Multi-view Reconstruction of Wires using a Catenary Model

Ratnesh Madaan, Michael Kaess, and Sebastian Scherer

***Abstract*— Reliable detection and reconstruction of wires is one of the hardest problems in the UAV community, with a wide ranging impact in the industry in terms of wire avoidance capabilities and powerline corridor inspection. In this work, we introduce a real-time, model-based, multi-view algorithm to reconstruct wires from a set of images with known camera poses, while exploiting their natural shape – the catenary curve. Using a model-based approach helps us deal with partial wire detections in images, which may occur due to natural occlusion and false negatives. In addition, using a parsimonious model makes our algorithm efficient as we only need to optimize for 5 model parameters, as opposed to hundreds of 3D points in bundle-adjustment approaches. Our algorithm obviates the need for pixel correspondences by computing the reprojection error via the distance transform of binarized wire segmentation images. Further, we make our algorithm robust to arbitrary initializations by introducing an on-demand, approximate extrapolation of the distance transform based objective. We demonstrate the effectiveness of our algorithm against false negatives and random initializations in simulation, and show qualitative results with real data collected from a small UAV.**

I. Introduction

Reliable wire and powerline detection is a critical capability for flying robots, as wires pose a serious collision hazard for vehicles of all size [1–3]. In addition, wire detection is useful for inspection and monitoring tasks, e.g., in powerline corridor inspection [4–7]. However, thin wires and powerlines are among the hardest obstacles to detect, regardless of sensor modality [3]. Due to their small cross-section and low reflectance, detection with active sensors such as radar and lidar is only reliable with high-resolution devices, which are expensive and unsuitable for small UAVs due to their weight and size constraints. Cameras – in either a monocular or stereo configuration – are a better alternative, as they offer high-resolution sensing at a comparatively low weight, cost, and size. However, current vision-based methods still struggle to detect wires in scenes with cluttered background, low contrast, and wires of barely visible width [8–14].

In this work, we present an approach for detection and 3D reconstruction of wires using a single monocular camera. For monocular detection, we leverage our previous work to obtain pixel-wise wire segmentations from RGB images [14]. For reconstruction, we present a novel model-based multi-view approach that represents wires as 3D catenary curves – their natural shape under uniform gravity. The reconstruction process attempts to recover these curves (or more specifically, the parameters of these curves) by minimizing a reprojection loss at runtime.

Our key insight is that using a parsimonious model-based approach allows us to reconstruct wires with more fidelity by exploiting the natural physical structure of the problem, even in the presence of errors (due to occlusion or missed detections) in the wire segmentation process. Compared to works that do not exploit this structure, such as voxel-grid mapping (e.g., [15]) our method has several advantages. Our algorithm can more effectively deal with spurious (false positive) segments, and predict the presence of wires despite missed (false negative) wire segments. This is not only useful for obstacle avoidance, but also in applications where exact reconstruction of wires is desired [16, 17]. We note that while this paper uses a monocular camera setup – due to its relative simplicity and reduced weight and cost requirements – the algorithm can naturally be adapted to a stereo setup, which we leave for future work.

To our knowledge, this is the first work presenting a multi-view wire detection and reconstruction method exploiting this kind of a model. While catenary models have been used before for wire detection and reconstruction from airborne lidar data [18, 19], recovering this model from images is a considerably different task than recovering it from point cloud data.

We evaluate our method by generating a dataset of problems with random camera views in simulation. We show that our algorithm is robust against false negatives and arbitrary initialization. We also demonstrate our method on real data gathered with a small UAV.

Our key contributions are a real-time, model-based, multi-view algorithm utilizing a minimal parameterization of the catenary, which obviates the need for pixel correspondences and is robust to arbitrary initializations.

II. Related Work

While catenary models have been used before for wire extraction from airborne lidar data [18, 19], this is the first work proposing a multiview algorithm. [18] first proposed to use catenary models and used numerical methods for optimization. [19] used a catenary model for powerline reconstruction from lidar data using non-linear least squares. Catenary models have also found uses beyond the task of wire detection; they have been used previously for analyzing dental arches [20, 21] and breast curvature [22, 23].

In the realm of generic thin obstacle reconstruction, [24] propose an incremental line-based 3D reconstruction algorithm that performs Structure from Motion (SfM) to recover camera poses and then attempts to recover a line-based model. However their method is not real-time, as they solve for a combinatorial number of possible matches of line features using epipolar geometry. Similar combinatorial

The authors are with the Robotics Institute at Carnegie Mellon University, Pittsburgh, Pennsylvania, USA. ratneshmadaan@gmail.com, [kaess, basti]@cs.cmu.edu

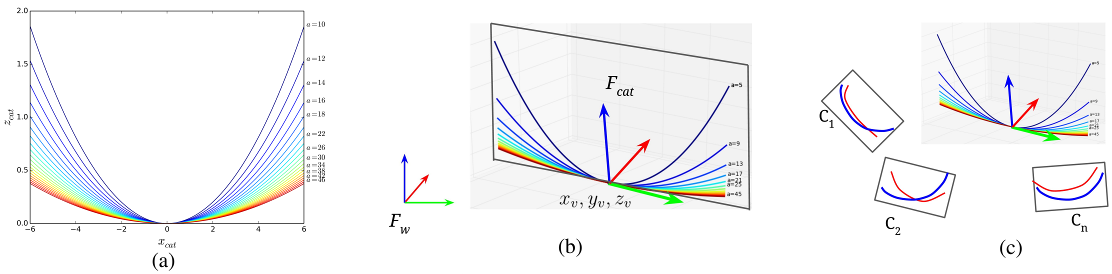


Fig. 1: (a) Catenary curves from Eq. 1 in the local catenary frame, F_{cat} with varying sag parameter (a) values. (b) Our 3D catenary model is obtained by defining F_{cat} with respect to a world frame, F_w. (c) Schematic of problem statement.

matching is observed in curve matching in [25]'s work on image based reconstruction of wire art.

The distance transform was first proposed as a solution to address the difficulty of data association for line and curve features by [26]. Subsequently, this idea has been used in recent works on subdivision curve reconstruction [27] and edge-based visual odometry [28, 29]. Recently, [30] developed a promising thin obstacle mapping approach building on previous work on edge-based visual odometry [31].

III. Approach

Given a set of images with known camera poses containing a single wire, we first perform pixel-wise wire segmentation with a convolutional neural network (CNN) using our previous work [14]. We then binarize the wire segmentation confidence maps by thresholding them. Further, we estimate the catenary model parameters via non-linear least squares optimization. To minimize the reprojection error, we use a computationally inexpensive distance transform as our loss function, thereby obviating the need to find wire pixel correspondences.

Given N views observing a single wire with known camera poses, the objective is to recover the wire model parameters Θ, minimizing the reprojection error $f(\Theta)$. Fig. 1c depicts this visually. We are given N views of binary wire segmentations (shown in red), along with their corresponding camera poses $\{C_1, C_2, ..., C_N\}$. The multiple curves in 3D represent various model hypotheses. The blue curves in each view denote the projection of the current hypothesis in each frame.

In the following sections, we discuss the model Θ in Section III-A, the objective function $f(\Theta)$ in III-B, and the optimization process in III-C.

A. Catenary Model

A wire with uniform density and thickness, when suspended by its two ends under uniform gravity, follows the shape of the planar, single-parameter, catenary curve [32]:

$$z_{cat} = a\left(cosh\left(\frac{x_{cat}}{a}\right) - 1\right),\ y_{cat} = 0 \tag{1}$$

where a is the *catenary sag parameter*, and $\mathbf{P}_{cat} = \{x_{cat}, y_{cat}, z_{cat}\}$ are the coordinates of points on the curve in the local catenary frame, F_{cat}. As the value of the sag parameter (a) increases, the curve flattens, as visualized in Fig. 1a. We define the *catenary vertex* to be the lowermost point of the curve, which is also the origin of F_{cat}. Now, this planar catenary can easily be extended to 3D by defining F_{cat} w.r.t. an arbitrary world frame F_w, related to F_{cat} by a relative translation and rotation.

Let the tuple $P_v = (x_v, y_v, z_v)$ be the 3D coordinates of the catenary vertex in F_w. Note that the gravity vector lies in the plane of the catenary; thus, if we choose an F_w such that its $+Z$ axis is anti-parallel to the negative gravity vector, there is only one degree of rotational freedom to consider when defining F_{cat}. We denote this rotational parameter, expressing the relative yaw between F_{cat} and F_w, as ψ. Therefore, the parameters of our catenary model are given by the 5-tuple $\Theta = \{x_v, y_v, z_v, \psi, a\}$. The parametrization encodes a 3D transform minus the roll and pitch in the rotation part, in addition to the sag parameter. We can obtain the points in F_w ($\mathbf{P}_w$) from the points in F_{cat} ($\mathbf{P}_{cat}$) by transformation matrices R and T built from Θ:

$$\Theta = \{x_v, y_v, z_v, \psi, a\} \tag{2}$$

$$\mathbf{P}_w(\Theta) = R(\psi)\,\mathbf{P}_{cat} + T(x_v, y_v, z_v) \tag{3}$$

So far, we have not discussed the wire length. We assume that it is known, and set it to an appropriately large constant, l_{cat}. Two possible alternatives, at the cost of extra parameters, would be to add the length as an extra model parameter, or to parameterize the two endpoints instead of the lowermost point. We are also not (explicitly) modelling wire thickness. Instead, we assume an ideal wire, and when computing reprojection error, we render the wire to 2D by projecting a fixed, uniformly sampled number of points M. Thus, $\mathbf{P}_{cat}$ and $\mathbf{P}_w$ above are both matrices of size $M \times 3$. For our simulation experiments, M was chosen as 100 and l_{cat} was set to 50 m. The assumption of a fixed l_{cat} can be relaxed by using 3D geometry: finding the intersection of viewing frustums and using the wire segmentations can give as an upper bound on the length of the curve.

With our model, the optimization variables are limited to five in number, instead of tens to thousands of points common in bundle-adjustment problems [33]. This parsimonious parameterization makes the model interpretable and allows efficient optimization, potentially allowing the current formulation to scale to multiple wires.

B. Objective Function

In order to recover the model parameters Θ, the projections of our inferred catenary model on each view should be as consistent as possible with the observed wire detections. To find the reprojection error, the challenge lies in defining the

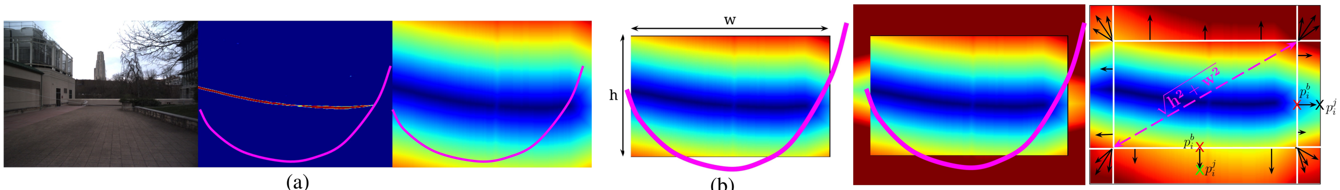


Fig. 2: (a) L-R: Input image with a single wire, wire segmentation confidence map (red is higher), distance transform of binarized segmentation. Pink curve represents the catenary hypothesis' projection. (b) However the pink projection may fall outside the image coordinates. (c) Extrapolating from the image center leads to discontinuity at the image boundary. (d) Extrapolation from the boundary pixels mitigates this issue. The base of the arrows indicate the corresponding boundary pixel for the pixel at the arrow head. The white lines divide the area outside the image into 8 regions.

wire-to-wire pixel correspondences, which is a known issue when performing reconstruction with thin features like lines, edges, and curves. To address this problem, we draw on the distance transform approach used in previous works [26–29].

Specifically, we first binarize the wire segmentations obtained via [14], and then compute the distance transform for each view. We normalize it by its maximum value in the image to obtain D_i, such that $D_i \in [0, 1]$. We can then define a *residual vector*, $\mathbf{r} \in \mathbb{R}^{MN}$, by projecting the M catenary points, $\mathbf{P}_{cat}$ onto each of the N views, and stacking the distance transform values. For each i^{th} point P_i in $\mathbf{P}_{cat}$, let its projection in the j^{th} view be the pixel, $p_{ij} = (u_{ij}, v_{ij})$. The residual for a pixel p_{ij} is then given by $r_{ij} = D_i(p_{ij})$. We can then obtain $\mathbf{r}$ by stacking all the r_{ij} values. This is visualized in Fig. 2a.

Another issue we must address in this formulation is defining a loss value for any *invalid pixels*, whose coordinate values fall outside the image height and width limits (h, w), as shown in Fig. 2b. A naive option is to apply padding on the images before computing the distance transform. However, it is difficult to choose the appropriate amount of padding *a priori*, and excessively large padding increases the computational cost of the optimization. We propose instead an *on-demand, approximate extrapolation* of D_i for such invalid pixels. The extrapolation is continuous and differentiable at image boundaries, aiding the non-linear optimization.

For any i^{th} invalid pixel in the j^{th} view, given by p_{ij}, we first find a corresponding boundary pixel p_{ij}^b by defining eight regions around the image as shown in Fig. 2d. Then, we compute the Euclidean distance between the pixels p_{ij} and p_{ij}^b, $d(p_{ij}, p_{ij}^b)$, and divide it by the maximum possible value of the distance transform, the image diagonal $\sqrt{h^2 + w^2}$. Finally, the loss value at any invalid pixel is obtained by adding this value to the residual value at the corresponding boundary pixel, by $D_i(p_{ij}^b)$.

This can be expressed formally as:

$$r_{ij} = \begin{cases} D_i(p_{ij}), & \text{if } u_{ij} \in [0, h), v_{ij} \in [0, w) \\ D_i(p_{ij}^b) + \frac{d(p_{ij}, p_{ij}^b)}{\sqrt{h^2+w^2}}, & \text{otherwise} \end{cases} \tag{4}$$

$$u_{ij}^b = \left\{ \begin{matrix} 0, & \text{if } u_{ij} < 0 \\ h-1, & \text{if } u_{ij} \geq h \end{matrix} \right\} , \quad v_{ij}^b = \left\{ \begin{matrix} 0, & \text{if } v_{ij} < 0 \\ w-1, & \text{if } v_{ij} \geq w \end{matrix} \right\}$$

We need to compute D_i for each of the N views only once, and reuse it for each iteration of the optimization. This takes less than 1 *ms*. For any pixel that falls outside the image, we can compute the residual by extrapolating D_i in an on-demand fashion.

$$\begin{tikzcd}
\Theta \arrow[r, shift left, "f_{\Theta \mapsto P_W}"] & P_W \arrow[l, shift left, "J_{\Theta \mapsto P_W} = \frac{\partial P_W}{\partial \Theta}"] \arrow[r, shift left, "f_{P_w \mapsto P_{C_i}}"] & P_{C_i} \arrow[l, shift left, "J_{P_w \mapsto P_{C_i}} = \frac{\partial P_{C_i}}{\partial P_W}"] \arrow[r, shift left, "f_{P_{C_i} \mapsto p_{C_i}}"] & p_{C_i} \arrow[l, shift left, "J_{P_{C_i} \mapsto p_{C_i}} = \frac{\partial p_{C_i}}{\partial P_{C_i}}"] \arrow[r, shift left, "f_{p_{C_i} \mapsto r_i}"] & r_i \arrow[l, shift left, "J_{p_{C_i} \mapsto r_i} = \frac{\partial r_i}{\partial p_{C_i}}"]
\end{tikzcd}$$

$$\mathbf{J} = \frac{\partial \mathbf{r}}{\partial \Theta} = \frac{\partial \text{Distance transform value}}{\partial \text{Pixel Coordinates}} \frac{\partial \text{ Pixel Coordinates}}{\partial \text{Points in Cam Frame}} \frac{\partial \text{Points in Cam Frame}}{\partial \text{ Points in World Frame}} \frac{\partial \text{ Points in World Frame}}{\partial \text{ Catenary Model Params}}$$

Fig. 3: Chain of transformations and Jacobians

C. Optimization Problem

As described in the previous sections, we first obtain the 3D curve in the world frame, $\mathbf{P}_w \in \mathbb{R}^{3M}$ from the model parameters Θ, using Eq. 1 and 3. Then, each point in $\mathbf{P}_w$ is transformed to each view's frame via the respective extrinsics matrices, to get $\mathbf{P}_{C_i} \in \mathbb{R}^{3M \times N}$. Next, the camera intrinsics are applied to get the projected pixels, $\mathbf{p}_{C_i} \in \mathbb{R}^{2M \times N}$. Finally, we use Eq. 4 to get a vector of residuals, $\mathbf{r} \in \mathbb{R}^{MN}$. Our objective function, $f(\Theta)$ is given by the squared sum of the vector of residuals. The chain of mappings is visualized in Fig. 3. Each mapping is differentiable and lends us a corresponding Jacobian. We can obtain the analytical Jacobian of $\mathbf{r}$ w.r.t. $\mathbf{\Theta}$ by the chain rule:

$$f(\Theta) = \frac{1}{2}\mathbf{r}^\top \mathbf{r} \ , \ \mathbf{J} = \frac{\partial \mathbf{r}}{\partial \Theta} = \frac{\partial \mathbf{r}}{\partial \mathbf{p}_C} \frac{\partial \mathbf{p}_C}{\partial \mathbf{P}_C} \frac{\partial \mathbf{P}_C}{\partial \mathbf{P}_w} \frac{\partial \mathbf{P}_w}{\partial \Theta} \tag{5}$$

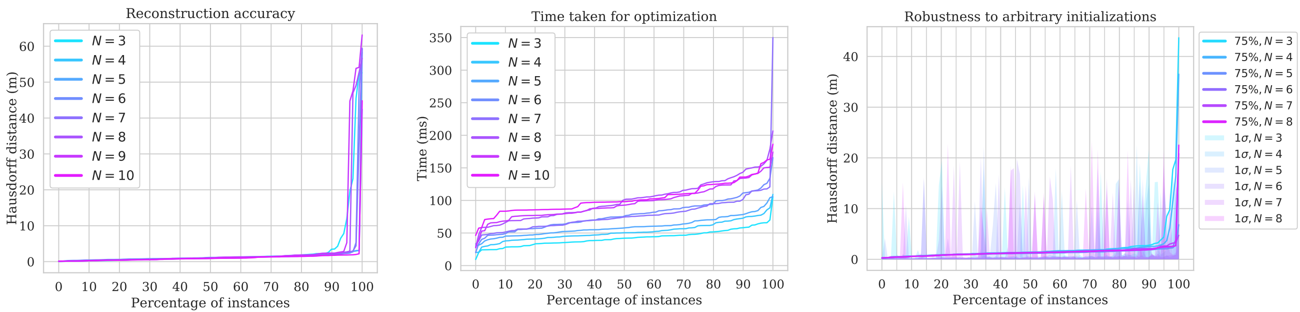
The derivation of $\mathbf{J}$ is shown in the Appendix. We use the subscript C to indicate stacking of pixels or points corresponding to all camera views. The analytical Jacobian is faster and more stable as compared to the numerical one; moreover, we observed that due to the hyperbolic term in the catenary's equation, the numerical Jacobian can be unstable for certain combinations of a and ψ values.

We optimize for this non-linear least squares objective using the Trust Region Reflective algorithm [34, 35] to recover Θ^*:

$$\Theta^* = \arg\min_{\Theta} f(\Theta) \tag{6}$$

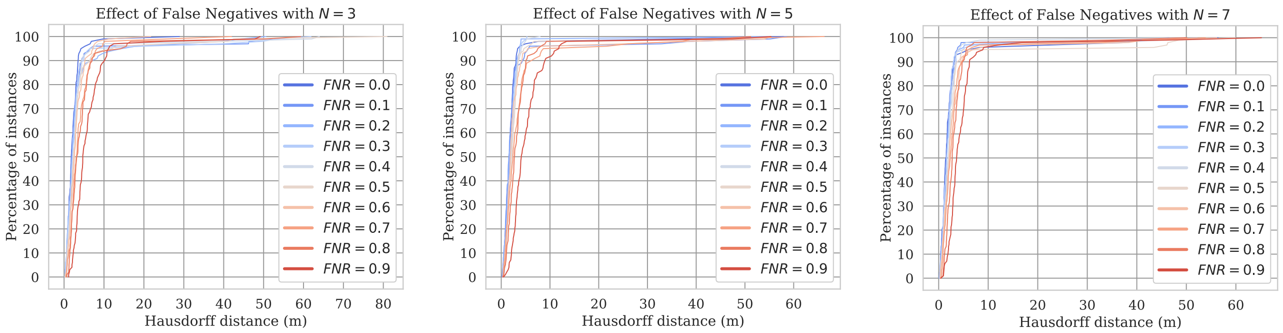
D. Implementation Details

We use OpenCV's distance transform function, which implements [36]. Specifically, we use a 3×3 mask and the Euclidean distance as the metric. This means that the distance is defined by the shortest path to the nearest wire pixel, when horizontal, vertical, or diagonal moves are allowed.



(a) (b) (c)

Fig. 4: Effect of (a) number of views (N) on the Hausdorff distance, and (b) time taken for optimization, over our datasets of 100 random scenarios. (c) Robustness to random initializations. Here we repeat the experiment for (a) but for 10 different intializations for each of 100 scenarios, and show the 75^{th} quartile Hausdorff distance, along with one standard deviation.



(a) (b) (c)

Fig. 5: Effect of systematically adding false negatives in wire segmentations on the Hausdorff distance for $N = 3, 5, 7$.

IV. Experiments and Results

A. Simulation

We simulate two types of scenarios:

- *Random scenarios*: Random camera views allow us to simulate inspection scenarios, where the goal is to reconstruct the wire with high accuracy by flying a drone around it. We first sample a *look-at* point on the catenary where the camera will be pointing towards. Then, we sample a random viewing direction relative to this point, via cylindrical coordinates. Once we have a viewing direction and a look-at point, we sample a distance along this vector which tells us where the camera will be placed relative to the wire.
- *Wire avoidance scenarios*: For avoidance scenarios, we limit the scope of the viewing direction, such that the cameras are all within a small angle, thereby mimicking the kind of views one might expect while flying a UAV head-on towards a wire.

A few qualitative results are shown in Fig. 7. For quantitative evaluation, we generate a dataset of 100 problems with random scenarios. We choose discretization $M = 100$, length $l_{cat} = 50$, and sample a distance between 50 to 150 *m* from the catenary to place cameras. We generate 100 such scenarios, for varying number of camera views, given by N.

Evaluation: We use the symmetric Hausdorff distance [37, 38] between the 3D point sets of the ground truth catenary $\mathbf{P}_{gt}$ and the resultant catenary solution returned by optimization $\mathbf{P}_{res}$, to evaluate the reconstruction accuracy:

$$\mathcal{H}(\mathbf{P}_{gt}, \mathbf{P}_{res}) = \max\big(h(\mathbf{P}_{gt}, \mathbf{P}_{res}), h(\mathbf{P}_{res}, \mathbf{P}_{gt})\big) \quad (7)$$

where $h(\mathbf{X}, \mathbf{Y}) = \max_{x \in \mathbf{X}} \min_{y \in \mathbf{Y}} d(x, y)$

Accuracy: Fig. 4a shows the performance of our method over the 100 problem instances of the random scenario dataset. We can see that our method works well for more than 90% of the cases, for which $\mathcal{H}$ is less than 5 *m*. We believe that the effect of adding more views is somewhat inconclusive owing to the small dataset size, relative placement of the camera views, ground truth catenary parameters, and initialization of the optimization. The high $\mathcal{H}$ values for a small percentage of cases indicate the failure of our method in a few cases. We investigate whether such failures are consistent despite of a different initialization, or can be mitigated by doing so in the upcoming paragraphs.

Time: Fig. 4b shows the time taken (in milliseconds) for optimization versus percentage of problem instances. As number of views increases to 10, we are always under 200 *ms*. The wire segmentation runs at approximately 3 Hz on a portable GPU (Nvidia TX-2), and our algorithm needs less time than that taken by monocular segmentation. Additionally, we stress test our algorithm for large number of views, as shown in Fig. 6. For $N = 50$, we converge within 1100 *ms*, and within 1500 *ms* for most cases with $N = 75$.

Robustness to arbitrary initializations: We verify the effectiveness of our objective function against arbitrary initializations – a common issue for mapping and localization algorithms – by repeating our experiment over the 100 scenarios, however by testing 10 different random initializations for each instance this time. We visualize the 75 percentile

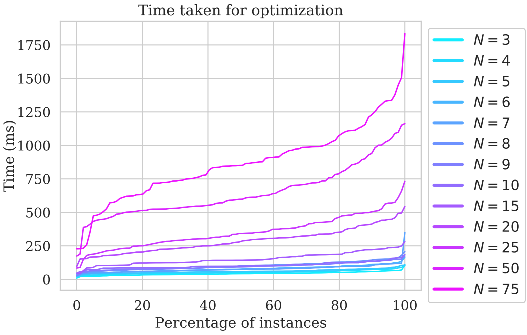


Fig. 6: Time taken by optimization for large number of views.

Hausdorff error, and 1 standard deviation in the resultant error plot in Fig. 4c. For $N = 3, 4$, we can see high accuracy of reconstruction for more than 95% of the problems. For $N \geq 5$, almost all the instances converge within a small error.

The occasional spikes in the standard deviation highlight that sometimes our algorithm fails to converge due to drastic initializations. However, the overall low number of the 75^{th} quartile indicates that our algorithm converges most of the time if one changes the initialization. This property can be leveraged in critical scenarios – if our algorithm fails, one can try a different random initialization due to the short term required for finding a solution, or run our algorithm in parallel with multiple initializations.

Robustness to false negatives: We test the robustness against false negatives by systematically removing true positive pixels from the projections of the ground truth curve in each view. We refer to the ratio of number of the pixels removed to the original number of projected pixels as *FNR* (false negative rate) in the context of this experiment. For each view, we select the pixels to remove randomly. We vary *FNR* from 0.0 to 0.9, in increments of 0.1, and show the results for $N = 3, 5, 7$ as shown in Fig. 5, due to constraints of space.

We can see that the hausdorff distance degrades as *FNR* increases, however it still within 10 *m* even when we remove 90% of the true positives. This indicates the effectiveness of our method in real world scenarios to predict where a wire could be in cases when it is partially detected, which is interestingly also a cause of wire strike accidents due to pilot error [1, 2].

B. Real Data

We collect and use a dataset logged on the DJI Matrice M100, retrofitted with in-house developed sensor suite consisting of a stereo camera, an FPGA stereo processor [39], a monocular color camera, an IMU, and a Nvidia TX2 ARM computer as shown in Fig. 8. For this section, we only use the monocular camera for detection and reconstruction. Camera pose estimation is obtained with a filtering approach using the IMU and built-in GPS. As the ground truth is hard to obtain, we demonstrate qualitative results on real data.

We extract and analyse 16 chronological views from the dataset, with a relative translation of at least 0.5 *m*, or a relative orientation of at least 18°(quaternion geodesic distance) between consecutive views. Fig. 9 shows two handpicked views to help examine the segmentation result.

Fig. 10a shows that the projection of the converged catenary hypothesis match the observed segmentations closely. Fig. 10b shows the odometry of the real data, along with the randomly initialized and the resultant curve. The real segmentations (in green) have a lot of false negatives, as expected. Our method is however able to handle this, unlike a voxel grid mapping method which would not be able to predict the presence of wires where they are not detected. Also, note that because the odometry is not ideal, we are not able to fit to the projections exactly.

We encourage the reader to look at the supplementary video for simulation and real results.

V. Conclusion

We proposed a multiview algorithm for wire reconstruction using a 5 parameter catenary model. Our algorithm is robust to bad initializations, which we validate on both real data, and simulated data with random camera poses and realistic scenarios.

VI. Future Work

To close the loop with a motion planner for tasks like wire avoidance and active perception, we need an estimate of the uncertainty of the reconstruction. This can be achieved by modeling P_v in the inverse-depth coordinates in the first view [40]. This new model, $\Psi = \{u_v^{C_1}, v_v^{C_1}, d_v^{C_1}, \psi, a\}$ is related with Θ by camera intrinsics and the extrinsics. We start by estimating the covariance of the solution Ψ^* with non-linear least squares theory: $\Sigma_\Psi^* = (J_{\Psi \mapsto r} J_{\Psi \mapsto r}^\top)^{-1}$. Then, we can get a first order estimate of the covariance of P_v by linearizing the $\Psi \mapsto \Theta$ mapping: $\Sigma_\Theta^* = J_{\Psi \mapsto \Theta} \Sigma_\Psi^* J_{\Psi \mapsto \Theta}^\top$. Similarly, we can linearize the catenary equation to get an estimate of the covariance for each point on the curve.

Secondly, real-world scenarios have multiple wires, and their projected curves may intersect each other. This can resolved by obtaining instance segmentations of the wires (by imposing a model in the image space itself via a polynomial), as well as curve-to-curve correspondences across all views. Then we can simply isolate each curve, and run our proposed algorithm on a per-wire basis.

Acknowledgements

We thank Daniel Maturana for his invaluable feedback and comprehensive reviews. This work was supported by Autel Robotics under award number A018532.

Appendix

Analytical Jacobian Derivation: In the following subsections, we show each component of $\mathbf{J}$ from Eq. 5.

A. Residual value r_{ij} w.r.t. i^{th} pixel in the j^{th} view, p_{ij}

r_{ij} is obtained from $p_{ij} = \{u_{ij}, v_{ij}\}$ as shown in Eq. 4. Differentiating Eq. 4, we get:

$$\frac{\partial r_{ij}}{\partial p_{ij}} = \begin{cases} \begin{bmatrix} \frac{\partial D_i}{\partial u_{ij}} & \frac{\partial D_i}{\partial v_{ij}} \end{bmatrix}, & \text{if } u_{ij} \in [0, h), v_{ij} \in [0, w) \\ \frac{1}{\sqrt{h^2+w^2}} \begin{bmatrix} \frac{u_{ij}-p_{ij}^b}{d(u_{ij}^b, p_{ij})} & \frac{v_{ij}-u_{ij}^b}{d(p_{ij}^b, p_{ij})} \end{bmatrix}, & \text{otherwise} \end{cases} \tag{8}$$

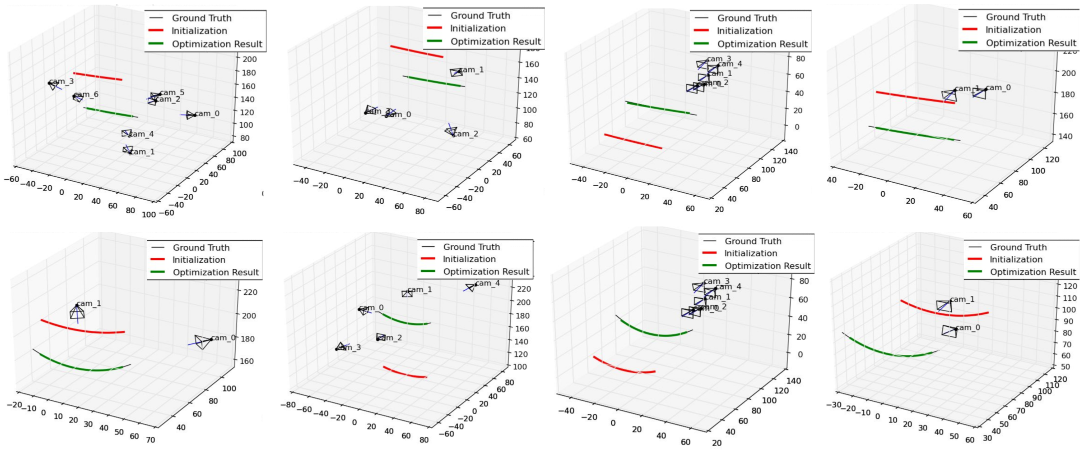


Fig. 7: Qualitative results on our simulation dataset. We show both random (first two columns) and avoidance scenarios (last two columns) here. The top row demonstrate situations where the wire is almost a straight line, which happens with large values of the sag parameter. The bottom row show results for lower values of a, where the catenary shape is evident.

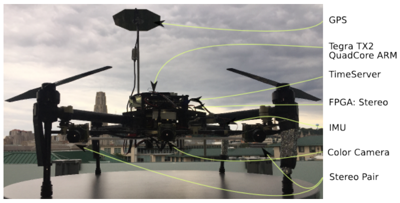


Fig. 8: Quadrotor platform used for experiments, equipped with a color (middle) monocular camera and an onboard ARM computer.

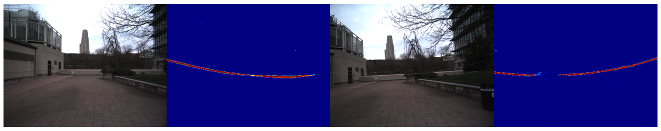
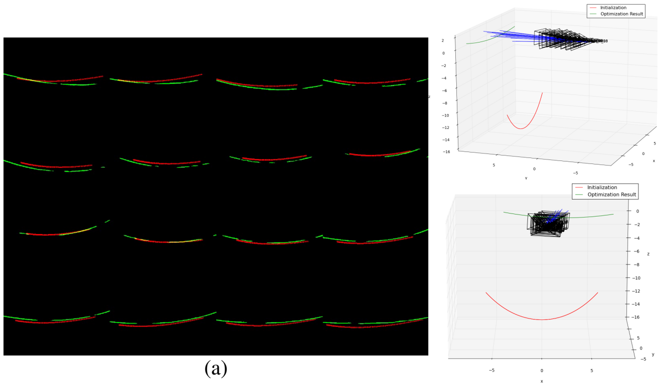


Fig. 9: Wire segmentation in 2 views from real data.

B. Coordinates of any pixel p_{C_i} w.r.t. corresponding 3D point P_{C_i} in the i^{th} view

$p_{C_i} = \{u_i, v_i\}$ is obtained from $P_{C_i} = \{X_i, Y_i, Z_i\}$ by camera intrinsics. Here, u_0, v_0 is the principal point and f_x, f_y are focal lengths.

$$p_{C_i} = \begin{bmatrix} u_i \\ v_i \end{bmatrix} = \begin{bmatrix} f_x(X_i/Z_i) + u_0 \\ f_y(Y_i/Z_i) + v_0 \end{bmatrix} \tag{9}$$



(a)

(b)

Fig. 10: Results with real data. (a) Montage of the 16 views used for reconstruction. Green shows detected wires, red shows the projection of the resultant catenary model. (b) 3D visualization: random initialization of catenary is shown in red, and the resultant catenary is shown in green.

We can differentiate the above to get:

$$\frac{\partial p_{C_i}}{\partial P_{C_i}} = \begin{bmatrix} f_x/Z_i & 0 & -f_x(X_i/Z_i^2) \\ 0 & f_y/Z_i & -f_y(Y_i/Z_i^2) \end{bmatrix} \tag{10}$$

C. 3D Points in the i^{th} view's frame $\mathbf{P}_{C_i}$ w.r.t. 3D points $\mathbf{P}_w$ in the world frame

$\mathbf{P}_{C_i}$ can be obtain from $\mathbf{P}_w$ via camera extrinsics:

$$\mathbf{P}_{C_i} = R_{ext}\mathbf{P}_w + T_{ext} \tag{11}$$

In our case, we assume to know the camera poses, hence extrinsics matrices are fixed. So the differentiating the above results in a constant term. We exploit this fact to make our implementation fast:

$$\frac{\partial \mathbf{P}_{C_i}}{\partial \mathbf{P}_w} = R_{ext} \tag{12}$$

D. $\mathbf{P}_w$ in the world frame w.r.t. catenary parameters Θ

From Eq. 1 and 3, we can write $\mathbf{P}_w$ in terms of Θ. Below, $X_{cat}, Y_{cat}, Z_{cat}$ refer to coordinates of the curve in the catenary frame.

$$\mathbf{P}_w(\Theta) = R(\psi)P_{cat} + T(x_v, y_v, z_v) \tag{13}$$

$$\begin{bmatrix} X_W \\ Y_W \\ Z_W \\ 1 \end{bmatrix} = \begin{bmatrix} cos(\psi) & -sin(\psi) & 0 & x_v \\ sin(\psi) & cos(\psi) & 0 & y_v \\ 0 & 0 & 1 & z_v \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_{cat} \\ 0 \\ a\left(cosh\left(\frac{X_{cat}}{a}\right) - 1\right) \\ 1 \end{bmatrix} \tag{14}$$

$$\begin{bmatrix} X_W \\ Y_W \\ Z_W \\ 1 \end{bmatrix} = \begin{bmatrix} X_{cat}cos(\psi) + x_v \\ X_{cat}sin(\psi) + y_v \\ a\left(cosh\left(\frac{X_{cat}}{a}\right) - 1\right) + z_v \\ 1 \end{bmatrix} \tag{15}$$

Differentiating the above, we get:

$$\frac{\partial \mathbf{P}_w}{\partial \Theta} = \begin{bmatrix} I_{33} & \begin{matrix} -X_{cat}sin(\psi) \\ X_{cat}cos(\psi) \\ 0 \end{matrix} & \begin{matrix} 0 \\ 0 \\ cosh\left(\frac{X_{cat}}{a}\right) - \frac{X_{cat}}{a}sinh\left(\frac{X_{cat}}{a}\right) - 1 \end{matrix} \end{bmatrix} \tag{16}$$

Now, we can write X_{cat} to get the above Jacobian expression in terms of the corresponding output (X_W) and input variables (x_v): $X_{cat} = (X_W - x_v)/cos(\psi)$

References

[1] Flight Safety Australia, “Wire, the invisible enemy,” http://www.flightsafetyaustralia.com/2017/11/wire-the-invisible-enemy/, 2017, online; Nov 20, 2017. I, IV-A

[2] Australian Transport Safety Bureau, “Wire-strike Accidents in General Aviation, Data Analysis 1994 to 2004,” https://www.atsb.gov.au/media/32640/wirestrikes_20050055.pdf, 2006, september 2006. IV-A

[3] S. Lau, “A plan for reducing wire strike accidents,” *Helo Air Safety, Professional Pilot*, pp. 84–87, 2012. I

[4] J. Katrasnik, F. Pernus, and B. Likar, “A survey of mobile robots for distribution power line inspection,” *IEEE Transactions on Power Delivery*, vol. 25, no. 1, pp. 485–493, 2010. I

[5] L. Ma and Y. Chen, “Aerial surveillance system for overhead power line inspection,” *Center for Self-Organizing and Intelligent Systems (CSOIS), Utah State Univ., Logan, Tech. Rep. USU-CSOIS-TR-04-08 (September 2000)*, 2004.

[6] C. Deng, S. Wang, Z. Huang, Z. Tan, and J. Liu, “Unmanned aerial vehicles for power line inspection: A cooperative way in platforms and communications,” *J. Commun*, vol. 9, no. 9, pp. 687–692, 2014.

[7] G. J. Adabo, “Long range unmanned aircraft system for power line inspection of brazilian electrical system,” *Journal of Energy and Power Engineering*, vol. 8, no. 2, 2014. I

[8] R. Kasturi and O. I. Camps, “Wire detection algorithms for navigation,” *NASA Technical Report*, 2002. I

[9] J. Candamo, R. Kasturi, D. Goldgof, and S. Sarkar, “Detection of thin lines using low-quality video from low-altitude aircraft in urban settings,” *IEEE Transactions on aerospace and electronic systems*, vol. 45, no. 3, 2009.

[10] G. Yan, C. Li, G. Zhou, W. Zhang, and X. Li, “Automatic extraction of power lines from aerial images,” *IEEE Geoscience and Remote Sensing Letters*, vol. 4, no. 3, pp. 387–391, 2007.

[11] Z. Li, Y. Liu, R. Walker, R. Hayward, and J. Zhang, “Towards automatic power line detection for a uav surveillance system using pulse coupled neural filter and an improved hough transform,” *Machine Vision and Applications*, vol. 21, no. 5, pp. 677–686, 2010.

[12] J. N. Sanders-Reed, D. J. Yelton, C. C. Witt, and R. R. Galetti, “Passive obstacle detection system (pods) for wire detection,” in *SPIE Defense, Security, and Sensing*. International Society for Optics and Photonics, 2009, pp. 732 804–732 804.

[13] B. Song and X. Li, “Power line detection from optical images,” *Neurocomputing*, vol. 129, pp. 350–361, 2014.

[14] R. Madaan, D. Maturana, and S. Scherer, “Wire detection using synthetic data and dilated convolutional networks for unmanned aerial vehicles,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, September 2017. I, III, III-B

[15] G. Dubey, R. Madaan, and S. Scherer, “Droan - disparity-space representation for obstacle avoidance: Enabling wire mapping avoidance,” in *Intelligent Robots and Systems (IROS), 2018 IEEE/RSJ International Conference on*. IEEE, 2018. I

[16] C. Sun, R. Jones, H. Talbot, X. Wu, K. Cheong, R. Beare, M. Buckley, and M. Berman, “Measuring the distance of vegetation from powerlines using stereo vision,” *ISPRS journal of photogrammetry and remote sensing*, vol. 60, no. 4, pp. 269–283, 2006. I

[17] J. Ahmad, A. S. Malik, L. Xia, and N. Ashikin, “Vegetation encroachment monitoring for transmission lines right-of-ways: A survey,” *Electric Power Systems Research*, vol. 95, pp. 339–352, 2013. I

[18] R. A. McLaughlin, “Extracting transmission lines from airborne lidar data,” *IEEE Geoscience and Remote Sensing Letters*, vol. 3, no. 2, pp. 222–226, 2006. I, II

[19] Y. Jwa and G. Sohn, “A piecewise catenary curve model growing for 3D power line reconstruction,” *Photogrammetric Engineering & Remote Sensing*, vol. 78, no. 12, pp. 1227–1240, 2012. I, II

[20] S. H. Pepe, “Polynomial and catenary curve fits to human dental arches,” *Journal of Dental Research*, vol. 54, no. 6, pp. 1124–1132, 1975. II

[21] A. R. Burdi and J. H. Lillie, “A catenary analysis of the maxillary dental arch during human embryogenesis,” *The Anatomical Record*, vol. 154, no. 1, pp. 13–20, 1966. II

[22] J. Lee, S. Chen, G. P. Reece, M. A. Crosby, E. K. Beahm, and M. K. Markey, “A novel quantitative measure of breast curvature based on catenary,” *IEEE Transactions on Biomedical Engineering*, vol. 59, no. 4, pp. 1115–1124, 2012. II

[23] J. Lee, G. P. Reece, and M. K. Markey, “Breast curvature of the upper and lower breast mound: 3D analysis of patients who underwent breast reconstruction,” in *3rd International Conference on 3D Body Scanning Technologies*, 2012, pp. 171–179. II

[24] M. Hofer, A. Wendel, and H. Bischof, “Incremental line-based 3D reconstruction using geometric constraints.” in *BMVC*, 2013. II

[25] L. Liu, D. Ceylan, C. Lin, W. Wang, and N. J. Mitra, “Image-based reconstruction of wire art,” *ACM Transactions on Graphics (TOG)*, vol. 36, no. 4, p. 63, 2017. II

[26] A. W. Fitzgibbon, “Robust registration of 2D and 3D point sets,” *Image and Vision Computing*, vol. 21, no. 13-14, pp. 1145–1153, 2003. II, III-B

[27] M. Kaess, R. Zboinski, and F. Dellaert, “MCMC-based multiview reconstruction of piecewise smooth subdivision curves with a variable number of control points,” in *European Conference on Computer Vision*. Springer, 2004, pp. 329–341. II

[28] M. Kuse and S. Shen, “Robust camera motion estimation using direct edge alignment and sub-gradient method,” in *Robotics and Automation (ICRA), 2016 IEEE International Conference on*. IEEE, 2016, pp. 573–579. II

[29] Y. Ling, M. Kuse, and S. Shen, “Edge alignment-based visual–inertial fusion for tracking of aggressive motions,” *Autonomous Robots*, vol. 42, no. 3, pp. 513–528, 2018. II, III-B

[30] C. Zhou, J. Yang, C. Zhao, and G. Hua, “Fast, accurate thin-structure obstacle detection for autonomous mobile robots,” in *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, July 2017, pp. 318–327. II

[31] J. Jose Tarrio and S. Pedre, “Realtime edge-based visual odometry for a monocular camera,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 702–710. II

[32] J. L. Troutman, *Variational calculus with elementary convexity*. Springer Science & Business Media, 2012. III-A

[33] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon, “Bundle adjustment - a modern synthesis,” in *International workshop on vision algorithms*. Springer, 1999, pp. 298–372. III-A

[34] M. A. Branch, T. F. Coleman, and Y. Li, “A subspace, interior, and conjugate gradient method for large-scale bound-constrained minimization problems,” *SIAM Journal on Scientific Computing*, vol. 21, no. 1, pp. 1–23, 1999. III-C

[35] scikitlearn, “Trust-region-reflective least squares,” 2011. [Online]. Available: https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.least_squares.html III-C

[36] G. Borgefors, “Distance transformations in digital images,” *Computer vision, graphics, and image processing*, vol. 34, no. 3, pp. 344–371, 1986. III-D

[37] D. P. Huttenlocher, G. A. Klanderman, and W. J. Rucklidge, “Comparing images using the Hausdorff distance,” *IEEE Transactions on pattern analysis and machine intelligence*,

vol. 15, no. 9, pp. 850–863, 1993. IV-A

[38] N. Aspert, D. Santa-Cruz, and T. Ebrahimi, "Mesh: Measuring errors between surfaces using the Hausdorff distance," in *Multimedia and Expo, 2002. ICME'02. Proceedings. 2002 IEEE International Conference on*, vol. 1. IEEE, 2002, pp. 705–708. IV-A

[39] K. Schauwecker, "SP1: Stereo vision in real time," 2015. IV-B

[40] J. Civera, A. J. Davison, and J. M. Montiel, "Inverse depth parametrization for monocular SLAM," *IEEE transactions on robotics*, vol. 24, no. 5, pp. 932–945, 2008. VI