 8 and refer to the accompanying video and supplementary material for more closeups. Our examples have varying complexities, each being composed of 1-3 wires. For each of these examples, our method

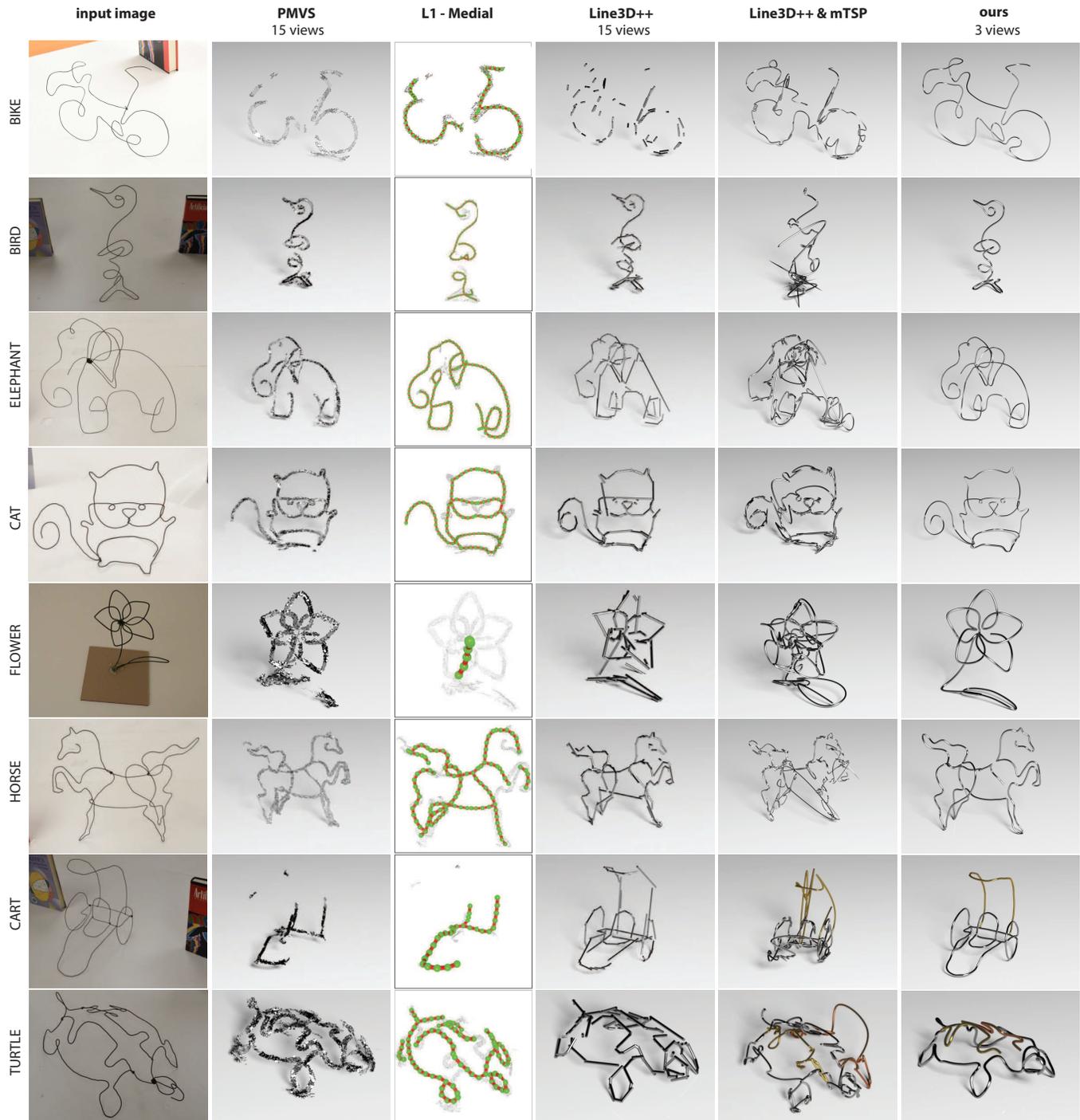
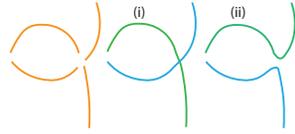


Fig. 8. We evaluate our method on several objects of varying complexity. For each object, we provide one of the input images and our final 3D reconstruction. We also provide comparisons with PMVS (Furukawa and Ponce 2010) and Line3D++ (Hofer et al. 2016) run on a set of input images consisting of 15 views. Finally, we evaluate the L1-medial skeleton extraction method of Huang et al. (2013) on the point clouds generated by PMVS. We also run the mTSP formulation on the output of Line3D++ output to demonstrate how small variations can lead to large topological errors with significant errors in the final output.

provides a faithful 3D geometry and a plausible wire decomposition. We note that during wire decomposition a specific challenging case is when four curve segments come together (see inset). In this case, there are two possible options for connecting these curves, i.e. (i) a crossing, or (ii) or a continuous path with sharp bends, both resulting in plausible decompositions. Our method prefers smooth crossings since we utilize the smoothness prior. If this is not the desired decomposition, the user may want to reclassify junction points.



6.1 Quantitative Evaluations

In order to quantitatively evaluate our algorithm, we reconstruct a synthetic wire object (similar in complexity to the CAT example) from its renderings using ground truth camera parameters. Furthermore, we generate reconstructions by adding noise to the camera parameters. We represent the rotation of the camera with Euler angles and add random noise to each of the angles sampled from a normal distribution with mean zero and standard deviation of one degree. For translation, we add random noise by sampling from a normal distribution of mean zero and a standard deviation of 0.1%, 0.2%, and 0.3% of the length of the diagonal of the bounding box of the synthetic model in 3D. In each case, we measure the closest point distances between the ground truth and the reconstructed model in 3D. We report the average and maximum of such closest point distances in Table 1. We note that we report the errors as a percentage of the length of the diagonal of the bounding box of synthetic model in 3D.

We also quantitatively evaluate the performance of our approach on real examples. We compute a re-projection error by sampling points on the reconstructed 3D wires, projecting these samples to the input views, and measuring the distance between the projected points and the closest points sampled on the 2D curves for each view. We report the average and maximum of such re-projection errors for each input view in Table 2. We note that we report the errors as a percentage of the length of the diagonal of the 2D bounding box of the wiry object in the corresponding view.

6.2 Performance

We measure the execution time of different stages of our algorithm on a machine with Intel i7 3.46 GHz CPU with 24 GB RAM. While 3D curve reconstruction (Section 4) takes about 10 to 22 seconds;

Table 1. We reconstruct a synthetic model from its rendering both using ground truth camera parameters (no noise) and by adding random noise to the camera parameters. We report the error as the average and maximum of the closest point distances between the ground truth and the reconstructed model in 3D. Distances are reported as a percentage of the length of the diagonal of the bounding box of synthetic model in 3D.

	no noise	0.1% noise	0.2% noise	0.3% noise
avg	0.17	0.35	0.54	0.6
max	0.98	1.02	2.06	2.89

Table 2. For real examples, we compute a re-projection error by sampling points on the reconstructed 3D wires, projecting these samples to each input view, and measuring the distance between the projected points and the closest points sampled on the 2D curves in the view. We report the average and maximum of such re-projection errors for each input view as a percentage of the length of the diagonal of the 2D bounding box of the wiry object in the corresponding view.

	view 1		view 2		view 3	
	avg	max	avg	max	avg	max
HUMAN	0.29	1.70	0.37	0.92	0.37	1.85
BIKE	0.19	0.94	0.19	1.36	0.22	1.65
BIRD	0.13	0.58	0.14	0.88	0.14	0.78
ELEPHANT	0.18	0.85	0.21	2.47	0.17	1.14
CAT	0.26	1.28	0.40	3.66	0.24	1.50
FLOWER	0.51	3.99	0.47	3.43	0.41	3.53
HORSE	0.24	1.11	0.24	2.01	0.21	1.56
CART	0.16	0.59	0.18	1.05	0.21	1.78
TURTLE	0.26	2.45	0.57	3.18	0.29	1.86

3D curve decomposition (Section 5) takes about 50 to 60 seconds for simple models (e.g., HUMAN) and a few minutes for complex models (e.g., CART). The complexity of a wire sculpture is not directly based on the number of wires it is composed of, instead it depends on the density of the wires since denser configurations result in large number of candidate 3D curves.

6.3 Comparisons

We compare our approach to *PMVS* (Furukawa and Ponce 2010), a state-of-the-art multi-view stereo (MVS) algorithm, and *Line3D++*, a recent line-based MVS algorithm (Hofer et al. 2016) developed for scene abstraction. We run each of these methods on an input image set composed of 15 views including the original 3 views we use for our method where the calibration is obtained by VisualSFM. (We run the same comparison on an input image set of 10 views and provide the results in the supplementary material.) As illustrated in Figure 8, even when using substantially more views, both approaches still suffer from noise and significant amount of missing data. Furthermore, neither point- nor individual line-based representations can capture the global topology of the wires and thus are not suitable for any post-processing operations. We evaluate the recent L1-medial skeleton extraction method (Huang et al. 2013) on the point clouds generated by PMVS (we include the parameters we use for this method in the supplementary material). As shown in



Fig. 9. Our method is not designed to handle very dense wire sculptures that are closer to surfaces instead of 1D elements. ©Pinterest

Figure 8, due to noise and missing data in the PMVS output, this approach fails to recover a complete and accurate curve skeleton in many cases demonstrating the impracticality of utilizing similar approaches as a post-processing step.

We also run our mTSP formulation on the output of Line3D++ and observe that small variations lead to large topological errors in the final output. This demonstrates the importance of our 3D curve selection formulation (Section 4.2).

Finally, in Figure 2, we provide an example depth map of a wiry object captured from a commodity depth sensor (Microsoft Kinect). Even though we paint the originally black wire model with a diffuse paint, the sensor fails to provide any useful depth information for the thin wires that is required for surface reconstruction.

6.4 Limitations

Our method provides an effective and practical solution to a very challenging problem: we digitize wire objects from as few as 3 images in the form of a global curve network readily available for post processing and fabrication. Nevertheless, there are certain limitations we would like to address in future work. Failures in extracting reliable 2D image curves, *e.g.*, due to cluttered background or insufficient contrast between the object and the background, will degrade the performance of our approach like any other image-based method operating at the level of curves. We assume the camera parameters of the input images to be given and rely on surrounding texture objects to obtain this information. Significant deviations from the true camera parameters will potentially result in noisy 3D curve segment reconstructions and failures in the wire decomposition. A joint framework that optimizes both for the camera parameters and the 3D wires is an interesting research direction. Finally, our method is not designed to handle very dense wire sculptures such as shown in Figure 9. Such examples are closer to surfaces instead of 1D elements and other modeling paradigms (*e.g.*, procedural modeling) would be more suitable.

7 CONCLUSION AND FUTURE WORK

We present an image-based reconstruction method of wire objects using as few as 3 images as input. Our method exploits unique characteristics of wire objects (*simplicity* – the object is composed of a few wires, and *smoothness* – each wire is bent smoothly) to recover the global 3D wire decomposition. We represent each wire as a curve in our reconstructions to facilitate editing and physical fabrication of the results. Combined with the power of wire sculpting, we believe that this paves the road to using wires as a rapid prototyping tool to bring creative ideas into the digital design process.

In addition to addressing the limitations of our method discussed in the previous section, there are several exciting future research directions. For our experiments, we utilized 3 input views for each example, which all together provide a full coverage of the input object. Exploring strategies for *next best view selection* in the context of wire objects is a promising research direction. Our current approach focuses on objects made by bending and connecting multiple wires. While bending is one of the most prominent techniques used in wire art, there are other techniques, *e.g.*, twisting two wires as seen in the bird example in Figure 9 and wrapping thin wires around thicker ones. Exploring such techniques as additional priors in the

reconstruction method (*e.g.*, physical stability of the reconstructed object) is a promising direction likely to expand the scope of the objects that can be captured accurately. Finally, other future directions include combining our work with an iWires (Gal et al. 2009) type of deformation tool to explore editing options and applying the proposed techniques to medical imaging where reconstruction of thin structures from a few views is necessary.

ACKNOWLEDGMENTS

We thank our reviewers for their invaluable comments. We also thank Aron Monszpart, Tuanfeng Y. Wang, Xiaoguang Han, and Paul Guerrero for their help, comments, and ideas. This work was partially funded by the ERC Starting Grant SmartGeometry (StG-2013-335373), the Research Grant Council of Hong Kong (HKU 717813E, 17209815, 17211017), an UCL visiting student program, and gifts from Adobe.

REFERENCES

- Caroline Baillard, Cordelia Schmid, Andrew Zisserman, and Andrew Fitzgibbon. 1999. Automatic Line Matching and 3D Reconstruction of Buildings from Multiple Views. In *ISPRS Conf. on Automatic Extraction of GIS Objects from Digital Imagery*. 69–80. DOI: <https://doi.org/10.1023/A:1011104020586>
- Matthew Berger, Andrea Tagliasacchi, Lee M. Seversky, Pierre Alliez, Joshua A. Levine, Andrei Sharf, and Claudio T. Silva. 2014. State of the Art in Surface Reconstruction from Point Clouds. In *CGF Eurographics - State of the Art Reports*. DOI: <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 (Feb. 2001), 171–182. DOI: <https://doi.org/10.1023/A:1011104020586>
- Tao Chen, Zhe Zhu, Ariel Shamir, Shi-Min Hu, and Daniel Cohen-Or. 2013. 3-Sweep: Extracting Editable Objects from a Single Photo. In *ACM SIGGRAPH Asia*. ACM, New York, NY, USA, Article 195, 10 pages. DOI: <https://doi.org/10.1145/2508363.2508378>
- Charlotte Delmas, Marie-Odile Berger, Erwan Kerrien, Cyril Riddell, Yves Troussel, René Anxionnat, and Serge Bracard. 2015. Three-dimensional curvilinear device reconstruction from two fluoroscopic views. In *SPIE, Medical Imaging 2015: Image-Guided Procedures, Robotic Interventions, and Modeling*, Vol. 9415. San Diego, CA, France, 94150F. DOI: <https://doi.org/10.1117/12.2081885>
- Ricardo Fabbri and Benjamin Kimia. 2010. 3D curve sketch: Flexible curve-based stereo reconstruction and calibration. In *IEEE CVPR*. 1538–1545. DOI: <https://doi.org/10.1109/CVPR.2010.5539787>
- Ricardo Fabbri and Benjamin B. Kimia. 2016. Multiview Differential Geometry of Curves. *IJCV* 120, 3 (Dec. 2016), 324–346. DOI: <https://doi.org/10.1007/s11263-016-0912-7>
- Xinyi Fan, Linguang Zhang, Benedict Brown, and Szymon Rusinkiewicz. 2016. Automated View and Path Planning for Scalable Multi-object 3D Scanning. In *ACM SIGGRAPH Asia*. ACM, New York, NY, USA, Article 239, 13 pages. DOI: <https://doi.org/10.1145/2980179.2980225>
- Yasutaka Furukawa and Jean Ponce. 2010. Accurate, Dense, and Robust Multiview Stereopsis. *IEEE PAMI* 32, 8 (Aug. 2010), 1362–1376. DOI: <https://doi.org/10.1109/TPAMI.2009.161>
- Ran Gal, Olga Sorkine, Niloy J. Mitra, and Daniel Cohen-Or. 2009. iWIRES: An Analyze-and-edit Approach to Shape Manipulation. *ACM Trans. Graph.* 28, 3, Article 33 (July 2009), 10 pages. DOI: <https://doi.org/10.1145/1531326.1531339>
- Michael Goesele, Noah Snavely, Brian Curless, Hugues Hoppe, and Steven M. Seitz. 2007. Multi-View Stereo for Community Photo Collections. In *IEEE ICCV*. 1–8. DOI: <https://doi.org/10.1109/ICCV.2007.4408933>
- Inc. Gurobi Optimization. 2016. Gurobi Optimizer Reference Manual. (2016). <http://www.gurobi.com>
- Richard Hartley and Andrew Zisserman. 2004. *Multiple View Geometry in Computer Vision* (second ed.). Cambridge University Press, ISBN: 0521540518.
- 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)*.
- Manuel Hofer, Michael Maurer, and Horst Bischof. 2016. Efficient 3D scene abstraction using line segments. *Computer Vision and Image Understanding* (2016). DOI: <https://doi.org/10.1016/j.cviu.2016.03.017>
- Manuel Hofer, Andreas Wendel, and Horst Bischof. 2013. Line-based 3D reconstruction of wiry objects. In *Computer Vision Winter Workshop (CVWW)*.
- Wei Hong, Yi Ma, and Yizhou Yu. 2004. Reconstruction of 3-D Symmetric Curves from Perspective Images without Discrete Features. In *ECCV*. Springer Berlin Heidelberg, Berlin, Heidelberg, 533–545. DOI: https://doi.org/10.1007/978-3-540-24672-5_42

- Hui Huang, Shihao Wu, Daniel Cohen-Or, Minglun Gong, Hao Zhang, Guiqing Li, and Baoquan Chen. 2013. L1-medial Skeleton of Point Cloud. In *ACM SIGGRAPH*. ACM, New York, NY, USA, Article 65, 8 pages. DOI: <https://doi.org/10.1145/2461912.2461913>
- Emmanuel Iarussi, Wilmot Li, and Adrien Bousseau. 2015. WrapIt: Computer-assisted Crafting of Wire Wrapped Jewelry. In *ACM SIGGRAPH Asia*. ACM, New York, NY, USA, Article 221, 8 pages. DOI: <https://doi.org/10.1145/2816795.2818118>
- 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*. San Francisco, CA.
- Fredrik Kahl and Jonas August. 2003. Multiview reconstruction of space curves. In *IEEE ICCV*. 1017–1024 vol.2. DOI: <https://doi.org/10.1109/ICCV.2003.1238461>
- Imdat Kara and Tolga Bektas. 2006. Integer linear programming formulations of multiple salesman problems and its variations. *European Journal of Operational Research* 174, 3 (2006), 1449 – 1458. DOI: <https://doi.org/10.1016/j.ejor.2005.03.008>
- Peter Kovsesi. 2017. Peter's Functions for Computer Vision. (2017). <http://www.peterkovsesi.com/matlabfns/>
- Guo Li, Ligang Liu, Hanlin Zheng, and Niloy J. Mitra. 2010. Analysis, Reconstruction and Manipulation Using Arterial Snakes. In *ACM SIGGRAPH Asia (SIGGRAPH ASIA '10)*. ACM, New York, NY, USA, Article 152, 10 pages. DOI: <https://doi.org/10.1145/1866158.1866178>
- Tobias Martin, Juan Montes, Jean-Charles Bazin, and Tiberiu Popa. 2014. Topology-aware Reconstruction of Thin Tubular Structures. In *SIGGRAPH Asia 2014 Technical Briefs (SA '14)*. ACM, New York, NY, USA, Article 12, 4 pages. DOI: <https://doi.org/10.1145/2669024.2669035>
- Eder Miguel, Mathias Lepoutre, and Bernd Bickel. 2016. Computational Design of Stable Planar-Rod Structures. *ACM SIGGRAPH* 35, 4 (2016).
- Irina Nurutdinova and Andrew Fitzgibbon. 2015. Towards Pointless Structure from Motion: 3D Reconstruction and Camera Parameters from General 3D Curves. In *IEEE ICCV*. IEEE Computer Society, Washington, DC, USA, 2363–2371. DOI: <https://doi.org/10.1109/ICCV.2015.272>
- Jack M. Ogden. 1991. Classical gold wire. Some aspects of its manufacture and use. *Jewellery Studies* (1991), 95–105, Abb.
- Dushyant Rao, Soon-Jo Chung, and Seth Hutchinson. 2012. CurveSLAM: An approach for vision-based navigation without point features. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. 4198–4204. DOI: <https://doi.org/10.1109/IROS.2012.6385764>
- Luc Robert and Olivier D. Faugeras. 1991. Curve-based stereo: figural continuity and curvature. In *IEEE CVPR*. 57–62. DOI: <https://doi.org/10.1109/CVPR.1991.139661>
- Noah Snavely, Steven M. Seitz, and Richard Szeliski. 2006. Photo Tourism: Exploring Photo Collections in 3D. In *ACM SIGGRAPH (SIGGRAPH '06)*. ACM, New York, NY, USA, 835–846. DOI: <https://doi.org/10.1145/1179352.1141964>
- Damien Teney and Justus Piater. 2012. Sampling-Based Multiview Reconstruction without Correspondences for 3D Edges. In *Int. Conf. on 3D Imaging, Modeling, Processing, Visualization Transmission*. 160–167. DOI: <https://doi.org/10.1109/3DIMPVT.2012.28>
- Inc. The MathWorks. 2016. Optimization Toolbox User's Guide. (2016). <http://www.mathworks.com/access/helpdesk/help/toolbox/optim/>
- Anil Usumezbas, Ricardo Fabbri, and Benjamin B. Kimia. 2016. From Multiview Image Curves to 3D Drawings. In *ECCV*. 70–87. DOI: https://doi.org/10.1007/978-3-319-46493-0_5
- Changchang Wu. 2011. VisualSFM: A Visual Structure from Motion System. (2011). <http://ccwu.me/vsfm/>
- Changchang Wu, Sameer Agarwal, Brian Curless, and Steven M. Seitz. 2011. Multicore bundle adjustment. In *IEEE CVPR*. 3057–3064. DOI: <https://doi.org/10.1109/CVPR.2011.5995552>
- Shihao Wu, Wei Sun, Pinxin Long, Hui Huang, Daniel Cohen-Or, Minglun Gong, Oliver Deussen, and Baoquan Chen. 2014. Quality-driven Poisson-guided Autoscanning. In *ACM SIGGRAPH Asia*. ACM, New York, NY, USA, Article 203, 12 pages. DOI: <https://doi.org/10.1145/2661229.2661242>
- 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 (Sep 2005), 1746–1762. DOI: <https://doi.org/10.1364/JOSAA.22.001746>
- Feilong Yan, Andrei Sharf, Wenzhen Lin, Hui Huang, and Baoquan Chen. 2014. Proactive 3D Scanning of Inaccessible Parts. In *ACM SIGGRAPH Asia*. ACM, New York, NY, USA, Article 157, 8 pages. DOI: <https://doi.org/10.1145/2601097.2601191>
- 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. In *ACM SIGGRAPH Asia*. ACM, New York, NY, USA, Article 202, 12 pages. DOI: <https://doi.org/10.1145/2661229.2661241>

APPENDIX

When a particular epipolar line becomes tangent to or overlaps with a 2D curve segment, multiple intersection points close to each

other on the same curve are generated. These cluttered intersections do not define meaningful disjoint curves and cause difficulty in reconstructing 3D curve segments in the concerned region. We propose a curve fitting strategy to handle such degenerate cases.

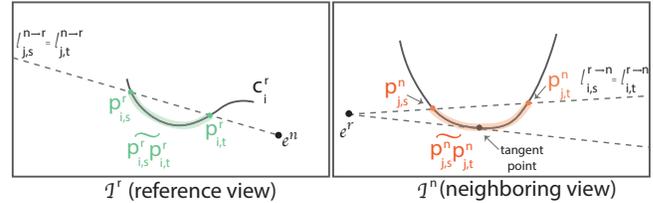


Fig. 10. In case an epipolar line becomes tangent or overlaps with a 2D curve segment, we solve the resulting degenerate case by a curve fitting strategy.

When an epipolar line, ℓ , is tangent to a smooth curve at a point of non-vanishing curvature, the part of the curve in a neighborhood of the tangent point lies entirely on the same side of ℓ . Given the epipolar line ℓ that passes through a point p sampled on a 2D curve segment, we identify p as a tangent point if the m -nearest samples ($m = 25$) of p on the curve all lie on the same side of ℓ . This neighborhood is unstable to compute exact intersections with ℓ .

Given a 2D curve segment c_i^r in a reference view I^r , assume the epipolar line, $\ell_{i,s}^{r \rightarrow n}$, of a sample point $p_{i,s}^r$ on c_i^r in a neighboring view I^n is at a threshold distance ($\lambda = 8$) from a tangent point detected on a curve segment c_j^n in I^n (see Figure 10). We denote the two intersection points between $\ell_{i,s}^{r \rightarrow n}$ and c_j^n as $p_{i,s}^n$ and $p_{i,t}^n$. Moving along c_i^r will yield another point $p_{i,t}^r$ whose epipolar line will be the same as $\ell_{i,s}^{r \rightarrow n}$. Under epipolar constraints, $p_{i,s}^r$ and $p_{i,t}^r$ have the same epipolar line in I^r and this line passes through $p_{i,s}^r$ and $p_{i,t}^r$. Thus the curve region, $\overline{p_{i,s}^r p_{i,t}^r}$ in I^r and the curve region, $\overline{p_{i,s}^n p_{i,t}^n}$ in I^n are corresponding regions.

There are two possible pairings between the matching regions $\overline{p_{i,s}^r p_{i,t}^r}$ and $\overline{p_{i,s}^n p_{i,t}^n}$ for subsequent 3D curve reconstruction, that is: (i) $(p_{i,s}^r, p_{i,s}^n)$ for one end and $(p_{i,t}^r, p_{i,t}^n)$ for the other end; or (ii) $(p_{i,s}^r, p_{i,t}^n)$ for one end and $(p_{i,t}^r, p_{i,s}^n)$ for the other end. For each pairing, we first compute the 3D positions of the paired endpoints and then fit a polyline between these 3D endpoints that is smooth and matches the two image observations as in Equation 9. We keep both of the results as part of the candidate 3D curve segment set.

If an epipolar line is tangent to a curve at an inflection point, we detect the unstable neighborhood by checking if the m -nearest points of the intersection point lie within a narrow strip centered around the epipolar line. The remaining curve region process remains unchanged.